Rockchip RK3399Pro TRM Part1

Revision 1.0
Jan. 2019

Revision History

Date	Revision	Description
2019-1-10	1.0	Initial Release

Table of Content

Figure Index

Table Index

NOTICE

1. By using this document, you hereby unequivocally acknowledge that you have read and agreed to be bound by the contents of this notice.
2. Fuzhou Rockchip Electronics Co., Ltd. ("Rockchip") may make changes to any information in this document at any time without any prior notice. The information herein is subject to change without notice. Do not finalize a design with this information.
3. Information in this document is provided in connection with Rockchip products.

4. THIS DOCUMENT IS PROVIDED "AS IS" WITHOUT ANY WARRANTY OR CONDITION OF ANY KIND, EITHER EXPRESS, IMPLIED OR STATUTORY, INCLUDING, WITHOUT LIMITATION, ANY WARRANTY OR CONDITION WITH RESPECT TO MERCHANTABILITY, FITNESS FOR ANY PARTICULAR PURPOSE, OR NON-INFRINGEMENT.ROCKCHIP DOES NOT ASSUME ANY RESPONSIBILITY AND LIABILITY FOR ITS USE NOR FOR ANY INFRINGEMENT OF PATENTS OR OTHER RIGHTS OF THE THIRD PARTIES WHICH MAY RESULT FROM ITS USE.
5. Rockchip products described in this document are not designed, intended for use in medical, life saving, life sustaining, critical control or safety systems, or in nuclear facility application.
6. Rockchip and Rockchip logo are trademarks or registered trademarks of Rockchip in China and other countries. All referenced brands, product names, service names and trademarks in this document are the property by their respective owners.

Chapter 1 Introduction

1.1 Overview

RK3399Pro is a low power, high performance processor for computing, personal mobile internet devices and other smart device applications. Based on Big.Little architecture, it integrates dual-core Cortex-A72 and quad-core Cortex-A53 with separate NEON coprocessor.
Equipped with one powerful neural network process unit(NPU), it supports mainstream platforms in the market, such as caffe, tensor flow, and so on.
Many embedded powerful hardware engines provide optimized performance for high-end application. RK3399Pro supports multi-format video decoders and encoders.
Embedded 3D GPU makes RK3399Pro completely compatible with OpenGL ES1.1/2.0/3.0/3.1, OpenCL and DirectX 11.1. Special 2D hardware engine with MMU will maximize display performance and provide very smooth operation.

1.2 Features

The features listed below which may or may not be present in actual product, may be subject to the third party licensing requirements. Please contact Rockchip for actual product feature configurations and licensing requirements.

1.2.1 Microprocessor

- Dual-core ARM Cortex-A72 MPCore processor and Quad-core ARM Cortex-A53 MPCore processor, both are high-performance, low-power and cached application processors
- Two CPU clusters. Big cluster with dual-coreCortex-A72 is optimized for high-performance and little cluster with quad-core Cortex-A53 is optimized for low power.
- Full implementation of the ARM architecture v8-A instruction set, ARM Neon Advanced SIMD (single instruction, multiple data) support for accelerating media and signal processing
- ARMv8 Cryptography Extensions
- CCI500 ensures the memory coherency between the two clusters
- Each Cortex-A72 integrates 48KB L1 instruction cache and 32KB L1 data cache with 4-way set associative. Each Cortex A53 integrates 32KB L1 instruction cache and 32kB L1 data cache separately with 4-way set associative
- 1MB unified L2 Cache for Big cluster, 512KB unified L2 Cache for Little cluster
- TrustZone technology support
- Full CoreSight debug solution
  - Debug and trace visibility of whole systems
  - ETM trace support
  - Invasive and non-invasive debug
- Eight separate power domains for CPU core system to support internal power switch and externally turn on/off based on different application scenario
  - PD_A72_B0: 1st Cortex-A72 + Neon + FPU + L1 I/D cache of big cluster
  - PD_A72_B1: 2nd Cortex-A72+ Neon + FPU + L1 I/D cache of big cluster
  - PD_SCU_B: SCU + L2 Cache controller, and including PD_A72_B0, PD_A72_B1, debug logic of big cluster
  - PD_A53_L0: 1st Cortex-A53 + Neon + FPU + L1 I/D Cache of little cluster
  - PD_A53_L1: 2nd Cortex-A53 + Neon + FPU + L1 I/D Cache of little cluster
  - PD_A53_L2: 3rd Cortex-A53 + Neon + FPU + L1 I/D Cache of little cluster
  - PD_A53_L3: 4th Cortex-A53 + Neon + FPU + L1 I/D Cache of little cluster
  - PD_SCU_L: SCU + L2 Cache controller, and including PD_A53_L0, PD_A53_L1, PD_A53_L2, PD_A53_L3, debug logic of little cluster
- Two isolated voltage domain to support DVFS for big cluster and little cluster separately.

1.2.2 Neural Process Unit

- Support 1920 Int8 MAC operations per cycle

- Support 64 FP16 MAC operations per cycle
- Support 192 Int16 MAC operations per cycle
- 512KB internal buffer
- One isolated voltage domain to support DVFS

1.2.3 Boot

- Support system boot from the following device :
  - SPI interface
  - eMMC interface
  - SD/MMC interface
- Support system code download by the following interface:
  - USB OTG interface

1.2.4 Internal Memory

- Internal BootROM
  - Size : 32KB
- Internal SRAM
  - Size : 200KB
  - Support security and non-security access
  - Security or non-security space is software programmable
  - Security space can be 0KB,4KB,8KB,12KB,16KB,… up to 64KB by 4KB step

1.2.5 External Memory or Storage device

- NPU Dedicated Dynamic Memory Interface (DDR3/DDR3L/LPDDR2/LPDDR3)
  - Compatible with JEDEC standards
  - Compatible with DDR3-1600/DDR3L-1600/ LPDDR2-1066 /LPDDR3-1600
    - Support 32-bit data width, 2 ranks (chip selects), max 2GB addressing space per rank, total addressing space is 2GB(max)
- Dual-Channel Dynamic Memory Interface (DDR3/DDR3L/LPDDR3/LPDDR4) ①
  - Compatible with JEDEC standard DDR3-1866 /DDR3L-1866 /LPDDR3-1866 / LPDDR4 SDRAM
  - Support 2 channels, each channel is 16 or 32bits data width
  - Support up to 2 ranks (chip selects) for each channel; totally 4GB(max) address space. Maximum address space of one rank in a channel is also 4GB, which is software-configurable
- eMMC Interface
  - Fully compliant with JEDEC eMMC 5.1and eMMC 5.0 specification
  - There is only one eMMC interface
  - It is backward compliant with eMMC 4.51 and earlier versions specification.
  - Supports HS400, HS200, DDR50 and legacy operating modes.
- SD/MMC Interface
  - Compatible with SD3.0, MMC ver4.51
  - There are 2 MMC interfaces which can be configured as SD/MMC or SDIO
  - Data bus width is 4bits

1.2.6 System Component

- Cortex-M0
  - Two Cortex-M0 inside RK3399Pro to cooperate with Cortex-A72/Cortex-A53
  - Fast code execution permits slower processor clock or increases sleep mode time
  - Deterministic, high-performance interrupt handling for time-critical applications
- CRU (clock & reset unit)
  - Support clock gating control for individual components inside RK3399Pro
  - One oscillator with 24MHz clock input and 8 embedded PLLs
  - Support global soft-reset control for whole SOC, also individual soft-reset for every components
- PMU (power management unit)
  - Multiple configurable work modes to save power by different frequency or

automatic clock gating control or power domain on/off control
  - Lots of wakeup sources in different mode
  - 6 separate voltage domains
  - 30 separate power domains, which can be power up/down by software based on different application scenes
- Timer
  - 14 on-chip 64-bit Timers in SoC with interrupt-based operation for non-secure application
  - 12 on-chip 64-bit Timers in SoC with interrupt-based operation for secure application
  - Provide two operation modes: free-running and user-defined count
  - Support timer work state checkable
  - Fixed 24MHz clock input
- PWM
  - Four on-chip PWMs with interrupt-based operation
  - Embedded 32-bit timer/counter facility
  - Support capture mode
  - Support continuous mode or one-shot mode
  - Provides reference mode and output various duty-cycle waveform
- Watchdog
  - Three Watchdogs in SoC with 32-bit counter width
  - Counter counts down from a preset value to 0 to indicate the occurrence of a timeout
  - WDT can perform two types of operations when timeout occurs:
    - Generate a system reset
    - First generate an interrupt and if this is not cleared by the service routine by the time a second timeout occurs then generate a system reset
  - Programmable reset pulse length
  - Totally 16 defined-ranges of main timeout period
- Mailbox
  - Two Mailboxes in SoC to service multi-core communication
  - Support four mailbox elements per mailbox, each element includes one data word, one command word register and one flag bit that can represent one interrupt
  - Provide 32 lock registers for software to use to indicate whether mailbox is occupied
- Bus Architecture
  - 128-bit/64-bit/32-bit multi-layer AXI/AHB/APB composite bus architecture
  - CCI500 embedded to support two clusters cache coherency
- Interrupt Controller
  - Support 8 PPI interrupt source and 148 SPI interrupt sources input from different components inside RK3399Pro
  - Support 16 software-triggered interrupts
  - Input interrupt level is fixed, high-level sensitive for SPI and low-level sensitive for PPI
  - Support Locality-specific Peripheral Interrupts (LPIs). These interrupts are generated by a peripheral writing to a memory-mapped register in the controller
  - Two AXI stream interrupt interfaces separately for each cluster
  - Support different interrupt priority for each interrupt source, and they are always software-programmable
- DMAC
  - Linked list DMA function is supported to complete scatter-gather transfer
  - Support data transfer types with memory-to-memory, memory-to-peripheral, peripheral-to-memory
  - Signals the occurrence of various DMA events using the interrupt output signals
  - Mapping relationship between each channel and different interrupt outputs is software-programmable
  - Two embedded DMA controller, BUS_DMAC is for bus system, PERI_DMAC is for peripheral system

- DMAC0 features:
  - 6 channels totally
  - 10 hardware request from peripherals
  - 2 interrupt output
  - Support TrustZone technology and programmable secure state for each DMA channel
- DMAC1 features:
  - 8 channels totally
  - 20 hardware request from peripherals
  - 2 interrupt output
  - Support TrustZone technology and programmable secure state for each DMA channel

- Security system
  - Support TrustZone technology for the following components inside RK3399Pro
    - Cortex-A72, support security and non-security mode, switch by software
    - Cortex-A53, support security and non-security mode, switch by software
    - Except Cortex-A72 and Cortex-A53, the other masters in the SoC can also support security and non-security mode by software-programmable
    - Some slave components in SoC can only be addressed by security master and the other slave components can be addressed by security master or non-security master by software-programmable
    - Internal memory, part of space is addressed only in security mode, detailed size is software-programmable together with TZMA (TrustZone memory adapter)
    - External DDR space can be divided into eight parts; each part can be software-programmable to be addressed in security mode or non-security mode
  - Embedded dual-channel encryption and decryption engine
    - Support AES 128/192/256-bit key mode, ECB/CBC/CTR/XTS chain mode, Slave/FIFO mode
    - Support DES/3DES (ECB and CBC chain mode), 3DES (EDE/EEE key mode), Slave/FIFO mode
    - Support SHA1/SHA256/MD5(with hardware padding) HASH function, FIFO mode only
    - Support 160-bit Pseudo Random Number Generator (PRNG)
    - Support 256-bit True Random Number Generator (TRNG)
    - Support PKA 512/1024/2048-bit Exp Modulator
  - Support security boot
  - Support security debug

1.2.7 Video CODEC

- Video Decoder
  - H.264/AVC, Base/Main/High/High10 profile @ level 5.1; up to 4Kx2K @ 30fps
  - H.265/HEVC, Main/Main10 profile @ level 5.1 High-tier; up to 4Kx2K @ 60fps
  - VP9, profile 0, up to 4Kx2K @ 60fps
  - MPEG-1, ISO/IEC 11172-2, up to 1080P @ 60fps
  - MPEG-2, ISO/IEC 13818-2, SP@ML, MP@HL, up to 1080P @ 60fps
  - MPEG-4, ISO/IEC 14496-2, SP@L0-3, ASP@L0-5, up to 1080P @ 60fps
  - VC-1, SP@ML, MP@HL, AP@L0-3, up to 1080P @ 60fps
  - MVC is supported based on H.264 or H.265, up to 1080P @ 60fps
  - Output data format YUV420 semi-planar, YUV400(monochrome), YUV422 is supported by H.264
  - For MPEG-4, GMC (global motion compensation) not supported
  - For VC-1, up-scaling and range mapping are supported in image post-processor
- Video Encoder
  - Support video encoder for H.264 UP to HP@level4.1, MVC and VP8
  - Only support I and P slices, not B slices
  - Input data format:

- YCbCr 4:2:0 planar
- YCbCr 4:2:0 semi-planar
- YCbYCr 4:2:2
- CbYCrY 4:2:2 interleaved
- RGB444 and BGR444
- RGB555 and BGR555
- RGB565 and BGR565
- RGB888 and BRG888
- RGB101010 and BRG101010

- Image size is from 96x96 to 1920x1080(Full HD)
- Maximum frame rate is up to 1920x1080@30FPS[②]

1.2.8 JPEG CODEC

- JPEG Decoder
  - Input JPEG file: YCbCr 4:0:0, 4:2:0, 4:2:2, 4:4:0, 4:1:1 and 4:4:4 sampling formats
  - Output raw image: YCbCr 4:0:0, 4:2:0, 4:2:2, 4:4:0, 4:1:1 and 4:4:4 semi-planar
  - Decoder size is from 48x48 to 8176x8176(66.8Mpixels)
  - Support JPEG ROI (region of image) decode
  - Maximum data rate[③] is up to 76million pixels per second
  - Embedded memory management unit(MMU)
- JPEG Encoder
  - Input raw image:
    - YCbCr 4:2:0 planar
    - YCbCr 4:2:0 semi-planar
    - YCbYCr 4:2:2
    - CbYCrY 4:2:2 interleaved
    - RGB444 and BGR444
    - RGB555 and BGR555
    - RGB565 and BGR565
    - RGB888 and BRG888
    - RGB101010 and BRG101010
  - Output JPEG file: JFIF file format 1.02 or Non-progressive JPEG
  - Encoder image size up to 8192x8192(64million pixels) from 96x32
  - Maximum data rate[③] up to 90million pixels per second

1.2.9 Image Enhancement-Processor (IEP)

- Image format
  - Input data: XRGB/RGB565/YUV420/YUV422
  - Output data: ARGB/RGB565/YUV420/YUV422
  - Max resolution for dynamic image
    - De-interlace: 1920x1080
    - Sampling noise reduction: 1920x1080
    - Compression noise reduction: 4096x2304
    - Enhancement: 4096x2304
- Enhancement
  - Gamma adjustment with programmable mapping table
  - Hue/Saturation/Brightness/Contrast enhancement
  - Programmable distance table for detail and edge enhancement
- Noise reduction
  - Spatial sampling noise reduction
  - Temporal sampling noise reduction
- De-interlace
  - Input 4 fields, output 2 frames mode
  - Input 4 fields, output 1 frames mode
  - Input 2 fields, output 1 frames mode

1.2.10 Graphics Engine

- 3D Graphics Engine:
  - ARM Mali-T860MP4 GPU, support OpenGL ES1.1/2.0/3.0, OpenCL1.2, DirectX11.1 etc.
  - Embedded 4 shader cores with shared hierarchical tiler
  - Provide MMU and L2 Cache with 256KB size
- 2D Graphics Engine:
  - Data format
    - Support input of ARGB/RGB888/RGB565/RGB4444/RGB5551/YUV420/YUV422
    - Support input of YUV422SP(10-bit)/YUV420SP(10-bit)
    - Support output of ARGB/RGB888/RGB565/RGB4444/RGB5551/YUV420/YUV422
    - Support output of YVYU422/420
    - Max resolution: 8192x8192 source, 4096x4096 destination
  - Scaling
    - Support scaling up and down
    - Arbitrary non-integer scaling ratio，from 1/16 to 16
  - Rotation
    - 0, 90, 180, 270 degree rotation
    - x-mirror, y-mirror& rotation operation
  - BitBLT
  - Alpha Blending

1.2.11 Video IN/OUT

- Camera Interface
  - One or two MIPI-CSI input interface
- Image Signal Processer
  - Input interface
    - DVP interface
      - ITU-R BT601/656 with raw8/raw10/raw12
    - MIPI interface
      - Support x1/x2/x4 DPHY RX data lanes
      - Support RAW8, RAW10, RAW12
    - Maximum input resolution is 4416x3312
  - ISP process
    - Support Black level compensation
    - Support 4 channels of Lens shade correction
    - Support AF/AWB/AE/Hist
  - Output interface
    - Support output format :
      - YUV422sp/YUV420sp, with UV swap
      - RGB888/RGB666/RGB565
      - RAW8/RAW12
  - Display Interface
    - Embedded two VOP, output from the following display interface.
      - Two MIPI-DSI port, and one of which can be configured with MIPI-CSI2
      - One eDP port
      - One DP port
      - One HDMI port
    - Support AFBC function co-operation with GPU
- Video Output Processor(VOP_BIG)
  - Display interface
    - HDMI interface
      - Support 480p/480i/576p/576i/720p/1080p/1080i/4k
      - Support RGB/YUV420(up to 10-bit) format
    - DP interface
      - Support progressive/interlace

- Support RGB/YUV420/YUV422/YUV444(up to 10-bit) format
- MIPI interface
  - MIPI DCS command mode
  - Dual-MIPI
- EDP interface
- Max resolution
  - Max input resolution：4096x2304
  - Max output resolution：4096x2160
- Scanning timing 8192x4096
- Support configurable polarity of DCLK/HSYNC/VSYNC/DEN

- Display process
  - GAMMA
  - X-MIRROR, Y-MIRROR
  - Post scale down for TV over scan
- Layer process
  - Background layer
    - programmable 30-bit color
  - Afbcd
    - format: ARGB8888/RGB888/RGB565
    - win_sel(win0/win1/win2/win3)
  - Win0/Win1 layer
    - Support data format
      - RGB888, ARGB888, RGB565,
      - YCbCr420SP, YCbCr422SP, CbCr444SP, YUYV420, YUYV422, YVYU420, YVYU422
      - RGB(8-bit), YUV(8-bit/10-bit), YVYU/YUYV(8-bit)
    - Support 1/8 to 8 scaling-down and scaling-up engine
  - Win2/Win3 layer
    - Support data format
      - RGB888, ARGB888, RGB565
      - 8BPP
    - 4 display regions
      - only one region at one scanning line
  - Hardware Cursor layer
    - Support data format
      - RGB888, ARGB888, RGB565
      - 8BPP
  - Overlay
    - support RGB and YUV domain overlay
    - Support 6 layers, background/win0/win1/win2/win3/hwc
    - Alpha blending
- Write back
  - Support format
    - RGB565(8-bit), RGB888P(8-bit)
    - YUV420(8-bit)
  - Support scale
    - horizontal scale down, 0.25~1.0
    - vertical throw odd/even line

- Video Output Processor(VOP_LIT)
  - Display interface
    - HDMI interface
      - Support 480p/480i/576p/576i/720p/1080p/1080i
      - Support RGB format
    - DP interface
      - Support progressive/interlace
      - Support RGB/YUV420/YUV422/YUV444format
    - MIPI interface

        - MIPI DCS command mode
        - Dual-MIPI
    - EDP interface
    - Max resolution
        - Max input resolution：4096x2304
        - Max output resolution：2560x1600
    - Scanning timing 8192x4096
    - Support configurable polarity of DCLK/HSYNC/VSYNC/DEN
- Display process
    - GAMMA
    - X-MIRROR, Y-MIRROR
    - Post scale down for TV overscan
- Layer process
    - Background layer
        - Programmable 30-bit color
    - Win0 layer
        - Support data format
            - RGB888, ARGB888, RGB565,
            - YCbCr420SP, YCbCr422SP, CbCr444SP, YUYV420, YUYV422, YVYU420, YVYU422
            - RGB(8-bit), YUV(8-bit), YVYU/YUYV(8-bit)
        - Support 1/8 to 8 scaling-down and scaling-up engine
    - Win2 layer
        - Support data format
            - RGB888, ARGB888, RGB565
            - 8BPP
        - 4 display regions
            - only one region at one scanning line
    - Hardware Cursor layer
        - Support data format
            - RGB888, ARGB888, RGB565
            - 8BPP
        - Support four hwc size: 32x32,64x64,96x96,128x128
    - Overlay
        - support RGB and YUV domain overlay
        - Support 4 layers, background/win0/win2/hwc
        - Alpha blending

1.2.12 HDMI

- Single Physical Layer PHY with support for HDMI 1.4 and 2.0 operation
- Support HDCP 1.4/2.2

1.2.13 MIPI PHY

- Embedded 3 MIPI PHY, MIPI0 only for DSI, MIPI1 for DSI or CSI, MIPI2 only for CSI
- Lane operation ranging from 80 Mbps to 1.5 Gbps in forward direction
- Each port has 4 data lane, providing up to 6.0 Gbps data rate

1.2.14 eDP PHY

- Compliant with eDP™ Specification, version 1.3
- Up to 4 physical lanes of 2.7/1.62 Gbps/lane
- Hot plug and unplug detection and link status monitor
- Support Panel Self Refresh(PSR)

1.2.15 DisplayPort

- Compliant with DisplayPort Specification, version 1.2
- Compliant with HDCP2.2 (and compatible with HDCP1.3)
- There is only one DisplayPort controller built-in RK3399Pro which is shared by Type-C

interface
- Supports up to 4kx2k @60fps resolution
- Variety of audio formats–PCM and compressed, over I2S or SPDIF interfaces
- 1Mbps AUX channel

1.2.16 TYPE-C Interface
- Embedded 1 Type-C PHY
- Compliant with USB Type-C Specification, revision 1.1
- Compliant with USB Power Delivery Specification, revision 2.0
- Attach/detach detection and signaling as DFP, UFP and DRP
- Plug orientation/cable twist detection
- Enable/disable VBUS as DFP and DRP (when operating as DFP)
- VBUS detection as UFP and DRP (when operating as UFP)
- USB Power Delivery communication across the CC wire
- Support USB3.0 Type-C and DisplayPort 1.2 Alt Mode on USB Type-C. Two PMA TX-only lanes and two PMA half-duplex TX/RX lanes (can be configured as TX-only or RX-only)
- Up to 5Gbps data rate for USB3.0
- Up to 5.4Gbps(HBR2) data rate for DP1.2, can support 1/2/4lane mode
- Support DisplayPort AUX channel

1.2.17 Audio Interface
- I2S/PCM
  - Three I2S/PCM in SoC
  - I2S0/I2S2 support up to 8 channels TX and 8 channels RX. I2S1 supports up to 2 channels TX and 2 channels RX
  - I2S2 is connected to HDMI and DisplayPort internally. I2S0 and I2S1 are exposed for peripherals.
  - Audio resolution from 16bits to 32bits
  - Sample rate up to 192KHz
  - Provides master and slave work mode, software configurable
  - Support 3 I2S formats (normal, left-justified, right-justified)
  - Support 4 PCM formats (early, late1, late2, late3)
  - I2S and PCM mode cannot be used at the same time
- SPDIF
  - Support two 16-bit audio data store together in one 32-bit wide location
  - Support biphase format stereo audio data output
  - Support 16 to 31-bit audio data left or right justified in 32-bit wide sample data buffer
  - Support 16, 20, 24-bit audio data transfer in linear PCM mode
  - Support non-linear PCM transfer

1.2.18 Connectivity
- SDIO interface
  - Compatible with SDIO 3.0 protocol
  - 4bits data bus width
  - There are 2 total MMC interfaces which may be configured as SD/MMC or SDIO
- GMAC 10/100/1000M ethernet controller
  - Supports 10/100/1000-Mbps RGMII interfaces and 10/100-Mbps RMII interface
  - Supports both full-duplex and half-duplex operation
    - Supports CSMA/CD Protocol for half-duplex operation
    - Supports packet bursting and frame extension in 1000 Mbps half-duplex operation
    - Supports IEEE 802.3x flow control for full-duplex operation
  - Preamble and start-of-frame data (SFD) insertion in Transmit, and deletion in receive paths
  - Automatic CRC and pad generation controllable on a per-frame basis
  - Options for Automatic Pad/CRC Stripping on receive frames

- Programmable fame length to support Standard Ethernet frames
- Supports IEEE 802.1Q VLAN tag detection for reception frames
- Support detection of LAN wake-up frames and AMD Magic Packet frames
- MDIO Master interface for PHY device configuration and management
- Support detection of LAN wake-up frames and AMD Magic Packet frames

- SPI Controller
  - 5 on-chip SPI controllers are inside
  - Support serial-master and serial-slave mode, software-configurable
  - DMA-based or interrupt-based operation
- UART Controller
  - 5 on-chip UART controllers inside RK3399Pro
  - DMA-based or interrupt-based operation
  - Support 5bits,6bits,7bits,8bits serial data transmit or receive
  - Standard asynchronous communication bits such as start,stop and parity
  - Support different input clock for UART operation to get up to 4Mbps or other special baud rate
  - Support non-integer clock divides for baud clock generation
  - Support auto flow control mode for UART0 and UART3
- I2C controller
  - 9 on-chip I2C controllers
  - Multi-master I2C operation
  - Support 7bits and 10bits address mode
  - Serial 8bits oriented and bidirectional data transfers can be made
  - Software programmable clock frequency
  - Data on the I2C-bus can be transferred at rates of up to 100KHz in the Standard-mode, up to 400KHz in the Fast-mode or up to 1MHz in Fast-mode Plus.
- GPIO
  - 5 groups of GPIO (GPIO0~GPIO4)
  - All of GPIOs can be used to generate interrupt to CPU
  - GPIO0 and GPIO1 can be used to wakeup system from low-power mode
  - The pull direction (pull-up or pull-down) for all of GPIOs are software-programmable
  - All of GPIOs are always in input direction in default after power-on-reset
  - The drive strength for all of GPIOs is software-programmable
- USB 3.0 DRD
  - Embedded 1 USB 3.0 interfaces
  - Compatible with USB3.0 Specification
    - Universal Serial Bus 3.0 Specification, Revision 1.0
    - Universal Serial Bus Specification, Revision 2.0
    - Extensible Host Controller Interface for Universal Serial Bus (xHCI), Revision 1.1
  - Support Control/Bulk (including stream)/Interrupt/Isochronous Transfer
  - Supports super-speed (5Gbps)
  - Descriptor Caching and Data Pre-fetching
  - USB 3.0 xHCI Host Features
    - Support up to 64 devices
    - Support 1 interrupter
    - Support 1 USB2.0 port and 1 Super-Speed port
    - Concurrent USB3.0/USB2.0 traffic, up to 8.48Gbps bandwidth
    - Support standard or open-source xHCI and class driver
    - Support xHCI Debug Capability
  - USB 3.0 Dual-Role Device (DRD) Features
    - Static Device operation
    - Static Host operation
    - USB3.0/USB2.0 OTG A device and B device basing on ID
    - UFP/DFP and Data Role Swap Defined in USB TypeC Specification

  - Not support USB3.0/USB2.0 OTG session request protocol(SRP), host negotiation protocol(HNP) and Role Swap Protocol(RSP)
- USB 2.0 Host
  - Embedded 2 USB 2.0 Host interfaces
  - Compatible with USB 2.0Host specification
  - Supports high-speed(480Mbps), full-speed(12Mbps) and low-speed(1.5Mbps) mode
  - Provides 16 host mode channels
  - Support periodic out channel in host mode
- PCIe
  - One PCIe port in RK3399Pro
  - Compatible with PCI Express Base Specification Revision 2.1
  - Dual operation mode: Root Complex(RC)and End Point(EP)
  - Maximum link width is 4, single bi-directional Link interface
  - Support 2.5Gbps serial data transmission rate per lane per direction
  - Support Single Physical PCI Functions in Endpoint Mode
  - Support Legacy Interrupt and MSI and MSI-X interrupt

1.2.19 Others

- Temperature Sensor(TSADC)
  - Embedded 2 channel TSADC in RK3399Pro
  - TSADC clock must be less than 800KHz
  - 10-bit TSADC up to 50Ksps sampling rate
  - -40~125C temperature range and 5℃ temperature resolution
- Successive Approximation Register(SARADC)
  - 6-channel single-ended 10-bit SAR analog-to-digital converter
  - SARADC clock must be less than 13MHz
  - Conversion speed range is up to 1Msps sampling rate
- eFuse
  - Two 1024bits(32x32) high-density electrical Fuse are integrated in RK3399Pro
  - Support standby mode and power down mode
  - Embedded power-switch
  - Embedded four redundancy bits
- Package Type
  - FCBGA1372(body: 27mmx27mm; ball size: 0.35mm)

Notes : ***① DDR3/DDR3L/LPDDR3/LPDDR4 could not be used simultaneously***

② Actual maximum frame rate will depend on the clock frequency and system bus performance

③ Actual maximum data rate will depend on the clock frequency and JPEG compression rate

1.3 Block Diagram

The following diagram shows the basic block diagram.

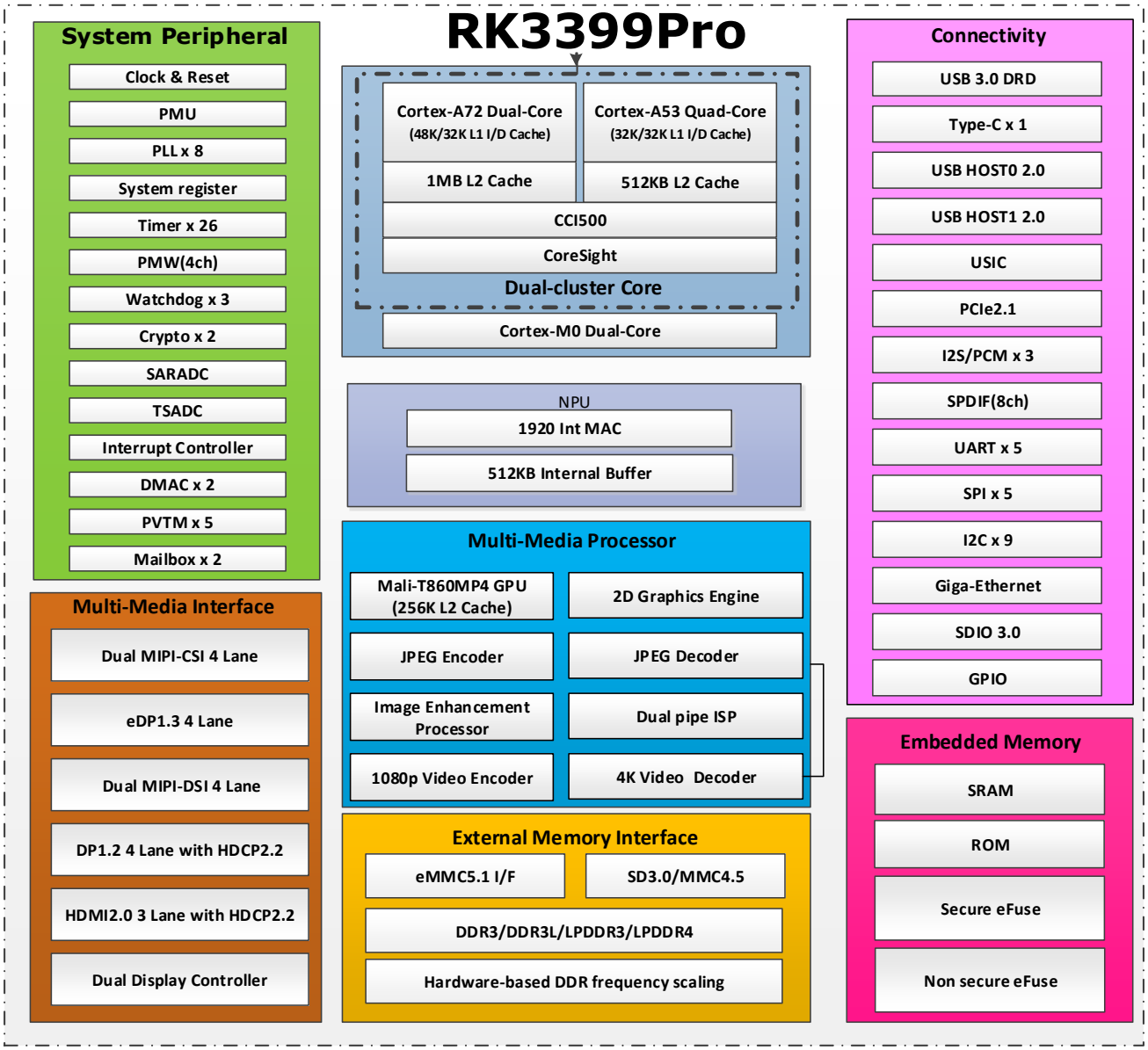


Fig. 1-1 Block Diagram

Notes :
USB3.0 DRD/USB HOST0 2.0 are only for internal use, could not be used by clients

Chapter 2 System Overview

2.1 Address Mapping

RK3399Pro supports to boot from internal bootrom, which supports remap function by software programming. Remap is controlled by SGRF_PMU_CON0[15]. When remap is set to 0, the 0xFFFF0000 address is mapped to bootrom. When remap is set to 1, the 0xFFFF0000 address is mapped to INTMEM0.

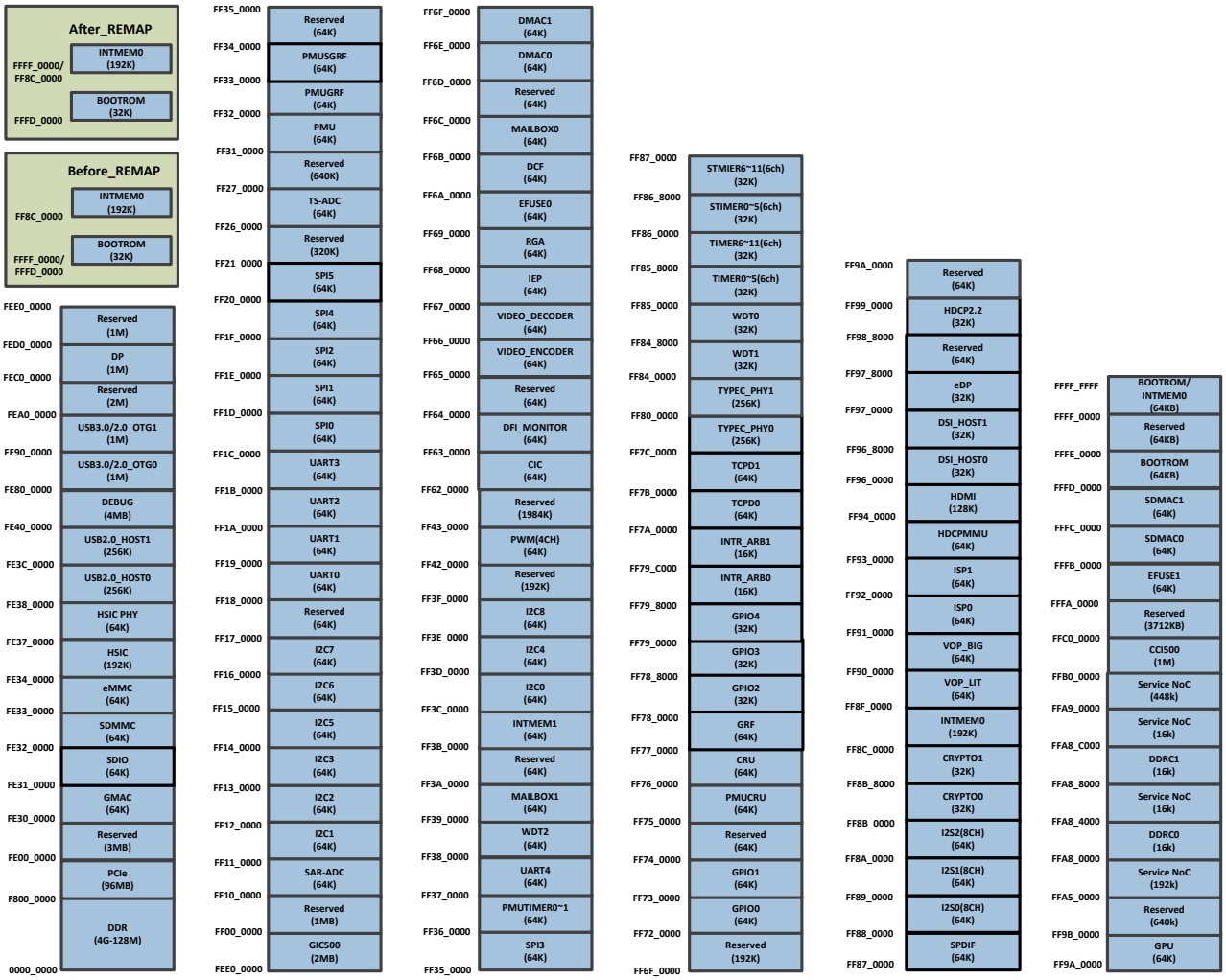


Fig. 2-1 RK3399Pro Address Mapping

2.2 System Boot

RK3399Pro provides system boot from off-chip devices such as serial nand or nor flash, eMMC memory, SD/MMC card. When boot code is not ready in these devices, also provide system code download into them by USB OTG interface. All of the boot code will be stored in internal bootrom. The following is the whole boot procedure for boot code, which will be stored in bootrom in advance.

The following features are supports.

- Support secure boot mode and non-secure boot mode
- Support system boot from the following device:
  - SPI interface
  - eMMC interface
  - SD/MMC Card
- Support system code download by USB OTG

Following figure shows RK3399Pro boot procedure flow.

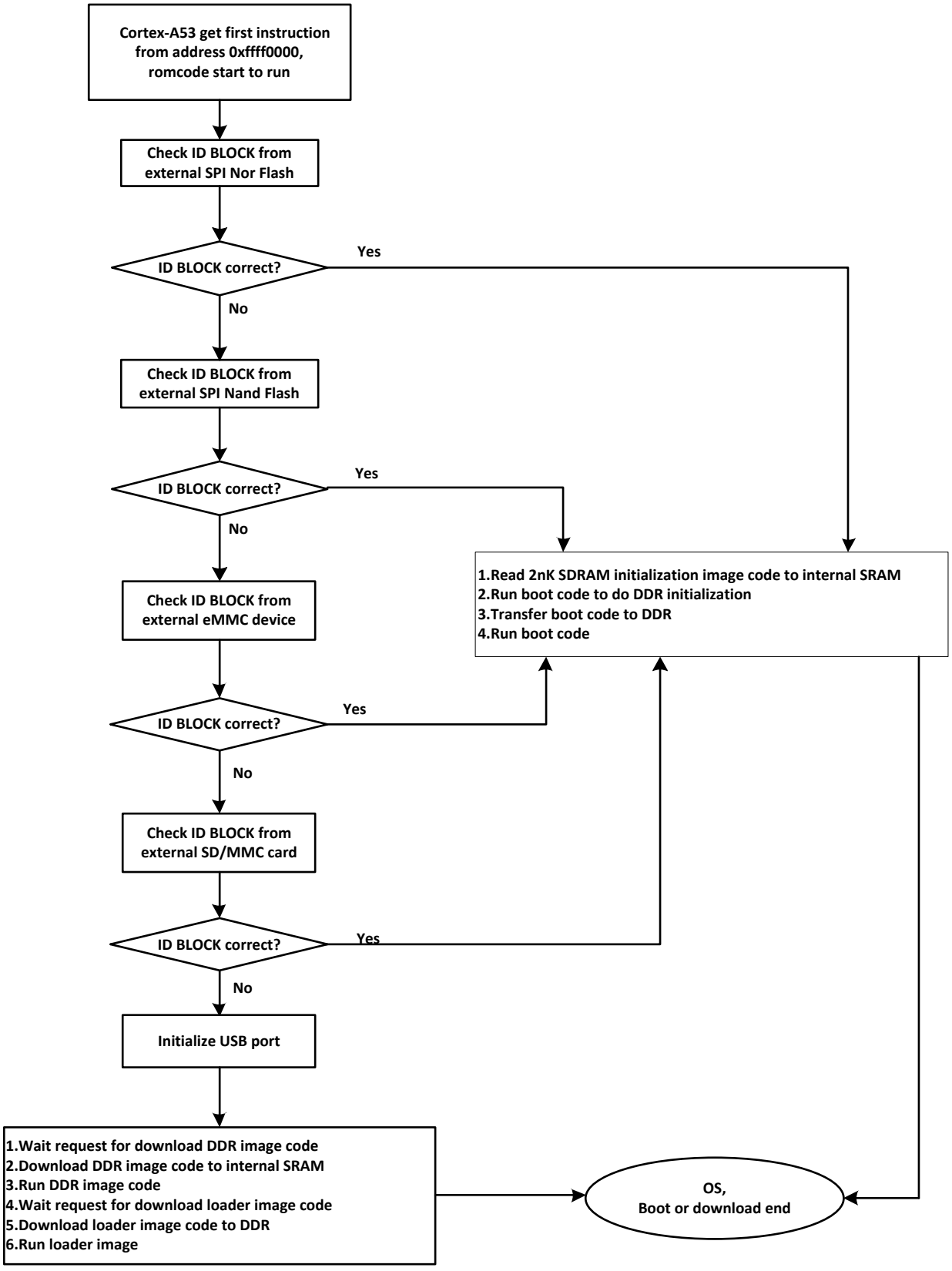


Fig. 2-2 RK3399Pro boot procedure flow

2.3 System Interrupt Connection for Cortex-A72/Cortex-A53

RK3399Pro provides an general interrupt controller(GIC) for Cortex-A72/Cortex-A53, which has 148 SPI(shared peripheral interrupts) interrupt sources and 8 PPI(Private peripheral interrupt) interrupt sources. GIC communicate with CPU through two axi stream interrupt interfaces separately for each cluster. The triggered type for each SPI interrupt is high level sensitive, and for each PPI interrupt is low level sensitive, not programmable. The detailed interrupt sources connection is in the following table. For detailed GIC setting, please refer

to Chapter GIC.

Table 2-1 RK3399Pro Interrupt connection list for Cortex-A72/Cortex-A53

Interrupt Type	Interrupt ID	Source	Polarity
Source(PPI)	16	NA	Low level
	17	NA	Low level
	18	NA	Low level
	19	NA	Low level
	20	NA	Low level
	21	NA	Low level
	22	ncommirq	Low level
	23	npmuirq	Low level
	24	nctiirqack	Low level
	25	nvcpumntirq	Low level
	26	ncnthpirq	Low level
	27	ncntvirq	Low level
	28	NA	Low level
	29	ncntpsirq	Low level
	30	ncntpnsirq	Low level
	31	NA	Low level
Source(SPI)	32	crypto0_int	High level
	33	dcf_done_int	High level
	34	dcf_error_int	High level
	35	ddrc0_int	High level
	36	ddrc1_int	High level
	37	dmac0_perilp_irq_abort	High level
	38	dmac0_perilp_irq	High level
	39	dmac1_perilp_irq_abort	High level
	40	dmac1_perilp_irq	High level
	41	dp_irq	High level
	42	edp_irq	High level
	43	emmccore_int	High level
	44	gmac_int	High level
	45	gmac_pmt_int	High level
	46	gpio0_int	High level
	47	gpio1_int	High level
	48	gpio2_intr	High level
	49	gpio3_intr	High level
	50	gpio4_intr	High level
	51	gpu_irqgpu	High level
	52	gpu_irqjob	High level
	53	gpu_irqmmu	High level
	54	hdcp22_irq	High level
	55	hdmi_irq	High level
	56	hdmi_wakeup_irq	High level

Interrupt Type	Interrupt ID	Source	Polarity
	57	host0_arb_int	High level
	58	host0_ehci_int	High level
	59	host0_linestate_irq	High level
	60	host0_ohci_int	High level
	61	host1_arb_int	High level
	62	host1_ehci_int	High level
	63	host1_linestate_irq	High level
	64	host1_ohci_int	High level
	65	hsic_int	High level
	66	i2c3_int	High level
	67	i2c2_int	High level
	68	i2c7_int	High level
	69	i2c6_int	High level
	70	i2c5_int	High level
	71	i2s0_int	High level
	72	i2s1_int	High level
	73	i2s2_int	High level
	74	iep_intr	High level
	75	isp0_irq	High level
	76	isp1_irq	High level
	77	mipi_dsi_host0_irq	High level
	78	mipi_dsi_host1_irq	High level
	79	errirq_cci	High level
	80	noc_intr	High level
	81	pcie_sys_int	High level
	82	pcie_legacy_int	High level
	83	pcie_client_int	High level
	84	spi2_int	High level
	85	spi1_int	High level
	86	pmu_int	High level
	87	rga_intr	High level
	88	i2c4_int	High level
	89	i2c0_int	High level
	90	i2c8_int	High level
	91	i2c1_int	High level
	92	spi3_int	High level
	93	pwm_int	High level
	94	saradc_int	High level
	95	sd_detectn_irq	High level
	96	sdio_int	High level
	97	sdmmc_int	High level
	98	spdif_int	High level

Interrupt Type	Interrupt ID	Source	Polarity
	99	spi4_int	High level
	100	spi0_int	High level
	101	stimer_intr0	High level
	102	stimer_intr1	High level
	103	stimer_intr2	High level
	104	stimer_intr3	High level
	105	stimer_intr4	High level
	106	stimer_intr5	High level
	107	stimer_intr6	High level
	108	stimer_intr7	High level
	109	stimer_intr8	High level
	110	stimer_intr9	High level
	111	stimer_intr10	High level
	112	stimer_intr11	High level
	113	timer_intr0	High level
	114	timer_intr1	High level
	115	timer_intr2	High level
	116	timer_intr3	High level
	117	timer_intr4	High level
	118	timer_intr5	High level
	119	timer_intr6	High level
	120	timer_intr7	High level
	121	timer_intr8	High level
	122	timer_intr9	High level
	123	timer_intr10	High level
	124	timer_intr11	High level
	125	perf_int_a53	High level
	126	perf_int_a72	High level
	127	pmutimer_int0	High level
	128	pmutimer_int1	High level
	129	tsadc_int	High level
	130	uart1_int	High level
	131	uart0_int	High level
	132	uart2_int	High level
	133	uart3_int	High level
	134	uart4_int	High level
	135	usb3otg0_bvalid_irq	High level
	136	usb3otg0_id_irq	High level
	137	usb3otg0_int	High level
	138	usb3otg0_linestate_irq	High level
	139	usb3otg0_rxdet_irq	High level
	140	usb3otg1_bvalid_irq	High level

Interrupt Type	Interrupt ID	Source	Polarity
	141	usb3otg1_id_irq	High level
	142	usb3otg1_int	High level
	143	usb3otg1_linestate_irq	High level
	144	usb3otg1_rxdet_irq	High level
	145	vcodec_dec_int	High level
	146	vcodec_enc_int	High level
	147	vcodec_mmu_int	High level
	148	vdu_dec_irq	High level
	149	vdu_mmu_irq	High level
	150	vopbig_irq	High level
	151	voplit_irq	High level
	152	wdt0_intr	High level
	153	wdt1_intr	High level
	154	wdt2_int	High level
	155	usb3otg0_pme_generation	High level
	156	usb3otg0_host_legacy_smi_interrupt	High level
	157	usb3otg0_host_sys_err	High level
	158	usb3otg1_pme_generation	High level
	159	usb3otg1_host_legacy_smi_interrupt	High level
	160	usb3otg1_host_sys_err	High level
	161	vopbig_irq_ddr	High level
	162	voplit_irq_ddr	High level
	163	ddr_mon_intr	High level
	164	spi5_int	High level
	165	tcpd_int0	High level
	166	tcpd_int1	High level
	167	crypto1_int	High level
	168	gasket_irq	High level
	169	pcie_rc_mode_elec_idle_irq	High level
	170	N/A	High level
	171	N/A	High level
	172	mailbox1_int[0]	High level
	173	mailbox1_int[1]	High level
	174	mailbox1_int[2]	High level
	175	mailbox1_int[3]	High level
	176	mailbox0_int[0]	High level
	177	mailbox0_int[1]	High level
	178	mailbox0_int[2]	High level
	179	mailbox0_int[3]	High level
	180	exterrirq_pd_core_l	High level
	181	exterrirq_pd_core_b	High level

2.4 System Interrupt Connection for Cortex-M0

RK3399Pro provides two interrupt arbiters for Cortex-M0, one for each Cortex-M0; Interrupt arbiter has 142 SPI interrupt sources and output 18 interrupt signals to M0 after arbitration. The triggered type for each SPI interrupt is high level sensitive, not programmable. The detailed interrupt sources connection is in the following table. In the Table, for perilpm0, the mailbox interrupt is from mailbox0; For pmum0, the mailbox interrupt is from mailbox1; For detailed interrupt arbiter setting, please refer to Chapter Cortex M0.

Table 2-2 RK3399Pro Interrupt connection list for Cortex-M0

Interrupt Type	Interrupt ID	Source	Polarity
Source(SPI)	0	crypto0_int	High level
	1	dcf_done_int	High level
	2	dcf_error_int	High level
	3	ddrc0_int	High level
	4	ddrc1_int	High level
	5	dmac0_perilp_irq_abort	High level
	6	dmac0_perilp_irq	High level
	7	dmac1_perilp_irq_abort	High level
	8	dmac1_perilp_irq	High level
	9	dp_irq	High level
	10	edp_irq	High level
	11	emmccore_int	High level
	12	gmac_int	High level
	13	gmac_pmt_int	High level
	14	gpio0_int	High level
	15	gpio1_int	High level
	16	gpio2_intr	High level
	17	gpio3_intr	High level
	18	gpio4_intr	High level
	19	gpu_irqgpu	High level
	20	gpu_irqjob	High level
	21	gpu_irqmmu	High level
	22	hdcp22_irq	High level
	23	hdmi_irq	High level
	24	hdmi_wakeup_irq	High level
	25	host0_arb_int	High level
	26	host0_ehci_int	High level
	27	host0_linestate_irq	High level
	28	host0_ohci_int	High level
	29	host1_arb_int	High level
	30	host1_ehci_int	High level
	31	host1_linestate_irq	High level
	32	host1_ohci_int	High level

Interrupt Type	Interrupt ID	Source	Polarity
	33	hsic_int	High level
	34	i2c3_int	High level
	35	i2c2_int	High level
	36	i2c7_int	High level
	37	i2c6_int	High level
	38	i2c5_int	High level
	39	i2s0_int	High level
	40	i2s1_int	High level
	41	i2s2_int	High level
	42	iep_intr	High level
	43	isp0_irq	High level
	44	isp1_irq	High level
	45	mipi_dsi_host0_irq	High level
	46	mipi_dsi_host1_irq	High level
	47	errirq_cci	High level
	48	noc_intr	High level
	49	pcie_sys_int	High level
	50	pcie_legacy_int	High level
	51	pcie_client_int	High level
	52	spi2_int	High level
	53	spi1_int	High level
	54	pmu_int	High level
	55	rga_intr	High level
	56	i2c4_int	High level
	57	i2c0_int	High level
	58	i2c8_int	High level
	59	i2c1_int	High level
	60	spi3_int	High level
	61	pwm_int	High level
	62	saradc_int	High level
	63	sd_detectn_irq	High level
	64	sdio_int	High level
	65	sdmmc_int	High level
	66	spdif_int	High level
	67	spi4_int	High level
	68	spi0_int	High level
	69	stimer_intr0	High level
	70	stimer_intr1	High level
	71	stimer_intr2	High level
	72	stimer_intr3	High level
	73	stimer_intr4	High level
	74	stimer_intr5	High level

Interrupt Type	Interrupt ID	Source	Polarity
	75	stimer_intr6	High level
	76	stimer_intr7	High level
	77	stimer_intr8	High level
	78	stimer_intr9	High level
	79	stimer_intr10	High level
	80	stimer_intr11	High level
	81	timer_intr0	High level
	82	timer_intr1	High level
	83	timer_intr2	High level
	84	timer_intr3	High level
	85	timer_intr4	High level
	86	timer_intr5	High level
	87	timer_intr6	High level
	88	timer_intr7	High level
	89	timer_intr8	High level
	90	timer_intr9	High level
	91	timer_intr10	High level
	92	timer_intr11	High level
	93	perf_int_a53	High level
	94	perf_int_a72	High level
	95	pmutimer_int0	High level
	96	pmutimer_int1	High level
	97	tsadc_int	High level
	98	uart1_int	High level
	99	uart0_int	High level
	100	uart2_int	High level
	101	uart3_int	High level
	102	uart4_int	High level
	103	usb3otg0_bvalid_irq	High level
	104	usb3otg0_id_irq	High level
	105	usb3otg0_int	High level
	106	usb3otg0_linestate_irq	High level
	107	usb3otg0_rxdet_irq	High level
	108	usb3otg1_bvalid_irq	High level
	109	usb3otg1_id_irq	High level
	110	usb3otg1_int	High level
	111	usb3otg1_linestate_irq	High level
	112	usb3otg1_rxdet_irq	High level
	113	vcodec_dec_int	High level
	114	vcodec_enc_int	High level
	115	vcodec_mmu_int	High level
	116	vdu_dec_irq	High level

Interrupt Type	Interrupt ID	Source	Polarity
	117	vdu_mmu_irq	High level
	118	vopbig_irq	High level
	119	voplit_irq	High level
	120	wdt0_intr	High level
	121	wdt1_intr	High level
	122	wdt2_int	High level
	123	usb3otg0_pme_generation	High level
	124	usb3otg0_host_legacy_smi_interrupt	High level
	125	usb3otg0_host_sys_err	High level
	126	usb3otg1_pme_generation	High level
	127	usb3otg1_host_legacy_smi_interrupt	High level
	128	usb3otg1_host_sys_err	High level
	129	vopbig_irq_ddr	High level
	130	voplit_irq_ddr	High level
	131	ddr_mon_intr	High level
	132	spi5_int	High level
	133	tcpd_int0	High level
	134	tcpd_int1	High level
	135	crypto1_int	High level
	136	gasket_irq	High level
	137	pcie_rc_mode_elec_idle_irq	High level
	138	N/A	High level
	139	N/A	High level
	140	mailbox*_int[0]	High level
	141	mailbox*_int[1]	High level
	142	mailbox*_int[2]	High level
	143	mailbox*_int[3]	High level

2.5 System DMA Hardware Request Connection

RK3399Pro provides two DMA controllers: DMAC0 and DMAC1, both are in the pd_peri_lp system. As for DMAC0, there are 10 hardware request ports. The trigger type for each of them is high level, not programmable. As for DMAC1, there are 20 hardware request ports. Also the trigger type for each of them is high level, not programmable. For detailed descriptions of DMAC0/DMAC1, please refer to Chapter DMAC. Following two tables include DMAC0/DMAC1 hardware request connection list separately.

Table 2-3 RK3399Pro DMAC0 Hardware Request Connection List

DMAC0		
Req Number	**Source**	**Polarity**
0	I2S0 tx	High level
1	I2S0 rx	High level
2	I2S1 tx	High level
3	I2S1 rx	High level

DMAC0		
Req Number	**Source**	**Polarity**
4	I2S2 tx	High level
5	I2S2 rx	High level
6	PWM rx	High level
7	SPDIF tx	High level
8	SPI5 tx	High level
9	SPI5 rx	High level

Table 2-4 RK3399Pro DMAC1 Hardware Request Connection List

DMAC1		
Req Number	**Source**	**Polarity**
0	UART0 tx	High level
1	UART0 rx	High level
2	UART1 tx	High level
3	UART1 rx	High level
4	UART2 tx	High level
5	UART2 rx	High level
6	UART3 tx	High level
7	UART3 rx	High level
8	UART4 tx	High level
9	UART4 rx	High level
10	SPI0 tx	High level
11	SPI0 rx	High level
12	SPI1 tx	High level
13	SPI1 rx	High level
14	SPI2 tx	High level
15	SPI2 rx	High level
16	SPI3 tx	High level
17	SPI3 rx	High level
18	SPI4 tx	High level
19	SPI4 rx	High level

Chapter 3 Clock & Reset Unit (CRU)

3.1 Overview

The CRU is an APB slave module that is designed for generating all of the internal and system clocks, resets of chip. CRU generates system clock from PLL output clock or external clock source, and generates system reset from external power-on-reset, watchdog timer reset or software reset.
CRU supports the following features:

- Compliance to the AMBA APB interface
- Embedded 8 PLLs: BPLL/LPLL/DPLL/CPLL/GPLL/NPLL/VPLL/PPLL
- Flexible selection of clock source
- Supports the respective gating of all clocks
- Supports the respective software reset of all modules

3.2 Block Diagram

The CRU comprises with:

- PLL
- Register configuration unit
- Clock generate unit
- Reset generate unit

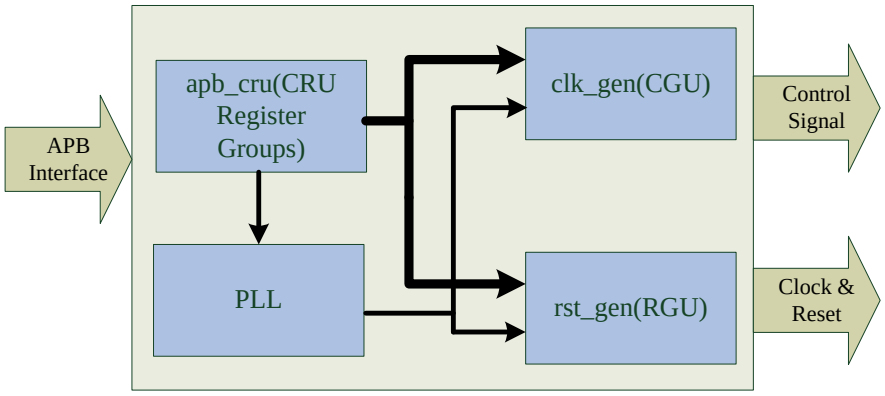


Fig. 3-1 CRU Architecture

3.3 System Clock Solution

The following tables show clock architecture (mux and divider information).

	clk		parents (ID)											
MODULE	ID	CLKNAME	0	1	2	3	4	5	6	7	MUX	GATE	DIV	FRAC
PMU	1000	pclk_pmu_src	8	-	-	-	-	-	-	-	-	PMUGRF0[4]	PS0[4:0]	-
PMU	1001	fclk_cm0s_pmu_ppll_src	8	-	-	-	-	-	-	-	-	PG0[1]	-	-
PMU	1516	fclk_cm0s_src_pmu	1001	0	-	-	-	-	-	-	GF_PS0[15]	-	PS0[12:8]	-
PMU	1040	clk_spi3_pmu	0	8	-	-	-	-	-	-	PS1[7]	PG0[2]	PS1[6:0]	-
PMU	1004	clk_wifi_div	8	0	-	-	-	-	-	-	PS1[13]	PG0[8]	PS1[12:8]	-
PMU	1005	clk_wifi_frac	1004	-	-	-	-	-	-	-	-	-	-	PS7
PMU	1006	clk_wifi_pmu	1004	1005	-	-	-	-	-	-	PS1[14]	-	-	-
PMU	1007	clk_timer_src_pmu	0	10	-	-	-	-	-	-	GF_PS1[15]	-	-	-
PMU	1020	clk_i2c0_pmu	8	-	-	-	-	-	-	-	-	PG0[9]	PS2[6:0]	-
PMU	1022	clk_i2c8_pmu	8	-	-	-	-	-	-	-	-	PG0[11]	PS2[14:8]	-
PMU	1021	clk_i2c4_pmu	8	-	-	-	-	-	-	-	-	PG0[10]	PS3[6:0]	-
cif_testout	1011	clk_32k_suspend_pmu	0	-	-	-	-	-	-	-		-	PS4[9:0]	-
PMU	1030	clk_uart4_div	0	8	-	-	-	-	-	-	PS5[10]	PG0[5]	PS5[6:0]	-
PMU	1031	clk_uart4_frac	1030	-	-	-	-	-	-	-	-	PG0[6]	-	PS6
PMU	1032	clk_uart4_pmu	1030	1031	0	-	-	-	-	-	PS5[9:8]	-	-	-
PMU	1012	clk_timer0_pmu	1007	-	-	-	-	-	-	-	-	PG0[3]	-	-
PMU	1013	clk_timer1_pmu	1007	-	-	-	-	-	-	-	-	PG0[4]	-	-
PMU	1014	clk_pvtm_pmu	0	-	-	-	-	-	-	-	-	PG0[7]	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
PMU	1015	pclk_pmu	1000	-	-	-	-	-	-	-	-	PG1[0]	-	-
PMU	1501	pclk_pmugrf_pmu	1000	-	-	-	-	-	-	-	-	PG1[1]	-	-
PMU	1502	pclk_intmem1_pmu	1000	-	-	-	-	-	-	-	-	PG1[2]	-	-
PMU	1503	pclk_gpio0_pmu	1000	-	-	-	-	-	-	-	-	PG1[3]	-	-
PMU	1504	pclk_gpio1_pmu	1000	-	-	-	-	-	-	-	-	PG1[4]	-	-

	clk		parents (ID)											
MODULE	ID	CLKNAME	0	1	2	3	4	5	6	7	MUX	GATE	DIV	FRAC
PMU	1505	pclk_sgrf_pmu	1000	-	-	-	-	-	-	-	-	PG1[5]	-	-
PMU	1506	pclk_noc_pmu	1000	-	-	-	-	-	-	-	-	PG1[6]	-	-
PMU	1507	pclk_i2c0_pmu	1000	-	-	-	-	-	-	-	-	PG1[7]	-	-
PMU	1508	pclk_i2c4_pmu	1000	-	-	-	-	-	-	-	-	PG1[8]	-	-
PMU	1509	pclk_i2c8_pmu	1000	-	-	-	-	-	-	-	-	PG1[9]	-	-
PMU	1510	pclk_rkpwm_pmu	1000	-	-	-	-	-	-	-	-	PG1[10]	-	-
PMU	1511	pclk_spi3_pmu	1000	-	-	-	-	-	-	-	-	PG1[11]	-	-
PMU	1512	pclk_timer_pmu	1000	-	-	-	-	-	-	-	-	PG1[12]	-	-
PMU	1513	pclk_mailbox_pmu	1000	-	-	-	-	-	-	-	-	PG1[13]	-	-
PMU	1514	pclk_uart4_pmu	1000	-	-	-	-	-	-	-	-	PG1[14]	-	-
PMU	1515	pclk_wdt_m0_pmu	1000	-	-	-	-	-	-	-	-	PG1[15]	-	-
PMU	1002	fclk_cm0s_pmu	1516	-	-	-	-	-	-	-	-	PG2[0]	-	-
PMU	1518	sclk_cm0s_pmu	1516	-	-	-	-	-	-	-	-	PG2[1]	-	-
PMU	1519	hclk_cm0s_pmu	1516	-	-	-	-	-	-	-	-	PG2[2]	-	-
PMU	1520	dclk_cm0s_pmu	1516	-	-	-	-	-	-	-	-	PG2[3]	-	-
PMU	1521	hclk_noc_pmu	1516	-	-	-	-	-	-	-	-	PG2[5]	-	-

Note□

*PS**　*PMUCRU_CLKSEL_CON**

*PG**　*PMUCRU_GATE_CON**

*GF_PS**　*glitch-free*

Fig. 3-2 RK3399Pro PMUCRU Clock Architecture Diagram

	clk		parents (ID)											
MODULE	ID	CLKNAME	0	1	2	3	4	5	6	7	MUX	GATE	DIV	FRAC
IO_CLK	0	clk_24m<IO>	-	-	-	-	-	-	-	-	-	-	-	-
PLL	1	lpll	0	-	-	-	-	-	-	-	-	-	-	-
PLL	2	bpll	0	-	-	-	-	-	-	-	-	-	-	-
PLL	3	dpll	0	-	-	-	-	-	-	-	-	-	-	-
PLL	4	cpll	0	-	-	-	-	-	-	-	-	-	-	-

	clk		parents (ID)											
MODULE	ID	CLKNAME	0	1	2	3	4	5	6	7	MUX	GATE	DIV	FRAC
PLL	5	gpll	0	-	-	-	-	-	-	-	-	-	-	-
PLL	6	npll	0	-	-	-	-	-	-	-	-	-	-	-
PLL	7	vpll	0	-	-	-	-	-	-	-	-	-	-	-
PLL	8	ppll	0	-	-	-	-	-	-	-	-	-	-	-
usbphy	9	upll	0	132	-	-	-	-	-	-	S14[15]	-	-	-
IO_CLK	10	clk_32k<IO>	-	-	-	-	-	-	-	-	-	-	-	-
IO_CLK	355	pclkin_cif<IO>	-	-	-	-	-	-	-	-	-	-	-	-
isp	356	pclkin_cifinv	355	-	-	-	-	-	-	-	-	-	INVERT	-
isp	357	pclkin_cifmux	355	356	-	-	-	-	-	-	GRF20[9]	-	-	-
corel	11	clk_core_l_lpll_src	1	-	-	-	-	-	-	-	-	G0[0]	-	-
corel	12	clk_core_l_bpll_src	2	-	-	-	-	-	-	-	-	G0[1]	-	-
corel	13	clk_core_l_dpll_src	3	-	-	-	-	-	-	-	-	G0[2]	-	-
corel	14	clk_core_l_gpll_src	5	-	-	-	-	-	-	-	-	G0[3]	-	-
corel	15	clk_core_l	11	12	13	14	-	-	-	-	GF_S0[7:6]	-	S0[4:0]	-
corel	16	aclkm_core_l	15	-	-	-	-	-	-	-	-	G0[4]	ICG_S0[12:8]	-
corel	17	atclk_core_l	15	-	-	-	-	-	-	-	-	G0[5]	ICG_S1[4:0]	-
corel	18	pclk_dbg_core_l	15	-	-	-	-	-	-	-	-	G0[6]	ICG_S1[12:8]	-
corel	19	clk_pvtm_core_l	0	-	-	-	-	-	-	-	-	G0[7]	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
coreb	20	clk_core_b_lpll_src	1	-	-	-	-	-	-	-	-	G1[0]	-	-
coreb	21	clk_core_b_bpll_src	2	-	-	-	-	-	-	-	-	G1[1]	-	-
coreb	22	clk_core_b_dpll_src	3	-	-	-	-	-	-	-	-	G1[2]	-	-
coreb	23	clk_core_b_gpll_src	5	-	-	-	-	-	-	-	-	G1[3]	-	-
coreb	24	clk_core_b	20	21	22	23	-	-	-	-	GF_S2[7:6]	-	S2[4:0]	-
coreb	25	aclkm_core_b	24	-	-	-	-	-	-	-	-	G1[4]	ICG_S2[12:8]	-
coreb	26	atclk_core_b	24	-	-	-	-	-	-	-	-	G1[5]	ICG_S3[4:0]	-
coreb	27	pclk_dbg_core_b	24	-	-	-	-	-	-	-	-	G1[6]	S3[12:8]	-
coreb	28	pclken_dbg_core_b	27	-	-	-	-	-	-	-	-	-	ICG_S3[14:13]	-

	clk		parents (ID)											
MODULE	ID	CLKNAME	0	1	2	3	4	5	6	7	MUX	GATE	DIV	FRAC
coreb	29	clk_pvtm_core_b	0	-	-	-	-	-	-	-	-	G1[7]	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
gmac	30	aclk_gmac_cpll_src	4	-	-	-	-	-	-	-	-	G6[9]	-	-
gmac	31	aclk_gmac_gpll_src	5	-	-	-	-	-	-	-	-	G6[8]	-	-
gmac	32	aclk_gmac_pre	30	31	-	-	-	-	-	-	S20[7]	G6[10]	S20[4:0]	-
gmac	33	pclk_gmac_pre	32	-	-	-	-	-	-	-	-	G6[11]	S19[10:8]	-
gmac	34	clk_gmac	4	5	6	-	-	-	-	-	S20[15:14]	G5[5]	S20[12:8]	-
gmac	35	clk_rmii_src	34	48	-	-	-	-	-	-	S19[4]	-	-	-
gmac	36	clk_mac_refout	35	-	-	-	-	-	-	-	-	G5[6]	-	-
gmac	37	clk_mac_ref	35	-	-	-	-	-	-	-	-	~GRF5[6]|G5[7]	-	-
gmac	38	clk_rmii_rx_src	35	-	-	-	-	-	-	-	-	~GRF5[6]|G5[8]	-	-
gmac	39	clk_rmii_d2	38	-	-	-	-	-	-	-	-	-	F2	-
gmac	40	clk_rmii_d20	38	-	-	-	-	-	-	-	-	-	F20	-
gmac	41	clk_rmii_tx_src	35	-	-	-	-	-	-	-	-	G5[9]	-	-
gmac	42	clk_rmii_d5	41	-	-	-	-	-	-	-	-	-	F5	-
gmac	43	clk_rmii_d50	41	-	-	-	-	-	-	-	-	-	F50	-
gmac	44	clk_rmii_rx_mux	39	40	-	-	-	-	-	-	GRF5[3]	-	-	-
gmac	45	clk_rmii_tx_mux	41	-	43	42	-	-	-	-	GRF5[5:4]	-	-	-
gmac	46	clk_mac_rx	49	44	-	-	-	-	-	-	GRF5[6]	-	-	-
gmac	47	clk_mac_tx	45	44	-	-	-	-	-	-	GRF5[6]	-	-	-
IO_CLK	48	clkin_gmac<IO>	-	-	-	-	-	-	-	-	-	-	-	-
IO_CLK	49	gmac_phy_rx_clk<IO>	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
i2s	50	clk_i2s0_div	4	5	-	-	-	-	-	-	S28[7]	G8[3]	S28[6:0]	-
i2s	51	clk_i2s0_frac	50	-	-	-	-	-	-	-	-	G8[4]	-	S96
i2s	52	clk_i2s0_mux	50	51	64	65	-	-	-	-	S28[9:8]	-	-	-
i2s	53	clk_i2s0	52	-	-	-	-	-	-	-	-	G8[5]	-	-
i2s	54	clk_i2s1_div	4	5	-	-	-	-	-	-	S29[7]	G8[6]	S29[6:0]	-

	clk		parents (ID)											
MODULE	ID	CLKNAME	0	1	2	3	4	5	6	7	MUX	GATE	DIV	FRAC
i2s	55	clk_i2s1_frac	54	-	-	-	-	-	-	-	-	G8[7]	-	S97
i2s	56	clk_i2s1_mux	54	55	64	65	-	-	-	-	S29[9:8]	-	-	-
i2s	57	clk_i2s1	56	-	-	-	-	-	-	-	-	G8[8]	-	-
i2s	58	clk_i2s2_div	4	5	-	-	-	-	-	-	S30[7]	G8[9]	S30[6:0]	-
i2s	59	clk_i2s2_frac	58	-	-	-	-	-	-	-	-	G8[10]	-	S98
i2s	60	clk_i2s2_mux	58	59	64	65	-	-	-	-	S30[9:8]	-	-	-
i2s	61	clk_i2s2	60	-	-	-	-	-	-	-	-	G8[11]	-	-
i2s	62	clk_i2sout_src	53	57	61	-	-	-	-	-	S31[1:0]	-	-	-
i2s	63	clk_i2sout	62	64	-	-	-	-	-	-	GF_S31[2]	G8[12]	-	-
i2s	64	clk_12m	0	-	-	-	-	-	-	-	-	-	F2	-
IO_CLK	65	clkin_i2s<IO>	-	-	-	-	-	-	-	-	-	-	-	-
i2s	66	clk_spdif_div	4	5	-	-	-	-	-	-	S32[7]	G8[13]	S32[6:0]	-
i2s	67	clk_spdif_frac	66	-	-	-	-	-	-	-	-	G8[14]	-	S99
i2s	68	clk_spdif	66	67	64	65	-	-	-	-	S32[14:13]	G8[15]	-	-
i2s	69	clk_spdif_rec_dptx	4	5	-	-	-	-	-	-	S32[15]	G10[6]	S32[12:8]	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
uart	70	clk_uart0_src	4	5	9	-	-	-	-	-	S33[13:12]	-	-	-
uart	71	clk_uart_src	4	5	-	-	-	-	-	-	S33[15]	-	-	-
uart	72	clk_uart0_div	70	-	-	-	-	-	-	-	-	G9[0]	S33[6:0]	-
uart	73	clk_uart0_frac	72	-	-	-	-	-	-	-	-	G9[1]	-	S100
uart	74	clk_uart0	72	73	0	-	-	-	-	-	S33[9:8]	-	-	-
uart	75	clk_uart1_div	71	-	-	-	-	-	-	-	-	G9[2]	S34[6:0]	-
uart	76	clk_uart1_frac	75	-	-	-	-	-	-	-	-	G9[3]	-	S101
uart	77	clk_uart1	75	76	0	-	-	-	-	-	S34[9:8]	-	-	-
uart	78	clk_uart2_div	71	-	-	-	-	-	-	-	-	G9[4]	S35[6:0]	-
uart	79	clk_uart2_frac	78	-	-	-	-	-	-	-	-	G9[5]	-	S102
uart	80	clk_uart2	78	79	0	-	-	-	-	-	S35[9:8]	-	-	-
uart	81	clk_uart3_div	71	-	-	-	-	-	-	-	-	G9[6]	S36[6:0]	-

	clk		parents (ID)											
MODULE	ID	CLKNAME	0	1	2	3	4	5	6	7	MUX	GATE	DIV	FRAC
uart	82	clk_uart3_frac	81	-	-	-	-	-	-	-	-	G9[7]	-	S103
uart	83	clk_uart3	81	82	0	-	-	-	-	-	S36[9:8]	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
ddrc	85	clk_ddrc_lpll_src	1	-	-	-	-	-	-	-	-	G3[0]	-	-
ddrc	86	clk_ddrc_bpll_src	2	-	-	-	-	-	-	-	-	G3[1]	-	-
ddrc	87	clk_ddrc_dpll_src	3	-	-	-	-	-	-	-	-	G3[2]	-	-
ddrc	88	clk_ddrc_gpll_src	5	-	-	-	-	-	-	-	-	G3[3]	-	-
ddrc	89	clk_ddrc	85	86	87	88	-	-	-	-	S6[5:4]	-	S6[2:0]	-
ddrc	90	clk_ddrc_div2	89	-	-	-	-	-	-	-	-	-	F2	-
ddrc	91	pclk_ddr	4	5	-	-	-	-	-	-	S6[15]	G3[4]	S6[12:8]	-
ddrc	92	clk_pvtm_ddr	0	-	-	-	-	-	-	-	-	G4[11]	-	-
ddrc	93	clk_dfimon0_timer	0	-	-	-	-	-	-	-	-	G3[5]	-	-
ddrc	94	clk_dfimon1_timer	0	-	-	-	-	-	-	-	-	G3[6]	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
cci	95	aclk_cci_cpll_src	4	-	-	-	-	-	-	-	-	G2[0]	-	-
cci	96	aclk_cci_gpll_src	5	-	-	-	-	-	-	-	-	G2[1]	-	-
cci	97	aclk_cci_npll_src	6	-	-	-	-	-	-	-	-	G2[2]	-	-
cci	98	aclk_cci_vpll_src	7	-	-	-	-	-	-	-	-	G2[3]	-	-
cci	99	aclk_cci_pre	95	96	97	98	-	-	-	-	GF_S5[7:6]	G2[4]	S5[4:0]	-
cci	100	clk_cci_trace_cpll_src	4	-	-	-	-	-	-	-	-	G2[5]	-	-
cci	101	clk_cci_trace_gpll_src	5	-	-	-	-	-	-	-	-	G2[6]	-	-
cci	102	clk_cci_trace	100	101	-	-	-	-	-	-	GF_S5[15]	G2[7]	S5[12:8]	-
cci	103	clk_cs_cpll_src	4	-	-	-	-	-	-	-	-	G2[8]	-	-
cci	104	clk_cs_gpll_src	5	-	-	-	-	-	-	-	-	G2[9]	-	-
cci	105	clk_cs_npll_src	6	-	-	-	-	-	-	-	-	G2[10]	-	-
cci	106	clk_cs	103	104	105	-	-	-	-	-	GF_S4[7:6]	-	S4[4:0]	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
vcodec	110	aclk_vcodec_pre	4	5	6	8	-	-	-	-	S7[7:6]	G4[0]	S7[4:0]	-

	clk		parents (ID)											
MODULE	ID	CLKNAME	0	1	2	3	4	5	6	7	MUX	GATE	DIV	FRAC
vcodec	111	hclk_vcodec_pre	110	-	-	-	-	-	-	-	-	G4[1]	S7[12:8]	-
vdu	112	aclk_vdu_pre	4	5	6	8	-	-	-	-	S8[7:6]	G4[2]	S8[4:0]	-
vdu	113	hclk_vdu_pre	112	-	-	-	-	-	-	-	-	G4[3]	S8[12:8]	-
vdu	114	clk_vdu_core	4	5	6	-	-	-	-	-	S9[7:6]	G4[4]	S9[4:0]	-
vdu	115	clk_vdu_ca	4	5	6	-	-	-	-	-	S9[15:14]	G4[5]	S9[12:8]	-
iep	116	aclk_iep_pre	4	5	6	8	-	-	-	-	S10[7:6]	G4[6]	S10[4:0]	-
iep	117	hclk_iep_pre	116	-	-	-	-	-	-	-	-	G4[7]	S10[12:8]	-
rga	118	aclk_rga_pre	4	5	6	8	-	-	-	-	S11[7:6]	G4[8]	S11[4:0]	-
rga	119	hclk_rga_pre	118	-	-	-	-	-	-	-	-	G4[9]	S11[12:8]	-
rga	120	clk_rga_core	4	5	6	8	-	-	-	-	S12[7:6]	G4[10]	S12[4:0]	-
center	121	aclk_center	4	5	6	-	-	-	-	-	GF_S12[15:14]	G3[7]	S12[12:8]	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
gpu	123	aclk_gpu_pre	8	4	5	6	9	-	-	-	GF_S13[7:5]	G13[0]	S13[4:0]	-
gpu	124	clk_pvtm_gpu	0	-	-	-	-	-	-	-	-	G13[1]	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
perihp	126	aclk_perihp_gpll_src	5	-	-	-	-	-	-	-	-	G5[0]	-	-
perihp	125	aclk_perihp_cpll_src	4	-	-	-	-	-	-	-	-	G5[1]	-	-
perihp	127	aclk_perihp	125	126	-	-	-	-	-	-	GF_S14[7]	G5[2]	S14[4:0]	-
perihp	128	hclk_perihp	127	-	-	-	-	-	-	-	-	G5[3]	S14[9:8]	-
perihp	129	pclk_perihp	127	-	-	-	-	-	-	-	-	G5[4]	S14[14:12]	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
usbphy	130	clk_usbphy0_480m<PHY>	-	-	-	-	-	-	-	-	-	-	-	-
usbphy	131	clk_usbphy1_480m<PHY>	-	-	-	-	-	-	-	-	-	-	-	-
usbphy	330	clk_usbphy0_480m_src	130	-	-	-	-	-	-	-	-	G13[12]	-	-
usbphy	331	clk_usbphy1_480m_src	131	-	-	-	-	-	-	-	-	G13[12]	-	-
usbphy	132	clk_usbphy_480m	330	331	-	-	-	-	-	-	S14[6]	-	-	-
usbphy	133	clk_hsicphy	4	5	6	132	-	-	-	-	S19[1:0]	G6[4]	-	-
usbphy	310	clk_usb2phy0_ref	0	-	-	-	-	-	-	-	-	G6[5]	-	-

	clk		parents (ID)											
MODULE	ID	CLKNAME	0	1	2	3	4	5	6	7	MUX	GATE	DIV	FRAC
usbphy	311	clk_usb2phy1_ref	0	-	-	-	-	-	-	-	-	G6[6]	-	-
usbphy	185	clk_uphy0_tcpdphy_ref	0	10	-	-	-	-	-	-	GF_S64[15]	G13[4]	S64[12:8]	-
usbphy	186	clk_uphy0_tcpdcore	0	10	4	5	-	-	-	-	GF_S64[7:6]	G13[5]	S64[4:0]	-
usbphy	187	clk_uphy1_tcpdphy_ref	0	10	-	-	-	-	-	-	GF_S65[15]	G13[6]	S65[12:8]	-
usbphy	188	clk_uphy1_tcpdcore	0	10	4	5	-	-	-	-	GF_S65[7:6]	G13[7]	S65[4:0]	-
usbphy	180	aclk_usb3	4	5	6	-	-	-	-	-	S39[7:6]	G12[0]	S39[4:0]	-
usbphy	183	clk_usb3otg0_suspend	0	10	-	-	-	-	-	-	S40[15]	G12[3]	S40[9:0]	-
usbphy	184	clk_usb3otg1_suspend	0	10	-	-	-	-	-	-	S41[15]	G12[4]	S41[9:0]	-
usbphy	181	clk_usb3otg0_ref	0	-	-	-	-	-	-	-	-	G12[1]	-	-
usbphy	182	clk_usb3otg1_ref	0	-	-	-	-	-	-	-	-	G12[2]	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
sd	325	hclk_sd	4	5	-	-	-	-	-	-	S13[15]	G12[13]	S13[12:8]	-
sd	135	clk_sdio	4	5	6	8	9	0	-	-	S15[10:8]	G6[0]	S15[6:0]	-
sd	136	clk_sdmmc	4	5	6	8	9	0	-	-	S16[10:8]	G6[1]	S16[6:0]	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
pcie	137	clk_pcie_pm	4	5	6	0	-	-	-	-	S17[10:8]	G6[2]	S17[6:0]	-
pcie	138	clk_pciephy_ref100m	6	-	-	-	-	-	-	-	-	G12[6]	S18[15:11]	-
pcie	139	clk_pciephy_ref	0	138	-	-	-	-	-	-	S18[10]	-	-	-
pcie	140	clk_pcie_core_cru	4	5	6	-	-	-	-	-	S18[9:8]	G6[3]	S18[6:0]	-
pcie	141	clk_pcie_core_phy<PHY>	-	-	-	-	-	-	-	-	-	-	-	-
pcie	142	clk_pcie_core	140	141	-	-	-	-	-	-	S18[7]	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
emmc	146	aclk_emmc_gpll_src	5	-	-	-	-	-	-	-	-	G6[12]	-	-
emmc	145	aclk_emmc_cpll_src	4	-	-	-	-	-	-	-	-	G6[13]	-	-
emmc	147	aclk_emmc	145	146	-	-	-	-	-	-	S21[7]	-	S21[4:0]	-
emmc	148	clk_emmc	4	5	6	9	0	-	-	-	S22[10:8]	G6[14]	S22[6:0]	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
perilp0	150	aclk_perilp0_cpll_src	4	-	-	-	-	-	-	-	-	G7[1]	-	-

	clk		parents (ID)											
MODULE	ID	CLKNAME	0	1	2	3	4	5	6	7	MUX	GATE	DIV	FRAC
perilp0	151	aclk_perilp0_gpll_src	5	-	-	-	-	-	-	-	-	G7[0]	-	-
perilp0	152	aclk_perilp0	150	151	-	-	-	-	-	-	GF_S23[7]	G7[2]	S23[4:0]	-
perilp0	153	hclk_perilp0	152	-	-	-	-	-	-	-	-	G7[3]	ICG_S23[9:8]	-
perilp0	154	pclk_perilp0	152	-	-	-	-	-	-	-	-	G7[4]	ICG_S23[14:12]	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
crypto	155	clk_crypto0	4	5	8	-	-	-	-	-	S24[7:6]	G7[7]	S24[4:0]	-
crypto	156	clk_crypto1	4	5	8	-	-	-	-	-	S26[7:6]	G7[8]	S26[4:0]	-
cm0s_perilp	157	fclk_cm0s_cpll_src	4	-	-	-	-	-	-	-	-	G7[6]	-	-
cm0s_perilp	158	fclk_cm0s_gpll_src	5	-	-	-	-	-	-	-	-	G7[5]	-	-
cm0s_perilp	159	fclk_cm0s	157	158	-	-	-	-	-	-	GF_S24[15]	G7[9]	S24[12:8]	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
perilp1	160	hclk_perilp1_cpll_src	4	-	-	-	-	-	-	-	-	G8[1]	-	-
perilp1	161	hclk_perilp1_gpll_src	5	-	-	-	-	-	-	-	-	G8[0]	-	-
perilp1	162	hclk_perilp1	160	161	-	-	-	-	-	-	GF_S25[7]	-	S25[4:0]	-
perilp1	163	pclk_perilp1	162	-	-	-	-	-	-	-	-	G8[2]	ICG_S25[10:8]	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
saradc	165	clk_saradc	0	-	-	-	-	-	-	-	-	G9[11]	S26[15:8]	-
tsadc	166	clk_tsadc	0	10	-	-	-	-	-	-	S27[15]	G9[10]	S27[9:0]	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
cif_testout	173	clk_testout1_pll_src	4	5	6	-	-	-	-	-	S38[7:6]	-	-	-
cif_testout	174	clk_testout1	173	0	-	-	-	-	-	-	S38[5]	G13[14]	S38[4:0]	-
cif_testout	178	clk_testout2_pll_src	4	5	6	-	-	-	-	-	S38[15:14]	-	-	-
cif_testout	179	clk_testout2	178	0	-	-	-	-	-	-	S38[13]	G13[15]	S38[12:8]	-
cif_testout	172	clk_testout2_2io	179	1011	-	-	-	-	-	-	PS4[15]	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
vio	190	aclk_vio	4	5	8	-	-	-	-	-	S42[7:6]	G11[0]	S42[4:0]	-
vio	191	pclk_vio	190	-	-	-	-	-	-	-	-	G11[1]	ICG_S43[4:0]	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

	clk		parents (ID)											
MODULE	ID	CLKNAME	0	1	2	3	4	5	6	7	MUX	GATE	DIV	FRAC
hdcp	193	aclk_hdcp	4	5	8	-	-	-	-	-	S42[15:14]	G11[2]	S42[12:8]	-
hdcp	194	hclk_hdcp	193	-	-	-	-	-	-	-	-	G11[3]	S43[9:5]	-
hdcp	195	pclk_hdcp	193	-	-	-	-	-	-	-	-	G11[10]	S43[14:10]	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
edp	197	pclk_edp	4	5	-	-	-	-	-	-	S44[15]	G11[11]	S44[13:8]	-
edp	201	clk_dp_core	6	4	5	-	-	-	-	-	S46[7:6]	G11[8]	S46[4:0]	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
hdmi	199	clk_hdmi_cec	0	10	-	-	-	-	-	-	S45[15]	G11[7]	S45[9:0]	-
hdmi	200	clk_hdmi_sfr	0	-	-	-	-	-	-	-	-	G11[6]	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
vop0	203	aclk_vop0_pre	7	4	5	6	-	-	-	-	GF_S47[7:6]	G10[8]	S47[4:0]	-
vop0	204	hclk_vop0_pre	203	-	-	-	-	-	-	-	-	G10[9]	S47[12:8]	-
vop1	205	aclk_vop1_pre	7	4	5	6	-	-	-	-	GF_S48[7:6]	G10[10]	S48[4:0]	-
vop1	206	hclk_vop1_pre	205	-	-	-	-	-	-	-	-	G10[11]	S48[12:8]	-
vop0	207	dclk_vop0_div	7	4	5	-	-	-	-	-	GF_S49[9:8]	G10[12]	S49[7:0]	-
vop0	208	dclk_vop0_frac	207	-	-	-	-	-	-	-	-	-	-	S106
vop0	209	dclk_vop0	207	208	-	-	-	-	-	-	S49[11]	-	-	-
vop1	210	dclk_vop1_div	7	4	5	-	-	-	-	-	GF_S50[9:8]	G10[13]	S50[7:0]	-
vop1	211	dclk_vop1_frac	210	-	-	-	-	-	-	-	-	-	-	S107
vop1	212	dclk_vop1	210	211	-	-	-	-	-	-	S50[11]	-	-	-
vop0	214	clk_vop0_pwm	7	4	5	0	-	-	-	-	GF_S51[7:6]	G10[14]	S51[4:0]	-
vop1	215	clk_vop1_pwm	7	4	5	0	-	-	-	-	GF_S52[7:6]	G10[15]	S52[4:0]	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
isp	217	aclk_isp0	4	5	8	-	-	-	-	-	S53[7:6]	G12[8]	S53[4:0]	-
isp	218	hclk_isp0	217	-	-	-	-	-	-	-	-	G12[9]	S53[12:8]	-
isp	219	clk_isp0	4	5	6	-	-	-	-	-	S55[7:6]	G11[4]	S55[4:0]	-
isp	221	aclk_isp1	4	5	8	-	-	-	-	-	S54[7:6]	G12[10]	S54[4:0]	-
isp	222	hclk_isp1	221	-	-	-	-	-	-	-	-	G12[11]	S54[12:8]	-

	clk		parents (ID)											
MODULE	ID	CLKNAME	0	1	2	3	4	5	6	7	MUX	GATE	DIV	FRAC
isp	223	clk_isp1	4	5	6	-	-	-	-	-	S55[15:14]	G11[5]	S55[12:8]	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
cif	225	clk_cifout_pll_src	4	5	6	-	-	-	-	-	S56[7:6]	G10[7]	-	-
cif	226	clk_cifout	225	0	-	-	-	-	-	-	S56[5]	-	S56[4:0]	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
gic	228	aclk_gic_pre	4	5	-	-	-	-	-	-	S56[15]	G12[12]	S56[12:8]	-
alive	230	pclk_alive_gpll_src	5	-	-	-	-	-	-	-	-	PMUGRF0[6]	-	-
alive	231	pclk_alive	230	-	-	-	-	-	-	-	-	-	S57[4:0]	-
testout	234	clk_test_frac	4	5	-	-	-	-	-	-	S58[7]	G13[9]	-	S105
testout	235	clk_test_24m	0	-	-	-	-	-	-	-	-	-	S57[15:6]	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
spi	240	clk_spi0	4	5	-	-	-	-	-	-	S59[7]	G9[12]	S59[6:0]	-
spi	241	clk_spi1	4	5	-	-	-	-	-	-	S59[15]	G9[13]	S59[14:8]	-
spi	242	clk_spi2	4	5	-	-	-	-	-	-	S60[7]	G9[14]	S60[6:0]	-
spi	243	clk_spi4	4	5	-	-	-	-	-	-	S60[15]	G9[15]	S60[14:8]	-
spi	244	clk_spi5	4	5	-	-	-	-	-	-	S58[15]	G13[13]	S58[14:8]	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
i2c	250	clk_i2c1	4	5	-	-	-	-	-	-	S61[7]	G10[0]	S61[6:0]	-
i2c	252	clk_i2c2	4	5	-	-	-	-	-	-	S62[7]	G10[2]	S62[6:0]	-
i2c	254	clk_i2c3	4	5	-	-	-	-	-	-	S63[7]	G10[4]	S63[6:0]	-
i2c	251	clk_i2c5	4	5	-	-	-	-	-	-	S61[15]	G10[1]	S61[14:8]	-
i2c	253	clk_i2c6	4	5	-	-	-	-	-	-	S62[15]	G10[3]	S62[14:8]	-
i2c	255	clk_i2c7	4	5	-	-	-	-	-	-	S63[15]	G10[5]	S63[14:8]	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
alive	260	clk_mipidphy_ref	0	-	-	-	-	-	-	-	-	G11[14]	-	-
alive	261	clk_mipidphy_cfg	0	-	-	-	-	-	-	-	-	G11[15]	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
timer	340	clk_timer0	0	-	-	-	-	-	-	-	-	G26[0]	-	-

	clk		parents (ID)											
MODULE	ID	CLKNAME	0	1	2	3	4	5	6	7	MUX	GATE	DIV	FRAC
timer	341	clk_timer1	0	-	-	-	-	-	-	-	-	G26[1]	-	-
timer	342	clk_timer2	0	-	-	-	-	-	-	-	-	G26[2]	-	-
timer	343	clk_timer3	0	-	-	-	-	-	-	-	-	G26[3]	-	-
timer	344	clk_timer4	0	-	-	-	-	-	-	-	-	G26[4]	-	-
timer	345	clk_timer5	0	-	-	-	-	-	-	-	-	G26[5]	-	-
timer	346	clk_timer6	0	-	-	-	-	-	-	-	-	G26[6]	-	-
timer	347	clk_timer7	0	-	-	-	-	-	-	-	-	G26[7]	-	-
timer	348	clk_timer8	0	-	-	-	-	-	-	-	-	G26[8]	-	-
timer	349	clk_timer9	0	-	-	-	-	-	-	-	-	G26[9]	-	-
timer	350	clk_timer10	0	-	-	-	-	-	-	-	-	G26[10]	-	-
timer	351	clk_timer11	0	-	-	-	-	-	-	-	-	G26[11]	-	-

	clk		parents (ID)																		
MODULE	ID	CLKNAME	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	MUX	GATE	DIV	FRAC
testout	232	clk_test	24	15	99	127	152	162	121	89	123	120	114	138	209	10	235	234	M[3:0]	G13[11]	S58[4:0]	-

Fig. 3-3 RK3399Pro CRU Clock Architecture Diagram

	clk		parents (ID)										
MODULE	ID	CLKNAME	0	1	2	3	4	5	6	7	MUX	GATE	DIV	FRAC
coreb	500	clk_dbg_pd_core_b	24	-	-	-	-	-	-	-	-	G14[1]	-	-
coreb	501	pclk_dbg_cxcs_pd_core_b	27	-	-	-	-	-	-	-	-	G14[2]	-	-
coreb	502	aclk_core_adb400_gic_2_core_b	24	-	-	-	-	-	-	-	-	G14[3]	-	-
coreb	503	aclk_core_adb400_core_b_2_gic	24	-	-	-	-	-	-	-	-	G14[4]	-	-
coreb	504	aclk_core_adb400_core_b_2_cci500	25	-	-	-	-	-	-	-	-	G14[5]	-	-
coreb	505	aclk_perf_core_b	25	-	-	-	-	-	-	-	-	G14[6]	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
corel	508	clk_dbg_pd_core_l	15	-	-	-	-	-	-	-	-	G14[9]	-	-
corel	509	aclk_core_adb400_gic_2_core_l	15	-	-	-	-	-	-	-	-	G14[10]	-	-

	clk		parents (ID)											
MODULE	ID	CLKNAME	0	1	2	3	4	5	6	7	MUX	GATE	DIV	FRAC
corel	510	aclk_core_adb400_core_l_2_gic	15	-	-	-	-	-	-	-	-	G14[11]	-	-
corel	511	aclk_core_adb400_core_l_2_cci500	16	-	-	-	-	-	-	-	-	G14[12]	-	-
corel	512	aclk_perf_core_l	16	-	-	-	-	-	-	-	-	G14[13]	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
cci	515	aclk_adb400m_pd_core_l	99	-	-	-	-	-	-	-	-	G15[0]	-	-
cci	516	aclk_adb400m_pd_core_b	99	-	-	-	-	-	-	-	-	G15[1]	-	-
cci	517	aclk_cci	99	-	-	-	-	-	-	-	-	G15[2]	-	-
cci	518	aclk_cci_noc0	99	-	-	-	-	-	-	-	-	G15[3]	-	-
cci	519	aclk_cci_noc1	99	-	-	-	-	-	-	-	-	G15[4]	-	-
cci	520	clk_dbg_cxcs	106	-	-	-	-	-	-	-	-	G15[5]	-	-
cci	521	clk_dbg_noc	106	-	-	-	-	-	-	-	-	G15[6]	-	-
cci	522	aclk_cci_grf	99	-	-	-	-	-	-	-	-	G15[7]	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
iep	525	aclk_iep	116	-	-	-	-	-	-	-	-	G16[0]	-	-
iep	526	aclk_iep_noc	116	-	-	-	-	-	-	-	-	G16[1]	-	-
iep	527	hclk_iep	117	-	-	-	-	-	-	-	-	G16[2]	-	-
iep	528	hclk_iep_noc	117	-	-	-	-	-	-	-	-	G16[3]	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
rga	530	aclk_rga	118	-	-	-	-	-	-	-	-	G16[8]	-	-
rga	531	aclk_rga_noc	118	-	-	-	-	-	-	-	-	G16[9]	-	-
rga	532	hclk_rga	119	-	-	-	-	-	-	-	-	G16[10]	-	-
rga	533	hclk_rga_noc	119	-	-	-	-	-	-	-	-	G16[11]	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
vcodec	535	aclk_vcodec	110	-	-	-	-	-	-	-	-	G17[0]	-	-
vcodec	536	aclk_vcodec_noc	110	-	-	-	-	-	-	-	-	G17[1]	-	-
vcodec	537	hclk_vcodec	111	-	-	-	-	-	-	-	-	G17[2]	-	-
vcodec	538	hclk_vcodec_noc	111	-	-	-	-	-	-	-	-	G17[3]	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

	clk		parents (ID)											
MODULE	ID	CLKNAME	0	1	2	3	4	5	6	7	MUX	GATE	DIV	FRAC
vdu	540	aclk_vdu	112	-	-	-	-	-	-	-	-	G17[8]	-	-
vdu	541	aclk_vdu_noc	112	-	-	-	-	-	-	-	-	G17[9]	-	-
vdu	542	hclk_vdu	113	-	-	-	-	-	-	-	-	G17[10]	-	-
vdu	543	hclk_vdu_noc	113	-	-	-	-	-	-	-	-	G17[11]	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
ddrc	545	clk_ddr0_msch	90	-	-	-	-	-	-	-	-	G18[0]	-	-
ddrc	546	clk_ddrc0	90	-	-	-	-	-	-	-	-	G18[1]	-	-
ddrc	547	clk_ddrphy_ctrl0	90	-	-	-	-	-	-	-	-	G18[2]	-	-
ddrc	548	clk_ddrphy0	90	-	-	-	-	-	-	-	-	G18[3]	-	-
ddrc	549	clk_ddrcfg_msch0	90	-	-	-	-	-	-	-	-	G18[4]	-	-
ddrc	550	clk_ddr1_msch	90	-	-	-	-	-	-	-	-	G18[5]	-	-
ddrc	551	clk_ddrc1	90	-	-	-	-	-	-	-	-	G18[6]	-	-
ddrc	552	clk_ddrphy_ctrl1	90	-	-	-	-	-	-	-	-	G18[7]	-	-
ddrc	553	clk_ddrphy1	90	-	-	-	-	-	-	-	-	G18[8]	-	-
ddrc	554	clk_ddrcfg_msch1	90	-	-	-	-	-	-	-	-	G18[9]	-	-
ddrc	555	pclk_center_main_noc	91	-	-	-	-	-	-	-	-	G18[10]	-	-
ddrc	556	clk_ddr_cic	90	-	-	-	-	-	-	-	-	G18[11]	-	-
ddrc	557	pclk_ddr_mon	91	-	-	-	-	-	-	-	-	G18[12]	-	-
ddrc	558	clk_ddr_mon	90	-	-	-	-	-	-	-	-	G18[13]	-	-
ddrc	559	clk_ddr_mon_timer	0	-	-	-	-	-	-	-	-	G18[14]	-	-
ddrc	560	pclk_cic	91	-	-	-	-	-	-	-	-	G18[15]	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
center	562	aclk_center_main_noc	121	-	-	-	-	-	-	-	-	G19[0]	-	-
center	563	aclk_center_peri_noc	121	-	-	-	-	-	-	-	-	G19[1]	-	-
ddrc	564	pclk_ddr_sgrf	91	-	-	-	-	-	-	-	-	G19[2]	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
perihp	566	aclk_perf_pcie	127	-	-	-	-	-	-	-	-	G20[2]	-	-
perihp	567	pclk_perihp_grf	129	-	-	-	-	-	-	-	-	G20[4]	-	-

	clk		parents (ID)											
MODULE	ID	CLKNAME	0	1	2	3	4	5	6	7	MUX	GATE	DIV	FRAC
perihp	568	hclk_host0	128	-	-	-	-	-	-	-	-	G20[5]	-	-
perihp	569	hclk_host0_arb	128	-	-	-	-	-	-	-	-	G20[6]	-	-
perihp	570	hclk_host1	128	-	-	-	-	-	-	-	-	G20[7]	-	-
perihp	571	hclk_host1_arb	128	-	-	-	-	-	-	-	-	G20[8]	-	-
perihp	572	hclk_hsic	128	-	-	-	-	-	-	-	-	G20[9]	-	-
perihp	573	aclk_pcie	127	-	-	-	-	-	-	-	-	G20[10]	-	-
perihp	574	pclk_pcie	129	-	-	-	-	-	-	-	-	G20[11]	-	-
perihp	575	aclk_perihp_noc	127	-	-	-	-	-	-	-	-	G20[12]	-	-
perihp	576	hclk_perihp_noc	128	-	-	-	-	-	-	-	-	G20[13]	-	-
perihp	577	pclk_perihp_noc	129	-	-	-	-	-	-	-	-	G20[14]	-	-
perihp	578	hclk_ahb1tom	128	-	-	-	-	-	-	-	-	G20[15]	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
alive	581	clk_dphy_pll	260	-	-	-	-	-	-	-	-	G21[0]	-	-
alive	582	clk_dphy_tx0_cfg	261	-	-	-	-	-	-	-	-	G21[1]	-	-
alive	583	clk_dphy_tx1rx1_cfg	261	-	-	-	-	-	-	-	-	G21[2]	-	-
alive	584	clk_dphy_rx0_cfg	261	-	-	-	-	-	-	-	-	G21[3]	-	-
alive	585	pclk_uphy_mux_g	231	-	-	-	-	-	-	-	-	G21[4]	-	-
alive	586	pclk_uphy0_tcphy_g	231	-	-	-	-	-	-	-	-	G21[5]	-	-
alive	587	pclk_uphy0_tcpd_g	231	-	-	-	-	-	-	-	-	G21[6]	-	-
alive	588	pclk_uphy1_tcphy_g	231	-	-	-	-	-	-	-	-	G21[8]	-	-
alive	589	pclk_uphy1_tcpd_g	231	-	-	-	-	-	-	-	-	G21[9]	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
perilp1	591	pclk_uart0	163	-	-	-	-	-	-	-	-	G22[0]	-	-
perilp1	592	pclk_uart1	163	-	-	-	-	-	-	-	-	G22[1]	-	-
perilp1	593	pclk_uart2	163	-	-	-	-	-	-	-	-	G22[2]	-	-
perilp1	594	pclk_uart3	163	-	-	-	-	-	-	-	-	G22[3]	-	-
perilp1	595	pclk_rki2c7	163	-	-	-	-	-	-	-	-	G22[5]	-	-
perilp1	596	pclk_rki2c1	163	-	-	-	-	-	-	-	-	G22[6]	-	-

	clk		parents (ID)											
MODULE	ID	CLKNAME	0	1	2	3	4	5	6	7	MUX	GATE	DIV	FRAC
perilp1	597	pclk_rki2c5	163	-	-	-	-	-	-	-	-	G22[7]	-	-
perilp1	598	pclk_rki2c6	163	-	-	-	-	-	-	-	-	G22[8]	-	-
perilp1	599	pclk_rki2c2	163	-	-	-	-	-	-	-	-	G22[9]	-	-
perilp1	600	pclk_rki2c3	163	-	-	-	-	-	-	-	-	G22[10]	-	-
perilp1	601	pclk_mailbox0	163	-	-	-	-	-	-	-	-	G22[11]	-	-
perilp1	602	pclk_saradc	163	-	-	-	-	-	-	-	-	G22[12]	-	-
perilp1	603	pclk_tsadc	163	-	-	-	-	-	-	-	-	G22[13]	-	-
perilp1	604	pclk_efuse1024ns	163	-	-	-	-	-	-	-	-	G22[14]	-	-
perilp1	605	pclk_efuse1024s	163	-	-	-	-	-	-	-	-	G22[15]	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
perilp0	607	aclk_intmem	152	-	-	-	-	-	-	-	-	G23[0]	-	-
perilp0	608	aclk_tzma	152	-	-	-	-	-	-	-	-	G23[1]	-	-
perilp0	609	clk_intmem0	152	-	-	-	-	-	-	-	-	G23[2]	-	-
perilp0	610	clk_intmem1	152	-	-	-	-	-	-	-	-	G23[3]	-	-
perilp0	611	clk_intmem2	152	-	-	-	-	-	-	-	-	G23[4]	-	-
perilp0	612	clk_intmem3	152	-	-	-	-	-	-	-	-	G23[5]	-	-
perilp0	613	clk_intmem4	152	-	-	-	-	-	-	-	-	G23[6]	-	-
perilp0	614	clk_intmem5	152	-	-	-	-	-	-	-	-	G23[7]	-	-
perilp0	615	aclk_dcf	152	-	-	-	-	-	-	-	-	G23[8]	-	-
perilp0	616	pclk_dcf	154	-	-	-	-	-	-	-	-	G23[9]	-	-
perilp1	617	pclk_spi0	163	-	-	-	-	-	-	-	-	G23[10]	-	-
perilp1	618	pclk_spi1	163	-	-	-	-	-	-	-	-	G23[11]	-	-
perilp1	619	pclk_spi2	163	-	-	-	-	-	-	-	-	G23[12]	-	-
perilp1	620	pclk_spi4	163	-	-	-	-	-	-	-	-	G23[13]	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
perilp0	622	hclk_rom	153	-	-	-	-	-	-	-	-	G24[4]	-	-
perilp0	623	hclk_m_crypto0	153	-	-	-	-	-	-	-	-	G24[5]	-	-
perilp0	624	hclk_s_crypto0	153	-	-	-	-	-	-	-	-	G24[6]	-	-

	clk		parents (ID)											
MODULE	ID	CLKNAME	0	1	2	3	4	5	6	7	MUX	GATE	DIV	FRAC
cm0s_perilp	625	sclk_m0_perilp	159	-	-	-	-	-	-	-	-	G24[8]	-	-
cm0s_perilp	626	hclk_m0_perilp	159	-	-	-	-	-	-	-	-	G24[9]	-	-
cm0s_perilp	627	dclk_m0_perilp	159	-	-	-	-	-	-	-	-	G24[10]	-	-
cm0s_perilp	628	clk_m0_perilp_dec	159	-	-	-	-	-	-	-	-	G24[11]	-	-
perilp1	629	pclk_perilp_sgrf	163	-	-	-	-	-	-	-	-	G24[13]	-	-
perilp0	630	hclk_m_crypto1	153	-	-	-	-	-	-	-	-	G24[14]	-	-
perilp0	631	hclk_s_crypto1	153	-	-	-	-	-	-	-	-	G24[15]	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
perilp0	634	aclk_dmac0_perilp	152	-	-	-	-	-	-	-	-	G25[5]	-	-
perilp0	635	aclk_dmac1_perilp	152	-	-	-	-	-	-	-	-	G25[6]	-	-
perilp0	636	aclk_perilp0_noc	152	-	-	-	-	-	-	-	-	G25[7]	-	-
perilp0	637	hclk_perilp0_noc	153	-	-	-	-	-	-	-	-	G25[8]	-	-
perilp1	638	hclk_perilp1_noc	162	-	-	-	-	-	-	-	-	G25[9]	-	-
perilp1	639	pclk_perilp1_noc	163	-	-	-	-	-	-	-	-	G25[10]	-	-
cm0s_perilp	640	hclk_m0_perilp_noc	159	-	-	-	-	-	-	-	-	G25[11]	-	-
perilp1	641	hclk_sdio_noc	162	-	-	-	-	-	-	-	-	G25[12]	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
isp	643	hclk_isp0_noc	218	-	-	-	-	-	-	-	-	G27[0]	-	-
isp	644	aclk_isp0_noc	217	-	-	-	-	-	-	-	-	G27[1]	-	-
isp	645	hclk_isp1_noc	222	-	-	-	-	-	-	-	-	G27[2]	-	-
isp	646	aclk_isp1_noc	221	-	-	-	-	-	-	-	-	G27[3]	-	-
isp	647	hclk_isp0_wrapper	218	-	-	-	-	-	-	-	-	G27[4]	-	-
isp	648	aclk_isp0_wrapper	217	-	-	-	-	-	-	-	-	G27[5]	-	-
isp	649	pclkin_isp1_wrapper	357	-	-	-	-	-	-	-	-	G27[6]	-	-
isp	650	hclk_isp1_wrapper	217	-	-	-	-	-	-	-	-	G27[7]	-	-
isp	651	aclk_isp1_wrapper	222	-	-	-	-	-	-	-	-	G27[8]	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
vop0	653	hclk_vop0_noc	204	-	-	-	-	-	-	-	-	G28[0]	-	-

	clk		parents (ID)											
MODULE	ID	CLKNAME	0	1	2	3	4	5	6	7	MUX	GATE	DIV	FRAC
vop0	654	aclk_vop0_noc	203	-	-	-	-	-	-	-	-	G28[1]	-	-
vop0	655	hclk_vop0	204	-	-	-	-	-	-	-	-	G28[2]	-	-
vop0	656	aclk_vop0	203	-	-	-	-	-	-	-	-	G28[3]	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
vop1	658	hclk_vop1_noc	206	-	-	-	-	-	-	-	-	G28[4]	-	-
vop1	659	aclk_vop1_noc	205	-	-	-	-	-	-	-	-	G28[5]	-	-
vop1	660	hclk_vop1	206	-	-	-	-	-	-	-	-	G28[6]	-	-
vop1	661	aclk_vop1	205	-	-	-	-	-	-	-	-	G28[7]	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
vio	663	aclk_vio_noc	190	-	-	-	-	-	-	-	-	G29[0]	-	-
vio	664	pclk_mipi_dsi0	191	-	-	-	-	-	-	-	-	G29[1]	-	-
vio	665	pclk_mipi_dsi1	191	-	-	-	-	-	-	-	-	G29[2]	-	-
hdcp	666	pclk_hdcp_noc	195	-	-	-	-	-	-	-	-	G29[3]	-	-
hdcp	667	aclk_hdcp_noc	193	-	-	-	-	-	-	-	-	G29[4]	-	-
hdcp	668	hclk_hdcp_noc	194	-	-	-	-	-	-	-	-	G29[5]	-	-
hdcp	669	pclk_hdmi_ctrl	195	-	-	-	-	-	-	-	-	G29[6]	-	-
hdcp	670	pclk_dp_ctrl	195	-	-	-	-	-	-	-	-	G29[7]	-	-
hdcp	671	pclk_hdcp22	195	-	-	-	-	-	-	-	-	G29[8]	-	-
hdcp	672	hclk_hdcp22	194	-	-	-	-	-	-	-	-	G29[9]	-	-
hdcp	673	aclk_hdcp22	193	-	-	-	-	-	-	-	-	G29[10]	-	-
hdcp	674	pclk_gasket	195	-	-	-	-	-	-	-	-	G29[11]	-	-
vio	675	pclk_vio_grf	191	-	-	-	-	-	-	-	-	G29[12]	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
usbphy	677	aclk_usb3_noc	180	-	-	-	-	-	-	-	-	G30[0]	-	-
usbphy	678	aclk_usb3otg0	180	-	-	-	-	-	-	-	-	G30[1]	-	-
usbphy	679	aclk_usb3otg1	180	-	-	-	-	-	-	-	-	G30[2]	-	-
usbphy	680	aclk_usb3_rksoc_axi_perf	180	-	-	-	-	-	-	-	-	G30[3]	-	-
usbphy	681	aclk_usb3_grf	180	-	-	-	-	-	-	-	-	G30[4]	-	-

	clk		parents (ID)											
MODULE	ID	CLKNAME	0	1	2	3	4	5	6	7	MUX	GATE	DIV	FRAC
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
gpu	683	aclk_gpu	123	-	-	-	-	-	-	-	-	G30[8]	-	-
gpu	684	aclk_perf_gpu	123	-	-	-	-	-	-	-	-	G30[10]	-	-
gpu	685	aclk_gpu_grf	123	-	-	-	-	-	-	-	-	G30[11]	-	-
-	-	-		-	-	-	-	-	-	-	-	-	-	-
alive	687	pclk_grf	231	-	-	-	-	-	-	-	-	G31[1]	-	-
alive	688	pclk_intr_arb	231	-	-	-	-	-	-	-	-	G31[2]	-	-
alive	689	pclk_gpio2	231	-	-	-	-	-	-	-	-	G31[3]	-	-
alive	690	pclk_gpio3	231	-	-	-	-	-	-	-	-	G31[4]	-	-
alive	691	pclk_gpio4	231	-	-	-	-	-	-	-	-	G31[5]	-	-
alive	692	pclk_timer0	231	-	-	-	-	-	-	-	-	G31[6]	-	-
alive	693	pclk_timer1	231	-	-	-	-	-	-	-	-	G31[7]	-	-
perihp	694	pclk_hsicphy	129	-	-	-	-	-	-	-	-	G31[8]	-	-
alive	695	pclk_pmu_intr_arb	231	-	-	-	-	-	-	-	-	G31[9]	-	-
alive	696	pclk_sgrf	231	-	-	-	-	-	-	-	-	G31[10]	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
gmac	698	aclk_gmac	32	-	-	-	-	-	-	-	-	G32[0]	-	-
gmac	699	aclk_gmac_noc	32	-	-	-	-	-	-	-	-	G32[1]	-	-
gmac	700	pclk_gmac	33	-	-	-	-	-	-	-	-	G32[2]	-	-
gmac	701	pclk_gmac_noc	33	-	-	-	-	-	-	-	-	G32[3]	-	-
gmac	702	aclk_perf_gmac	32	-	-	-	-	-	-	-	-	G32[4]	-	-
-	-	-		-	-	-	-	-	-	-	-	-	-	-
emmc	704	aclk_emmc_core	147	-	-	-	-	-	-	-	-	G32[8]	-	-
emmc	705	aclk_emmc_noc	147	-	-	-	-	-	-	-	-	G32[9]	-	-
emmc	706	aclk_emmc_grf	147	-	-	-	-	-	-	-	-	G32[10]	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
edp	708	pclk_edp_noc	197	-	-	-	-	-	-	-	-	G32[12]	-	-
edp	709	pclk_edp_ctrl	197	-	-	-	-	-	-	-	-	G32[13]	-	-

	clk		parents (ID)											
MODULE	ID	CLKNAME	0	1	2	3	4	5	6	7	MUX	GATE	DIV	FRAC
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
gic	711	aclk_gic	228	-	-	-	-	-	-	-	-	G33[0]	-	-
gic	712	aclk_gic_noc	228	-	-	-	-	-	-	-	-	G33[1]	-	-
gic	713	aclk_gic_adb400_core_l_2_gic	228	-	-	-	-	-	-	-	-	G33[2]	-	-
gic	714	aclk_gic_adb400_core_b_2_gic	228	-	-	-	-	-	-	-	-	G33[3]	-	-
gic	715	aclk_gic_adb400_gic_2_core_l	228	-	-	-	-	-	-	-	-	G33[4]	-	-
gic	716	aclk_gic_adb400_gic_2_core_b	228	-	-	-	-	-	-	-	-	G33[5]	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
sd	718	hclk_sdmmc	325	-	-	-	-	-	-	-	-	G33[8]	-	-
sd	719	hclk_sdmmc_noc	325	-	-	-	-	-	-	-	-	G33[9]	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
perilp1	721	hclk_i2s0	162	-	-	-	-	-	-	-	-	G34[0]	-	-
perilp1	722	hclk_i2s1	162	-	-	-	-	-	-	-	-	G34[1]	-	-
perilp1	723	hclk_i2s2	162	-	-	-	-	-	-	-	-	G34[2]	-	-
perilp1	724	hclk_spdif	162	-	-	-	-	-	-	-	-	G34[3]	-	-
perilp1	725	hclk_sdio	162	-	-	-	-	-	-	-	-	G34[4]	-	-
perilp1	726	pclk_spi5	162	-	-	-	-	-	-	-	-	G34[5]	-	-
perilp1	727	hclk_sdioaudio_noc	162	-	-	-	-	-	-	-	-	G34[6]	-	-

Note:

*S**	*CRU_CLKSEL_CON**
*G**	*CRU_GATE_CON**
*M**	*CRU_MISC_CON*
GRF5[3]	*rmii_clk_sel*
GRF5[5:4]	*gmac_clk_sel[1:0]*
GRF5[6]	*rmii_mode*
*GF_**	*glitch-free*
*ICG_**	*ICG DIV*
*F**	*Fixed_div=**

Fig. 3-4 RK3399Pro Clock Architecture Diagram-ipgating

3.4 System Reset Solution

The following diagram shows reset architecture.

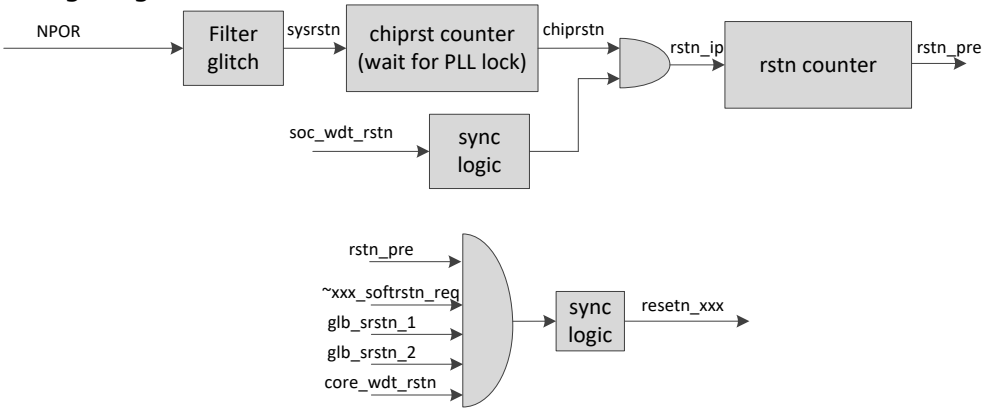


Fig. 3-5 Reset Architecture Diagram

Reset source of each reset signal includes hardware reset(NPOR), soc watch dog reset(soc_wdt_rstn), software reset request(xxx_softrstn_req), global software reset1(glb_srstn_1), global software reset2(glb_srstn_2) and core watch dog reset(core_wdt_rstn).

The 'xxx' of resetn_xxx and xxx_softrstn_req is the module name.

soc_wdt_rstn is the reset from watch-dog IP in the SoC, but core_wdt_rstn is the reset from core watch-dog block.

glb_srstn_1 and glb_srstn_2 are the global software reset by programming CRU register. When writing register CRU_GLB_SRST_FST_VALUE as 0xfdb9, glb_srstn_1 will be asserted, and when writing register CRU_GLB_SRST_SND_VALUE as 0xeca8, glb_srstn_2 will be asserted. The two software reset will be self-clear by hardware. glb_srstn_1 will reset the all logic, and glb_srstn_2 will reset the all logic except GRF and all GPIOs.

3.5 Function Description

There are eight PLLs in the chip: LPLL, BPLL, CPLL, GPLL, NPLL, VPLL, VPLL, PPLL, and it supports only one crystal oscillator: 24MHz. Each PLL can only receive 24MHz oscillator.

Eight PLLs all can be set to slow mode or deep slow mode, directly output selectable 24MHz. When power on or changing PLL setting, we must force PLL into slow mode to ensure output stable clock.

To maximize the flexibility, some of clocks can select divider source from eight PLLs.

To provide some specific frequency, another solution is integrated: fractional divider.

In order to be sure the performance for divided clock, there is some usage limit, we can only get low frequency and divider factor must be larger than 20.

All clocks can be software gated and all reset can be software generated.

3.6 PLL Introduction

3.6.1 Overview

The chip uses 3.2GHz VCO PLL for all eight PLLs. The 3.2GHz VCO PLL is a general purpose, high-performance PLL-based clock generator. The PLL is a multi-function, general purpose frequency synthesizer. Ultra-wide input and output ranges along with best-in-class jitter performance allow the PLL to be used for almost any clocking application. With excellent supply noise immunity, the PLL is ideal for use in noisy mixed signal SoC environments. By combining ultra-low jitter output clocks into a low power, low area, widely programmable design, It can greatly simplify a SoC by enabling a single macro to be used for all clocking applications in the system.

3.2GHz VCO PLL supports the following features:

- Input frequency range:1MHz to 1200MHz(Integer Mode) and 10MHz to 1200MHz (Fractional Mode)
- Output Frequency Range: 16MHz to 3.2GHz
- 24 bit fractional accuracy, and fractional mode jitter performance to nearly match

integer mode performance.
- 4:1 VCO frequency range allows PLL to be optimized for minimum jitter or minimum power.
- Isolated analog supply(1.8V) allows for excellent supply rejection in noisy SoC applications.
- Lock Detect Signal indicates when frequency lock has been achieved.

3.6.2 Block diagram

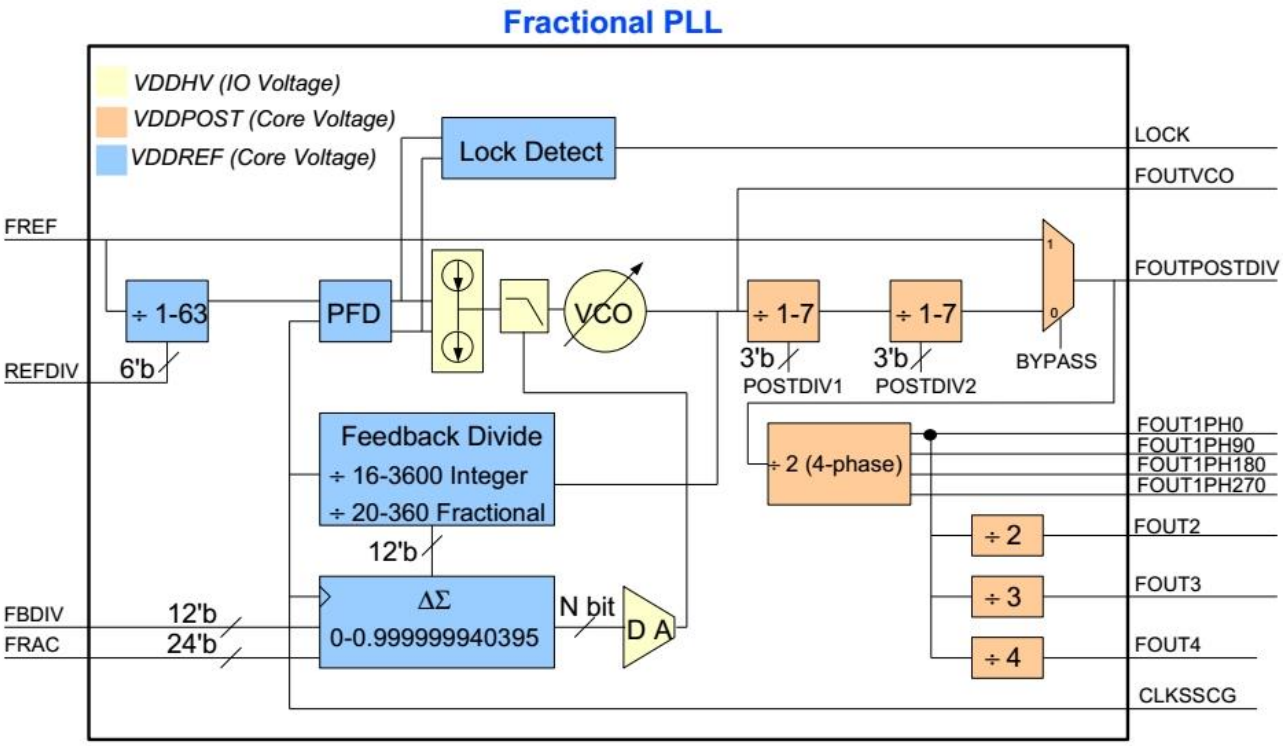


Fig. 3-6 PLL Block Diagram

How to calculate the PLL

The Fractional PLL output frequency can be calculated using some simple formulas. These formulas also embedded within the Fractional PLL Verilog model:
If DSMPD = 1 (DSM is disabled, "integer mode")
FOUTVCO = FREF / REFDIV * FBDIV
FOUTPOSTDIV = FOUTVCO / POSTDIV1 / POSTDIV2
If DSMPD = 0 (DSM is enabled, "fractional mode")
FOUTVCO = FREF / REFDIV * (FBDIV + FRAC / 224)
FOUTPOSTDIV = FOUTVCO / POSTDIV1 / POSTDIV2
Where:
FOUTVCO = Fractional PLL non-divided output frequency
FOUTPOSTDIV = Fractional PLL divided output frequency (output of second post divider)
FREF = Fractional PLL input reference frequency
REFDIV = Fractional PLL input reference clock divider
FVCO = Frequency of internal VCO
FBDIV = Integer value programmed into feedback divide
FRAC = Fractional value programmed into DSM

Changing the PLL Programming
In most cases the PLL programming can be changed on-the-fly and the PLL will simply slew to the new frequency. However, certain changes have the potential to cause glitches on the PLL output clocks. These changes include:
- Switching into or out of BYPASS mode may cause a glitch on FOUTPOSTDIV
- Changing POSTDIV1 or POSTDIV2 may cause a short pulse with width equal to as little as one VCO period on FOUTPOSTDIV

- Changing POSTDIV could cause a shortened pulse on FOUT1PH* or FOUT2/3/4
- Asserting PD or FOUTPOSTDIVPD may cause a glitch on FOUTPOSTDIV

3.7 Register Description

This section describes the control/status registers of the design.

3.7.1 Registers Summary-PMUCRU

Name	Offset	Size	Reset Value	Description
PMUCRU_PPLL_CON0	0x0000	W	0x000000a9	PPLL configuration register0
PMUCRU_PPLL_CON1	0x0004	W	0x00001203	PPLL configuration register1
PMUCRU_PPLL_CON2	0x0008	W	0x0000031f	PPLL configuration register2
PMUCRU_PPLL_CON3	0x000c	W	0x00000008	PPLL configuration register3
PMUCRU_PPLL_CON4	0x0010	W	0x00000007	PPLL configuration register4
PMUCRU_PPLL_CON5	0x0014	W	0x00007f00	PPLL configuration register5
PMUCRU_CLKSEL_CON0	0x0080	W	0x00000706	Internal clock select and divide register0
PMUCRU_CLKSEL_CON1	0x0084	W	0x00001986	Internal clock select and divide register1
PMUCRU_CLKSEL_CON2	0x0088	W	0x00000303	Internal clock select and divide register2
PMUCRU_CLKSEL_CON3	0x008c	W	0x00000003	Internal clock select and divide register3
PMUCRU_CLKSEL_CON4	0x0090	W	0x000002dc	Internal clock select and divide register4
PMUCRU_CLKSEL_CON5	0x0094	W	0x00000200	Internal clock select and divide register5
PMUCRU_CLKFRAC_CON0	0x0098	W	0x0bb8ea60	Internal clock select and divide register6
PMUCRU_CLKFRAC_CON1	0x009c	W	0x0bb8ea60	Internal clock select and divide register7
PMUCRU_CLKGATE_CON0	0x0100	W	0x00000000	Internal clock gating register0
PMUCRU_CLKGATE_CON1	0x0104	W	0x00000000	Internal clock gating register1
PMUCRU_CLKGATE_CON2	0x0108	W	0x00000000	Internal clock gating register2
PMUCRU_SOFTRST_CON0	0x0110	W	0x00000024	Internal software reset control register0
PMUCRU_SOFTRST_CON1	0x0114	W	0x00000000	Internal software reset control register1
PMUCRU_RSTNHOLD_CON0	0x0120	W	0x00000000	Internal reset hold control register0
PMUCRU_RSTNHOLD_CON1	0x0124	W	0x00000000	Internal reset hold control register1
PMUCRU_GATEDIS_CON0	0x0130	W	0x00000000	Internal gate disable control register0

*Notes:Size:**B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access*

3.7.2 Detail Register Description

PMUCRU_PPLL_CON0

Address: Operational Base + offset (0x0000)

PPLL configuration register0

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:12	RO	0x0	reserved
11:0	RW	0x0a9	fbdiv Feedback Divide Value Valid divider settings are: [16, 3200] in integer mode [20, 320] in fractional mode Tips: no plus one operation

PMUCRU_PPLL_CON1

Address: Operational Base + offset (0x0004)

PPLL configuration register1

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RO	0x0	reserved
14:12	RW	0x1	postdiv2 Second Post Divide Value (1-7)
11	RO	0x0	reserved
10:8	RW	0x2	postdiv1 First Post Divide Value (1-7)
7:6	RO	0x0	reserved
5:0	RW	0x03	refdiv Reference Clock Divide Value (1-63)

PMUCRU_PPLL_CON2

Address: Operational Base + offset (0x0008)

PPLL configuration register2

Bit	Attr	Reset Value	Description
31	RO	0x0	pll_lock PLL lock status 1'b0: unlock 1'b1: lock

Bit	Attr	Reset Value	Description
30:24	RO	0x0	reserved
23:0	RW	0x00031f	fracdiv Fractional part of feedback divide (fraction = FRAC/2^24)

PMUCRU_PPLL_CON3
Address: Operational Base + offset (0x000c)
PPLL configuration register3

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:10	RO	0x0	reserved
9:8	RW	0x0	pll_work_mode PLL work mode select 2'b00: Slow mode, clock from external 24MHz/26MHz OSC (default) 2'b01: Normal mode, clock from PLL output 2'b10: Deep slow mode, clock from external 32.768kHz
7	RO	0x0	reserved
6	RW	0x0	fout4phasepd Power down 4-phase clocks and 2X, 3X, 4X clocks 1'b0: no power down 1'b1: power down
5	RW	0x0	foutvcopd Power down buffered VCO clock 1'b0: no power down 1'b1: power down
4	RW	0x0	foutpostdivpd Power down all outputs except for buffered VCO clock 1'b0: no power down 1'b1: power down
3	RW	0x1	dsmpd PLL saturation behavior enable 1'b0: no power down 1'b1: power down DSMPD = 1'b1 (modulator is disabled, "integer mode")
2	RW	0x0	dacpd Power down quantization noise cancellation DAC 1'b0: no power down 1'b1: power down

Bit	Attr	Reset Value	Description
1	RW	0x0	bypass PLL Bypass. FREF bypasses PLL to FOUTPOSTDIV 1'b0: no bypass 1'b1: bypass
0	RW	0x0	power_down Global power down 1'b0: no power down 1'b1: power down

PMUCRU_PPLL_CON4
Address: Operational Base + offset (0x0010)
PPLL configuration register4

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:13	RO	0x0	reserved
12:8	RW	0x00	ssmod_spread spread amplitude % = 0.1 * SPREAD[4:0]
7:4	RW	0x0	ssmod_divval Divider required to set the modulation frequency Divider required to set the modulation frequency
3	RW	0x0	ssmod_downspread Selects center spread or downs pread 1'b0: center spread 1'b1: down spread
2	RW	0x1	ssmod_reset Reset modulator state 1'b0: no reset 1'b1: reset
1	RW	0x1	ssmod_disable_sscg Bypass SSMOD by module 1'b0: no bypass 1'b1: bypass
0	RW	0x1	ssmod_bp Bypass SSMOD by integration 1'b0: no bypass 1'b1: bypass

PMUCRU_PPLL_CON5
Address: Operational Base + offset (0x0014)
PPLL configuration register5

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:8	RW	0x7f	ssmod_ext_maxaddr External wave table data inputs (0-255)
7:1	RO	0x0	reserved
0	RW	0x0	ssmod_sel_ext_wave select external wave 1'b0: no select ext_wave 1'b1: select ext_wave

PMUCRU_CLKSEL_CON0
Address: Operational Base + offset (0x0080)
Internal clock select and divide register0

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	cm0s_clk_pll_sel cm0s_clk divider control register 1'b0:PPLL 1'b1:xin_24m
14:13	RO	0x0	reserved
12:8	RW	0x07	cm0s_div_con cm0s clock source select control register clk=clk_src/(div_con+1)
7:5	RO	0x0	reserved
4:0	RW	0x06	pmu_pclk_div_con pmu_pclk divider control register clk=clk_src/(div_con+1)

PMUCRU_CLKSEL_CON1
Address: Operational Base + offset (0x0084)
Internal clock select and divide register1

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit

Bit	Attr	Reset Value	Description
15	RW	0x0	clk_timer_sel clk_timer source select control register 1'b0:xin_24m 1'b1:clk_32k
14	RW	0x0	clk_wifi_sel clk_wifi source select control register 1'b0:clk_wifi_divout 1'b1:clk_wifi_frac
13	RW	0x0	clk_wifi_pll_sel clk_wifi_pll source select control register 1'b0:PPLL 1'b1:xin_24m
12:8	RW	0x19	clk_wifi_div_con clk_wifi divider control register clk=clk_src/(div_con+1)
7	RW	0x1	clk_spi3_pll_sel clk_spi3_pll source select control register 1'b0:xin_24m 1'b1:PPLL
6:0	RW	0x06	clk_spi3_div_con clk_spi3 divider control register clk=clk_src/(div_con+1)

PMUCRU_CLKSEL_CON2
Address: Operational Base + offset (0x0088)
Internal clock select and divide register2

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RO	0x0	reserved
14:8	RW	0x03	i2c8_div_con i2c8 divider control register clk=clk_src/(div_con+1)
7	RO	0x0	reserved
6:0	RW	0x03	i2c0_div_con i2c0 divider control register clk=clk_src/(div_con+1)

PMUCRU_CLKSEL_CON3
Address: Operational Base + offset (0x008c)
Internal clock select and divide register3

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:7	RO	0x0	reserved
6:0	RW	0x03	i2c4_div_con i2c4 divider control register clk=clk_src/(div_con+1)

PMUCRU_CLKSEL_CON4
Address: Operational Base + offset (0x0090)
Internal clock select and divide register4

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	clk_32k_suspend_sel clk_32k_suspend source select control register 1'b0:test clock out 1'b1:32k from pmu 24m div
14:10	RO	0x0	reserved
9:0	RW	0x2dc	clk_32k_suspend_div_con clk_32k_suspend divider control register clk=clk_src/(div_con+1)

PMUCRU_CLKSEL_CON5
Address: Operational Base + offset (0x0094)
Internal clock select and divide register5

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:11	RO	0x0	reserved
10	RW	0x0	clk_uart_pll_sel clk_uart_pll source select control register 1'b0:xin_24m 1'b1:PPLL
9:8	RW	0x2	uart4_clk_sel uart4_clk source select control register 2'b00:clk_uart4_divout 2'b01:clk_uart4_frac 2'b10:xin_24m
7	RO	0x0	reserved

Bit	Attr	Reset Value	Description
6:0	RW	0x00	uart4_div_con uart4 divider control register clk=clk_src/(div_con+1)

PMUCRU_CLKFRAC_CON0
Address: Operational Base + offset (0x0098)
Internal clock select and divide register6

Bit	Attr	Reset Value	Description
31:0	RW	0x0bb8ea60	uart4_frac_div_con uart4_frac divider control register Fout = Fsrc*numerator/denominator High 16-bit for numerator Low 16-bit for denominator

PMUCRU_CLKFRAC_CON1
Address: Operational Base + offset (0x009c)
Internal clock select and divide register7

Bit	Attr	Reset Value	Description
31:0	RW	0x0bb8ea60	wifi_frac_div_con wifi_frac divider control register Fout = Fsrc*numerator/denominator High 16-bit for numerator Low 16-bit for denominator

PMUCRU_CLKGATE_CON0
Address: Operational Base + offset (0x0100)
Internal clock gating register0

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:12	RO	0x0	reserved
11	RW	0x0	clk_i2c8_src_en clk_i2c8_src clock disable bit When HIGH, disable clock
10	RW	0x0	clk_i2c4_src_en clk_i2c4_src clock disable bit When HIGH, disable clock
9	RW	0x0	clk_i2c0_src_en clk_i2c0_src clock disable bit When HIGH, disable clock
8	RW	0x0	clk_wifi_en clk_wifi clock disable bit When HIGH, disable clock

Bit	Attr	Reset Value	Description
7	RW	0x0	clk_pvtm_pmu_en clk_pvtm_pmu clock disable bit When HIGH, disable clock
6	RW	0x0	clk_uart4_frac_src_en clk_uart4_frac_src clock disable bit When HIGH, disable clock
5	RW	0x0	clk_uart4_src_en clk_uart4_src clock disable bit When HIGH, disable clock
4	RW	0x0	clk_timer1_en clk_timer1 clock disable bit When HIGH, disable clock
3	RW	0x0	clk_timer0_en clk_timer0 clock disable bit When HIGH, disable clock
2	RW	0x0	clk_spi3_src_en clk_spi3_src clock disable bit When HIGH, disable clock
1	RW	0x0	fclk_cm0s_pmu_ppll_src_en fclk_cm0s_pmu_ppll_src clock disable bit When HIGH, disable clock
0	RO	0x0	reserved

PMUCRU_CLKGATE_CON1
Address: Operational Base + offset (0x0104)
Internal clock gating register1

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	pclk_wdt_m0_pmu_en pclk_wdt_m0_pmu clock disable bit When HIGH, disable clock
14	RW	0x0	pclk_uartm0_en pclk_uartm0 clock disable bit When HIGH, disable clock
13	RW	0x0	pclk_mailbox_pmu_en pclk_mailbox_pmu clock disable bit When HIGH, disable clock
12	RW	0x0	pclk_timer_pmu_en pclk_timer_pmu clock disable bit When HIGH, disable clock

Bit	Attr	Reset Value	Description
11	RW	0x0	pclk_spi3_en pclk_spi3 clock disable bit When HIGH, disable clock
10	RW	0x0	pclk_rkpwm_pmu_en pclk_rkpwm_pmu clock disable bit When HIGH, disable clock
9	RW	0x0	pclk_i2c8_en pclk_i2c8 clock disable bit When HIGH, disable clock
8	RW	0x0	pclk_i2c4_en pclk_i2c4 clock disable bit When HIGH, disable clock
7	RW	0x0	pclk_i2c0_en pclk_i2c0 clock disable bit When HIGH, disable clock
6	RW	0x0	pclk_noc_pmu_en pclk_noc_pmu clock disable bit When HIGH, disable clock Suggest always on
5	RW	0x0	pclk_sgrf_en pclk_sgrf clock disable bit When HIGH, disable clock Suggest always on
4	RW	0x0	pclk_gpio1_en pclk_gpio1 clock disable bit When HIGH, disable clock
3	RW	0x0	pclk_gpio0_en pclk_gpio0 clock disable bit When HIGH, disable clock
2	RW	0x0	pclk_intmem1_en pclk_intmem1 clock disable bit When HIGH, disable clock
1	RW	0x0	pclk_pmugrf_en pclk_pmugrf clock disable bit When HIGH, disable clock Suggest always on
0	RW	0x0	pclk_pmu_en pclk_pmu clock disable bit When HIGH, disable clock

PMUCRU_CLKGATE_CON2
Address: Operational Base + offset (0x0108)
Internal clock gating register2

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:6	RO	0x0	reserved
5	RW	0x0	hclk_noc_pmu_en hclk_noc_pmu clock disable bit When HIGH, disable clock Suggest always on
4	RO	0x0	reserved
3	RW	0x0	dclk_cm0s_en dclk_cm0s clock disable bit When HIGH, disable clock
2	RW	0x0	hclk_cm0s_en hclk_cm0s clock disable bit When HIGH, disable clock
1	RW	0x0	sclk_cm0s_en sclk_cm0s clock disable bit When HIGH, disable clock
0	RW	0x0	fclk_cm0s_en fclk_cm0s clock disable bit When HIGH, disable clock

PMUCRU_SOFTRST_CON0

Address: Operational Base + offset (0x0110)

Internal software reset control register0

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:14	RO	0x0	reserved
13	RW	0x0	presetn_wdt_pmu_req presetn_wdt_pmu request bit When HIGH, reset relative logic
12	RW	0x0	resetn_uart_m0_pmu_req resetn_uart_m0_pmu request bit When HIGH, reset relative logic
11	RW	0x0	presetn_uart_m0_pmu_req presetn_uart_m0_pmu request bit When HIGH, reset relative logic
10	RW	0x0	resetn_timer_pmu_1_req resetn_timer_pmu_1 request bit When HIGH, reset relative logic

Bit	Attr	Reset Value	Description
9	RW	0x0	resetn_timer_pmu_0_req resetn_timer_pmu_0 request bit When HIGH, reset relative logic
8	RW	0x0	presetn_timer_pmu_0_1_req presetn_timer_pmu_0_1 request bit When HIGH, reset relative logic
7	RW	0x0	resetn_spi3_req resetn_spi3 request bit When HIGH, reset relative logic
6	RW	0x0	presetn_spi3_req presetn_spi3 request bit When HIGH, reset relative logic
5	RW	0x1	poresetn_cm0s_pmu_req poresetn_cm0s_pmu request bit When HIGH, reset relative logic
4	RW	0x0	dbgresetn_cm0s_pmu_req dbgresetn_cm0s_pmu request bit When HIGH, reset relative logic
3	RW	0x0	hresetn_cm0s_noc_pmu_req hresetn_cm0s_noc_pmu request bit When HIGH, reset relative logic
2	RW	0x1	hresetn_cm0s_pmu_req hresetn_cm0s_pmu request bit When HIGH, reset relative logic
1	RW	0x0	presetn_intmem_pmu_req presetn_intmem_pmu request bit When HIGH, reset relative logic
0	RW	0x0	presetn_noc_pmu_req presetn_noc_pmu request bit When HIGH, reset relative logic

PMUCRU_SOFTRST_CON1

Address: Operational Base + offset (0x0114)

Internal software reset control register1

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RO	0x0	reserved
14	RW	0x0	resetn_i2c8_req resetn_i2c8 request bit When HIGH, reset relative logic

Bit	Attr	Reset Value	Description
13	RW	0x0	resetn_i2c4_req resetn_i2c4 request bit When HIGH, reset relative logic
12	RW	0x0	resetn_i2c0_req resetn_i2c0 request bit When HIGH, reset relative logic
11	RW	0x0	resetn_pvtm_pmu_req resetn_pvtm_pmu request bit When HIGH, reset relative logic
10	RW	0x0	presetn_intr_arb_req presetn_intr_arb request bit When HIGH, reset relative logic
9	RW	0x0	presetn_cru_pmu_req presetn_cru_pmu request bit When HIGH, reset relative logic
8	RW	0x0	presetn_gpio1_req presetn_gpio1 request bit When HIGH, reset relative logic
7	RW	0x0	presetn_gpio0_req presetn_gpio0 request bit When HIGH, reset relative logic
6	RW	0x0	presetn_sgrf_req presetn_sgrf request bit When HIGH, reset relative logic
5	RW	0x0	presetn_pmugrf_req presetn_pmugrf request bit When HIGH, reset relative logic
4	RW	0x0	presetn_rkpwm_pmu_req presetn_rkpwm_pmu request bit When HIGH, reset relative logic
3	RW	0x0	presetn_mailbox_pmu_req presetn_mailbox_pmu request bit When HIGH, reset relative logic
2	RW	0x0	presetn_i2c8_req presetn_i2c8 request bit When HIGH, reset relative logic
1	RW	0x0	presetn_i2c4_req presetn_i2c4 request bit When HIGH, reset relative logic
0	RW	0x0	presetn_i2c0_req presetn_i2c0 request bit When HIGH, reset relative logic

PMUCRU_RSTNHOLD_CON0
Address: Operational Base + offset (0x0120)

Internal reset hold control register0

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RO	0x0	reserved
14	RW	0x0	presetn_wdt_pmu_hold presetn_wdt_pmu_hold control bit When HIGH, reset hold, can't be reset by any reset source
13	RW	0x0	resetn_uart_m0_pmu_hold resetn_uart_m0_pmu_hold control bit When HIGH, reset hold, can't be reset by any reset source
12	RW	0x0	presetn_uart_m0_pmu_hold presetn_uart_m0_pmu_hold control bit When HIGH, reset hold, can't be reset by any reset source
11	RW	0x0	resetn_timer_pmu_1_hold resetn_timer_pmu_1_hold control bit When HIGH, reset hold, can't be reset by any reset source
10	RW	0x0	resetn_timer_pmu_0_hold resetn_timer_pmu_0_hold control bit When HIGH, reset hold, can't be reset by any reset source
9	RW	0x0	presetn_timer_pmu_0_1_hold presetn_timer_pmu_0_1_hold control bit When HIGH, reset hold, can't be reset by any reset source
8	RW	0x0	resetn_spi3_hold resetn_spi3_hold control bit When HIGH, reset hold, can't be reset by any reset source
7	RW	0x0	presetn_spi3_hold presetn_spi3_hold control bit When HIGH, reset hold, can't be reset by any reset source
6	RW	0x0	poresetn_cm0s_pmu_hold poresetn_cm0s_pmu_hold control bit When HIGH, reset hold, can't be reset by any reset source
5	RW	0x0	dbgresetn_cm0s_pmu_hold dbgresetn_cm0s_pmu_hold control bit When HIGH, reset hold, can't be reset by any reset source
4	RW	0x0	hresetn_cm0s_noc_pmu_hold hresetn_cm0s_noc_pmu_hold control bit When HIGH, reset hold, can't be reset by any reset source
3	RW	0x0	hresetn_cm0s_pmu_hold hresetn_cm0s_pmu_hold control bit When HIGH, reset hold, can't be reset by any reset source
2	RW	0x0	presetn_intmem_pmu_hold presetn_intmem_pmu_hold control bit When HIGH, reset hold, can't be reset by any reset source

Bit	Attr	Reset Value	Description
1	RW	0x0	presetn_noc_pmu_hold presetn_noc_pmu_hold control bit When HIGH, reset hold, can't be reset by any reset source
0	RO	0x0	reserved

PMUCRU_RSTNHOLD_CON1
Address: Operational Base + offset (0x0124)
Internal reset hold control register1

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RO	0x0	reserved
14	RW	0x0	resetn_i2c8_hold resetn_i2c8_hold control bit When HIGH, reset hold, can't be reset by any reset source
13	RW	0x0	resetn_i2c4_hold resetn_i2c4_hold control bit When HIGH, reset hold, can't be reset by any reset source
12	RW	0x0	resetn_i2c0_hold resetn_i2c0_hold control bit When HIGH, reset hold, can't be reset by any reset source
11	RW	0x0	resetn_pvtm_pmu_hold resetn_pvtm_pmu_hold control bit When HIGH, reset hold, can't be reset by any reset source
10	RW	0x0	presetn_intr_arb_hold presetn_intr_arb_hold control bit When HIGH, reset hold, can't be reset by any reset source
9	RW	0x0	presetn_cru_pmu_hold presetn_cru_pmu_hold control bit When HIGH, reset hold, can't be reset by any reset source
8	RW	0x0	presetn_gpio1_hold presetn_gpio1_hold control bit When HIGH, reset hold, can't be reset by any reset source
7	RW	0x0	presetn_gpio0_hold presetn_gpio0_hold control bit When HIGH, reset hold, can't be reset by any reset source
6	RW	0x0	presetn_sgrf_hold presetn_sgrf_hold control bit When HIGH, reset hold, can't be reset by any reset source
5	RW	0x0	presetn_pmugrf_hold presetn_pmugrf_hold control bit When HIGH, reset hold, can't be reset by any reset source

Bit	Attr	Reset Value	Description
4	RW	0x0	presetn_rkpwm_pmu_hold presetn_rkpwm_pmu_hold control bit When HIGH, reset hold, can't be reset by any reset source
3	RW	0x0	presetn_mailbox_pmu_hold presetn_mailbox_pmu_hold control bit When HIGH, reset hold, can't be reset by any reset source
2	RW	0x0	presetn_i2c8_hold presetn_i2c8_hold control bit When HIGH, reset hold, can't be reset by any reset source
1	RW	0x0	presetn_i2c4_hold presetn_i2c4_hold control bit When HIGH, reset hold, can't be reset by any reset source
0	RW	0x0	presetn_i2c0_hold presetn_i2c0_hold control bit When HIGH, reset hold, can't be reset by any reset source

PMUCRU_GATEDIS_CON0

Address: Operational Base + offset (0x0130)

Internal gate disable control register0

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved
29	RW	0x0	clk_sdioaudio_gating_dis clk_sdioaudio gate disable bit When HIGH, gate disable, open all clocks power domain idle request needed
28	RW	0x0	clk_sd_gating_dis clk_sd gate disable bit When HIGH, gate disable, open all clocks power domain idle request needed
27	RW	0x0	clk_gic_gating_dis clk_gic gate disable bit When HIGH, gate disable, open all clocks power domain idle request needed
26	RW	0x0	clk_gpu_gating_dis clk_gpu gate disable bit When HIGH, gate disable, open all clocks power domain idle request needed
25	RW	0x0	clk_perilp_gating_dis clk_perilp gate disable bit When HIGH, gate disable, open all clocks power domain idle request needed
24	RW	0x0	clk_perihp_gating_dis clk_perihp gate disable bit When HIGH, gate disable, open all clocks power domain idle request needed

Bit	Attr	Reset Value	Description
23	RW	0x0	clk_vcodec_gating_dis clk_vcodec gate disable bit When HIGH, gate disable, open all clocks power domain idle request needed
22	RW	0x0	clk_vdu_gating_dis clk_vdu gate disable bit When HIGH, gate disable, open all clocks power domain idle request needed
21	RW	0x0	clk_rga_gating_dis clk_rga gate disable bit When HIGH, gate disable, open all clocks power domain idle request needed
20	RW	0x0	clk_iep_gating_dis clk_iep gate disable bit When HIGH, gate disable, open all clocks power domain idle request needed
19	RW	0x0	clk_vopb_gating_dis clk_vopb gate disable bit When HIGH, gate disable, open all clocks power domain idle request needed
18	RW	0x0	clk_vopl_gating_dis clk_vopl gate disable bit When HIGH, gate disable, open all clocks power domain idle request needed
17	RW	0x0	clk_isp0_gating_dis clk_isp0 gate disable bit When HIGH, gate disable, open all clocks power domain idle request needed
16	RW	0x0	clk_isp1_gating_dis clk_isp1 gate disable bit When HIGH, gate disable, open all clocks power domain idle request needed
15	RW	0x0	clk_hdcp_gating_dis clk_hdcp gate disable bit When HIGH, gate disable, open all clocks power domain idle request needed
14	RW	0x0	clk_usb3_gating_dis clk_usb3 gate disable bit When HIGH, gate disable, open all clocks power domain idle request needed
13	RW	0x0	clk_perilpm0_gating_dis clk_perilpm0 gate disable bit When HIGH, gate disable, open all clocks power domain idle request needed

Bit	Attr	Reset Value	Description
12	RW	0x0	clk_center_gating_dis clk_center gate disable bit When HIGH, gate disable, open all clocks power domain idle request needed
11	RW	0x0	clk_ccim0_gating_dis clk_ccim0 gate disable bit When HIGH, gate disable, open all clocks power domain idle request needed
10	RW	0x0	clk_ccim1_gating_dis clk_ccim1 gate disable bit When HIGH, gate disable, open all clocks power domain idle request needed
9	RW	0x0	clk_vio_gating_dis clk_vio gate disable bit When HIGH, gate disable, open all clocks power domain idle request needed
8	RW	0x0	clk_msch0_gating_dis clk_msch0 gate disable bit When HIGH, gate disable, open all clocks power domain idle request needed
7	RW	0x0	clk_msch1_gating_dis clk_msch1 gate disable bit When HIGH, gate disable, open all clocks power domain idle request needed
6	RW	0x0	clk_alive_gating_dis clk_alive gate disable bit When HIGH, gate disable, open all clocks power domain idle request needed
5	RW	0x0	clk_pmu_gating_dis clk_pmu gate disable bit When HIGH, gate disable, open all clocks power domain idle request needed
4	RW	0x0	clk_edp_gating_dis clk_edp gate disable bit When HIGH, gate disable, open all clocks power domain idle request needed
3	RW	0x0	clk_gmac_gating_dis clk_gmac gate disable bit When HIGH, gate disable, open all clocks power domain idle request needed
2	RW	0x0	clk_emmc_gating_dis clk_emmc gate disable bit When HIGH, gate disable, open all clocks power domain idle request needed

Bit	Attr	Reset Value	Description
1	RW	0x0	clk_center1_gating_dis clk_center1 gate disable bit When HIGH, gate disable, open all clocks power domain idle request needed
0	RW	0x0	clk_pmum0_gating_dis clk_pmum0 gate disable bit When HIGH, gate disable, open all clocks power domain idle request needed

3.7.3 Registers Summary-CRU

Name	Offset	Size	Reset Value	Description
CRU_LPLL_CON0	0x0000	W	0x00000096	LPLL configuration register0
CRU_LPLL_CON1	0x0004	W	0x00001202	LPLL configuration register1
CRU_LPLL_CON2	0x0008	W	0x0000031f	LPLL configuration register2
CRU_LPLL_CON3	0x000c	W	0x00000008	LPLL configuration register3
CRU_LPLL_CON4	0x0010	W	0x00000007	LPLL configuration register4
CRU_LPLL_CON5	0x0014	W	0x00007f00	LPLL configuration register5
CRU_BPLL_CON0	0x0020	W	0x00000064	BPLL configuration register0
CRU_BPLL_CON1	0x0024	W	0x00001201	BPLL configuration register1
CRU_BPLL_CON2	0x0028	W	0x0000031f	BPLL configuration register2
CRU_BPLL_CON3	0x002c	W	0x00000008	BPLL configuration register3
CRU_BPLL_CON4	0x0030	W	0x00000007	BPLL configuration register4
CRU_BPLL_CON5	0x0034	W	0x00007f00	BPLL configuration register5
CRU_DPLL_CON0	0x0040	W	0x00000064	DPLL configuration register0
CRU_DPLL_CON1	0x0044	W	0x00001301	DPLL configuration register1
CRU_DPLL_CON2	0x0048	W	0x0000031f	DPLL configuration register2
CRU_DPLL_CON3	0x004c	W	0x00000008	DPLL configuration register3
CRU_DPLL_CON4	0x0050	W	0x00000007	DPLL configuration register4
CRU_DPLL_CON5	0x0054	W	0x00007f00	DPLL configuration register5
CRU_CPLL_CON0	0x0060	W	0x000000c0	CPLL configuration register0
CRU_CPLL_CON1	0x0064	W	0x00001302	CPLL configuration register1
CRU_CPLL_CON2	0x0068	W	0x0000031f	CPLL configuration register2
CRU_CPLL_CON3	0x006c	W	0x00000008	CPLL configuration register3
CRU_CPLL_CON4	0x0070	W	0x00000007	CPLL configuration register4
CRU_CPLL_CON5	0x0074	W	0x00007f00	CPLL configuration register5
CRU_GPLL_CON0	0x0080	W	0x000000c6	GPLL configuration register0
CRU_GPLL_CON1	0x0084	W	0x00002202	GPLL configuration register1
CRU_GPLL_CON2	0x0088	W	0x0000031f	GPLL configuration register2
CRU_GPLL_CON3	0x008c	W	0x00000008	GPLL configuration register3
CRU_GPLL_CON4	0x0090	W	0x00000007	GPLL configuration register4
CRU_GPLL_CON5	0x0094	W	0x00007f00	GPLL configuration register5
CRU_NPLL_CON0	0x00a0	W	0x000000fa	NPLL configuration register0

Name	Offset	Size	Reset Value	Description
CRU_NPLL_CON1	0x00a4	W	0x00001203	NPLL configuration register1
CRU_NPLL_CON2	0x00a8	W	0x0000031f	NPLL configuration register2
CRU_NPLL_CON3	0x00ac	W	0x00000008	NPLL configuration register3
CRU_NPLL_CON4	0x00b0	W	0x00000007	NPLL configuration register4
CRU_NPLL_CON5	0x00b4	W	0x00007f00	NPLL configuration register5
CRU_VPLL_CON0	0x00c0	W	0x000000c6	VPLL configuration register0
CRU_VPLL_CON1	0x00c4	W	0x00001202	VPLL configuration register1
CRU_VPLL_CON2	0x00c8	W	0x0000031f	VPLL configuration register2
CRU_VPLL_CON3	0x00cc	W	0x00000008	VPLL configuration register3
CRU_VPLL_CON4	0x00d0	W	0x00000007	VPLL configuration register4
CRU_VPLL_CON5	0x00d4	W	0x00007f00	VPLL configuration register5
CRU_CLKSEL_CON0	0x0100	W	0x00000101	Internal clock select and divide register0
CRU_CLKSEL_CON1	0x0104	W	0x00000303	Internal clock select and divide register1
CRU_CLKSEL_CON2	0x0108	W	0x00000141	Internal clock select and divide register2
CRU_CLKSEL_CON3	0x010c	W	0x00006303	Internal clock select and divide register3
CRU_CLKSEL_CON4	0x0110	W	0x00000041	Internal clock select and divide register4
CRU_CLKSEL_CON5	0x0114	W	0x00008341	Internal clock select and divide register5
CRU_CLKSEL_CON6	0x0118	W	0x00000320	Internal clock select and divide register6
CRU_CLKSEL_CON7	0x011c	W	0x00000101	Internal clock select and divide register7
CRU_CLKSEL_CON8	0x0120	W	0x00000101	Internal clock select and divide register8
CRU_CLKSEL_CON9	0x0124	W	0x00004141	Internal clock select and divide register9
CRU_CLKSEL_CON10	0x0128	W	0x00000101	Internal clock select and divide register10
CRU_CLKSEL_CON11	0x012c	W	0x00000101	Internal clock select and divide register11
CRU_CLKSEL_CON12	0x0130	W	0x00000100	Internal clock select and divide register12
CRU_CLKSEL_CON13	0x0134	W	0x00000361	Internal clock select and divide register13
CRU_CLKSEL_CON14	0x0138	W	0x00003181	Internal clock select and divide register14
CRU_CLKSEL_CON15	0x013c	W	0x00000500	Internal clock select and divide register15

Name	Offset	Size	Reset Value	Description
CRU_CLKSEL_CON16	0x0140	W	0x00000500	Internal clock select and divide register16
CRU_CLKSEL_CON17	0x0144	W	0x00000300	Internal clock select and divide register17
CRU_CLKSEL_CON18	0x0148	W	0x00004a87	Internal clock select and divide register18
CRU_CLKSEL_CON19	0x014c	W	0x00000100	Internal clock select and divide register19
CRU_CLKSEL_CON20	0x0150	W	0x00009303	Internal clock select and divide register20
CRU_CLKSEL_CON21	0x0154	W	0x00000003	Internal clock select and divide register21
CRU_CLKSEL_CON22	0x0158	W	0x00000400	Internal clock select and divide register22
CRU_CLKSEL_CON23	0x015c	W	0x00003181	Internal clock select and divide register23
CRU_CLKSEL_CON24	0x0160	W	0x00008103	Internal clock select and divide register24
CRU_CLKSEL_CON25	0x0164	W	0x00000183	Internal clock select and divide register25
CRU_CLKSEL_CON26	0x0168	W	0x00000103	Internal clock select and divide register26
CRU_CLKSEL_CON27	0x016c	W	0x000002dc	Internal clock select and divide register27
CRU_CLKSEL_CON28	0x0170	W	0x00000300	Internal clock select and divide register28
CRU_CLKSEL_CON29	0x0174	W	0x00000300	Internal clock select and divide register29
CRU_CLKSEL_CON30	0x0178	W	0x00000300	Internal clock select and divide register30
CRU_CLKSEL_CON31	0x017c	W	0x00000000	Internal clock select and divide register31
CRU_CLKSEL_CON32	0x0180	W	0x00006300	Internal clock select and divide register32
CRU_CLKSEL_CON33	0x0184	W	0x00000200	Internal clock select and divide register33
CRU_CLKSEL_CON34	0x0188	W	0x00000200	Internal clock select and divide register34
CRU_CLKSEL_CON35	0x018c	W	0x00000200	Internal clock select and divide register35
CRU_CLKSEL_CON36	0x0190	W	0x00000200	Internal clock select and divide register36
CRU_CLKSEL_CON38	0x0198	W	0x00003f3f	Internal clock select and divide register38

Name	Offset	Size	Reset Value	Description
CRU_CLKSEL_CON39	0x019c	W	0x00000041	Internal clock select and divide register39
CRU_CLKSEL_CON40	0x01a0	W	0x00000000	Internal clock select and divide register40
CRU_CLKSEL_CON41	0x01a4	W	0x00000000	Internal clock select and divide register41
CRU_CLKSEL_CON42	0x01a8	W	0x00000101	Internal clock select and divide register42
CRU_CLKSEL_CON43	0x01ac	W	0x00000421	Internal clock select and divide register43
CRU_CLKSEL_CON44	0x01b0	W	0x00000700	Internal clock select and divide register44
CRU_CLKSEL_CON45	0x01b4	W	0x000002dc	Internal clock select and divide register45
CRU_CLKSEL_CON46	0x01b8	W	0x00000004	Internal clock select and divide register46
CRU_CLKSEL_CON47	0x01bc	W	0x00000102	Internal clock select and divide register47
CRU_CLKSEL_CON48	0x01c0	W	0x00000102	Internal clock select and divide register48
CRU_CLKSEL_CON49	0x01c4	W	0x00000001	Internal clock select and divide register49
CRU_CLKSEL_CON50	0x01c8	W	0x00000003	Internal clock select and divide register50
CRU_CLKSEL_CON51	0x01cc	W	0x00000005	Internal clock select and divide register51
CRU_CLKSEL_CON52	0x01d0	W	0x00000005	Internal clock select and divide register52
CRU_CLKSEL_CON53	0x01d4	W	0x00000101	Internal clock select and divide register53
CRU_CLKSEL_CON54	0x01d8	W	0x00000101	Internal clock select and divide register54
CRU_CLKSEL_CON55	0x01dc	W	0x00008181	Internal clock select and divide register55
CRU_CLKSEL_CON56	0x01e0	W	0x00000320	Internal clock select and divide register56
CRU_CLKSEL_CON57	0x01e4	W	0x00000005	Internal clock select and divide register57
CRU_CLKSEL_CON58	0x01e8	W	0x0000071f	Internal clock select and divide register58
CRU_CLKSEL_CON59	0x01ec	W	0x00000707	Internal clock select and divide register59
CRU_CLKSEL_CON60	0x01f0	W	0x00000707	Internal clock select and divide register60

Name	Offset	Size	Reset Value	Description
CRU_CLKSEL_CON61	0x01f4	W	0x00000303	Internal clock select and divide register61
CRU_CLKSEL_CON62	0x01f8	W	0x00000303	Internal clock select and divide register62
CRU_CLKSEL_CON63	0x01fc	W	0x00000303	Internal clock select and divide register63
CRU_CLKSEL_CON64	0x0200	W	0x000000c5	Internal clock select and divide register64
CRU_CLKSEL_CON65	0x0204	W	0x000000c5	Internal clock select and divide register65
CRU_CLKSEL_CON96	0x0280	W	0x0bb8ea60	Internal clock select and divide register80
CRU_CLKSEL_CON97	0x0284	W	0x0bb8ea60	Internal clock select and divide register81
CRU_CLKSEL_CON98	0x0288	W	0x0bb8ea60	Internal clock select and divide register82
CRU_CLKSEL_CON99	0x028c	W	0x0bb8ea60	Internal clock select and divide register83
CRU_CLKSEL_CON100	0x0290	W	0x0bb8ea60	Internal clock select and divide register84
CRU_CLKSEL_CON101	0x0294	W	0x0bb8ea60	Internal clock select and divide register85
CRU_CLKSEL_CON102	0x0298	W	0x0bb8ea60	Internal clock select and divide register86
CRU_CLKSEL_CON103	0x029c	W	0x0bb8ea60	Internal clock select and divide register87
CRU_CLKSEL_CON105	0x02a4	W	0x0bb8ea60	Internal clock select and divide register89
CRU_CLKSEL_CON106	0x02a8	W	0x0bb8ea60	Internal clock select and divide register90
CRU_CLKSEL_CON107	0x02ac	W	0x0bb8ea60	Internal clock select and divide register91
CRU_CLKGATE_CON0	0x0300	W	0x00000000	Internal clock gating register0
CRU_CLKGATE_CON1	0x0304	W	0x00000000	Internal clock gating register1
CRU_CLKGATE_CON2	0x0308	W	0x00000000	Internal clock gating register2
CRU_CLKGATE_CON3	0x030c	W	0x00000000	Internal clock gating register3
CRU_CLKGATE_CON4	0x0310	W	0x00000000	Internal clock gating register4
CRU_CLKGATE_CON5	0x0314	W	0x00000000	Internal clock gating register5
CRU_CLKGATE_CON6	0x0318	W	0x00000000	Internal clock gating register6
CRU_CLKGATE_CON7	0x031c	W	0x00000000	Internal clock gating register7
CRU_CLKGATE_CON8	0x0320	W	0x00000000	Internal clock gating register8
CRU_CLKGATE_CON9	0x0324	W	0x00000000	Internal clock gating register9
CRU_CLKGATE_CON10	0x0328	W	0x00000000	Internal clock gating register10

Name	Offset	Size	Reset Value	Description
CRU_CLKGATE_CON11	0x032c	W	0x00000000	Internal clock gating register11
CRU_CLKGATE_CON12	0x0330	W	0x00000000	Internal clock gating register12
CRU_CLKGATE_CON13	0x0334	W	0x00000000	Internal clock gating register13
CRU_CLKGATE_CON14	0x0338	W	0x00000000	Internal clock gating register14
CRU_CLKGATE_CON15	0x033c	W	0x00000000	Internal clock gating register15
CRU_CLKGATE_CON16	0x0340	W	0x00000000	Internal clock gating register16
CRU_CLKGATE_CON17	0x0344	W	0x00000000	Internal clock gating register17
CRU_CLKGATE_CON18	0x0348	W	0x00000000	Internal clock gating register18
CRU_CLKGATE_CON19	0x034c	W	0x00000000	Internal clock gating register19
CRU_CLKGATE_CON20	0x0350	W	0x00000000	Internal clock gating register20
CRU_CLKGATE_CON21	0x0354	W	0x00000000	Internal clock gating register21
CRU_CLKGATE_CON22	0x0358	W	0x00000000	Internal clock gating register22
CRU_CLKGATE_CON23	0x035c	W	0x00000000	Internal clock gating register23
CRU_CLKGATE_CON24	0x0360	W	0x00000000	Internal clock gating register24
CRU_CLKGATE_CON25	0x0364	W	0x00000000	Internal clock gating register25
CRU_CLKGATE_CON26	0x0368	W	0x00000000	Internal clock gating register26
CRU_CLKGATE_CON27	0x036c	W	0x00000000	Internal clock gating register27
CRU_CLKGATE_CON28	0x0370	W	0x00000000	Internal clock gating register28
CRU_CLKGATE_CON29	0x0374	W	0x00000000	Internal clock gating register29
CRU_CLKGATE_CON30	0x0378	W	0x00000000	Internal clock gating register30
CRU_CLKGATE_CON31	0x037c	W	0x00000000	Internal clock gating register31
CRU_CLKGATE_CON32	0x0380	W	0x00000000	Internal clock gating register32
CRU_CLKGATE_CON33	0x0384	W	0x00000000	Internal clock gating register33
CRU_CLKGATE_CON34	0x0388	W	0x00000000	Internal clock gating register34
CRU_SOFTRST_CON0	0x0400	W	0x00000000	Internal software reset control register0
CRU_SOFTRST_CON1	0x0404	W	0x00000000	Internal software reset control register1
CRU_SOFTRST_CON2	0x0408	W	0x00000000	Internal software reset control register2
CRU_SOFTRST_CON3	0x040c	W	0x00000010	Internal software reset control register3
CRU_SOFTRST_CON4	0x0410	W	0x00000000	Internal software reset control register4
CRU_SOFTRST_CON5	0x0414	W	0x00000000	Internal software reset control register5
CRU_SOFTRST_CON6	0x0418	W	0x00000000	Internal software reset control register6
CRU_SOFTRST_CON7	0x041c	W	0x00000000	Internal software reset control register7
CRU_SOFTRST_CON8	0x0420	W	0x000000bc	Internal software reset control register8

Name	Offset	Size	Reset Value	Description
CRU_SOFTRST_CON9	0x0424	W	0x00000000	Internal software reset control register9
CRU_SOFTRST_CON10	0x0428	W	0x00000000	Internal software reset control register10
CRU_SOFTRST_CON11	0x042c	W	0x00000014	Internal software reset control register11
CRU_SOFTRST_CON12	0x0430	W	0x00000000	Internal software reset control register12
CRU_SOFTRST_CON13	0x0434	W	0x00000000	Internal software reset control register13
CRU_SOFTRST_CON14	0x0438	W	0x00000000	Internal software reset control register14
CRU_SOFTRST_CON15	0x043c	W	0x00000000	Internal software reset control register15
CRU_SOFTRST_CON16	0x0440	W	0x00000000	Internal software reset control register16
CRU_SOFTRST_CON17	0x0444	W	0x00000000	Internal software reset control register17
CRU_SOFTRST_CON18	0x0448	W	0x00000000	Internal software reset control register18
CRU_SOFTRST_CON19	0x044c	W	0x00000000	Internal software reset control register19
CRU_SOFTRST_CON20	0x0450	W	0x00000000	Internal software reset control register20
CRU_GLB_SRST_FST_VALUE	0x0500	W	0x00000000	The first global software reset config value
CRU_GLB_SRST_SND_VALUE	0x0504	W	0x00000000	The second global software reset config value
CRU_GLB_CNT_TH	0x0508	W	0x00000000	Global soft reset counter threshold
CRU_MISC_CON	0x050c	W	0x00000000	Output clock selection for test
CRU_GLB_RST_CON	0x0510	W	0x00000000	Global reset trigger select
CRU_GLB_RST_ST	0x0514	W	0x00000000	Global reset status
CRU_SDMMC_CON0	0x0580	W	0x00000004	sdmmc control0
CRU_SDMMC_CON1	0x0584	W	0x00000000	sdmmc control1
CRU_SDIO0_CON0	0x0588	W	0x00000004	sdio0 control0
CRU_SDIO0_CON1	0x058c	W	0x00000000	sdio0 control1

*Notes:Size:**B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access*

3.7.4 Detail Register Description

CRU_LPLL_CON0

Address: Operational Base + offset (0x0000)

LPLL configuration register0

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:12	RO	0x0	reserved
11:0	RW	0x096	fbdiv Feedback Divide Value Valid divider settings are: [16, 3200] in integer mode [20, 320] in fractional mode Tips: no plus one operation

CRU_LPLL_CON1
Address: Operational Base + offset (0x0004)
LPLL configuration register1

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RO	0x0	reserved
14:12	RW	0x1	postdiv2 Second Post Divide Value (1-7)
11	RO	0x0	reserved
10:8	RW	0x2	postdiv1 First Post Divide Value (1-7)
7:6	RO	0x0	reserved
5:0	RW	0x02	refdiv Reference Clock Divide Value (1-63)

CRU_LPLL_CON2
Address: Operational Base + offset (0x0008)
LPLL configuration register2

Bit	Attr	Reset Value	Description
31	RO	0x0	pll_lock PLL lock status 1'b0: unlock 1'b1: lock
30:24	RO	0x0	reserved
23:0	RW	0x00031f	fracdiv Fractional part of feedback divide (fraction = FRAC/2^24)

CRU_LPLL_CON3
Address: Operational Base + offset (0x000c)
LPLL configuration register3

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:10	RO	0x0	reserved
9:8	RW	0x0	pll_work_mode PLL work mode select 2'b00: Slow mode, clock from external 24MHz/26MHz OSC (default) 2'b01: Normal mode, clock from PLL output 2'b10: Deep slow mode, clock from external 32.768kHz
7	RO	0x0	reserved
6	RW	0x0	fout4phasepd Power down 4-phase clocks and 2X, 3X, 4X clocks 1'b0: no power down 1'b1: power down
5	RW	0x0	foutvcopd Power down buffered VCO clock 1'b0: no power down 1'b1: power down
4	RW	0x0	foutpostdivpd Power down all outputs except for buffered VCO clock 1'b0: no power down 1'b1: power down
3	RW	0x1	dsmpd PLL saturation behavior enable 1'b0: no power down 1'b1: power down DSMPD = 1'b1 (modulator is disabled, "integer mode")
2	RW	0x0	dacpd Power down quantization noise cancellation DAC 1'b0: no power down 1'b1: power down
1	RW	0x0	bypass PLL Bypass. FREF bypasses PLL to FOUTPOSTDIV 1'b0: no bypass 1'b1: bypass
0	RW	0x0	power_down Global power down 1'b0: no power down 1'b1: power down

CRU_LPLL_CON4
Address: Operational Base + offset (0x0010)
LPLL configuration register4

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:13	RO	0x0	reserved
12:8	RW	0x00	ssmod_spread spread amplitude % = 0.1 * SPREAD[4:0]
7:4	RW	0x0	ssmod_divval Divider required to set the modulation frequency Divider required to set the modulation frequency
3	RW	0x0	ssmod_downspread Selects center spread or downs pread 1'b0: center spread 1'b1: down spread
2	RW	0x1	ssmod_reset Reset modulator state 1'b0: no reset 1'b1: reset
1	RW	0x1	ssmod_disable_sscg Bypass SSMOD by module 1'b0: no bypass 1'b1: bypass
0	RW	0x1	ssmod_bp Bypass SSMOD by integration 1'b0: no bypass 1'b1: bypass

CRU_LPLL_CON5
Address: Operational Base + offset (0x0014)
LPLL configuration register5

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:8	RW	0x7f	ssmod_ext_maxaddr External wave table data inputs (0-255)
7:1	RO	0x0	reserved

Bit	Attr	Reset Value	Description
0	RW	0x0	ssmod_sel_ext_wave select external wave 1'b0: no select ext_wave 1'b1: select ext_wave

CRU_BPLL_CON0
Address: Operational Base + offset (0x0020)
BPLL configuration register0

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:12	RO	0x0	reserved
11:0	RW	0x064	fbdiv Feedback Divide Value Valid divider settings are: [16, 3200] in integer mode [20, 320] in fractional mode Tips: no plus one operation

CRU_BPLL_CON1
Address: Operational Base + offset (0x0024)
BPLL configuration register1

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RO	0x0	reserved
14:12	RW	0x1	postdiv2 Second Post Divide Value (1-7)
11	RO	0x0	reserved
10:8	RW	0x2	postdiv1 First Post Divide Value (1-7)
7:6	RO	0x0	reserved
5:0	RW	0x01	refdiv Reference Clock Divide Value (1-63)

CRU_BPLL_CON2
Address: Operational Base + offset (0x0028)
BPLL configuration register2

Bit	Attr	Reset Value	Description
31	RO	0x0	pll_lock PLL lock status 1'b0: unlock 1'b1: lock
30:24	RO	0x0	reserved
23:0	RW	0x00031f	fracdiv Fractional part of feedback divide (fraction = FRAC/2^24)

CRU_BPLL_CON3
Address: Operational Base + offset (0x002c)
BPLL configuration register3

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:10	RO	0x0	reserved
9:8	RW	0x0	pll_work_mode PLL work mode select 2'b00: Slow mode, clock from external 24MHz/26MHz OSC (default) 2'b01: Normal mode, clock from PLL output 2'b10: Deep slow mode, clock from external 32.768kHz
7	RO	0x0	reserved
6	RW	0x0	fout4phasepd Power down 4-phase clocks and 2X, 3X, 4X clocks 1'b0: no power down 1'b1: power down
5	RW	0x0	foutvcopd Power down buffered VCO clock 1'b0: no power down 1'b1: power down
4	RW	0x0	foutpostdivpd Power down all outputs except for buffered VCO clock 1'b0: no power down 1'b1: power down
3	RW	0x1	dsmpd PLL saturation behavior enable 1'b0: no power down 1'b1: power down DSMPD = 1'b1 (modulator is disabled, "integer mode")

Bit	Attr	Reset Value	Description
2	RW	0x0	dacpd Power down quantization noise cancellation DAC 1'b0: no power down 1'b1: power down
1	RW	0x0	bypass PLL Bypass. FREF bypasses PLL to FOUTPOSTDIV 1'b0: no bypass 1'b1: bypass
0	RW	0x0	power_down Global power down 1'b0: no power down 1'b1: power down

CRU_BPLL_CON4
Address: Operational Base + offset (0x0030)
BPLL configuration register4

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:13	RO	0x0	reserved
12:8	RW	0x00	ssmod_spread spread amplitude % = 0.1 * SPREAD[4:0]
7:4	RW	0x0	ssmod_divval Divider required to set the modulation frequency Divider required to set the modulation frequency
3	RW	0x0	ssmod_downspread Selects center spread or downs pread 1'b0: center spread 1'b1: down spread
2	RW	0x1	ssmod_reset Reset modulator state 1'b0: no reset 1'b1: reset
1	RW	0x1	ssmod_disable_sscg Bypass SSMOD by module 1'b0: no bypass 1'b1: bypass
0	RW	0x1	ssmod_bp Bypass SSMOD by integration 1'b0: no bypass 1'b1: bypass

CRU_BPLL_CON5
Address: Operational Base + offset (0x0034)
BPLL configuration register5

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:8	RW	0x7f	ssmod_ext_maxaddr External wave table data inputs (0-255)
7:1	RO	0x0	reserved
0	RW	0x0	ssmod_sel_ext_wave select external wave 1'b0: no select ext_wave 1'b1: select ext_wave

CRU_DPLL_CON0
Address: Operational Base + offset (0x0040)
DPLL configuration register0

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:12	RO	0x0	reserved
11:0	RW	0x064	fbdiv Feedback Divide Value Valid divider settings are: [16, 3200] in integer mode [20, 320] in fractional mode Tips: no plus one operation

CRU_DPLL_CON1
Address: Operational Base + offset (0x0044)
DPLL configuration register1

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RO	0x0	reserved
14:12	RW	0x1	postdiv2 Second Post Divide Value (1-7)
11	RO	0x0	reserved

Bit	Attr	Reset Value	Description
10:8	RW	0x3	postdiv1 First Post Divide Value (1-7)
7:6	RO	0x0	reserved
5:0	RW	0x01	refdiv Reference Clock Divide Value (1-63)

CRU_DPLL_CON2
Address: Operational Base + offset (0x0048)
DPLL configuration register2

Bit	Attr	Reset Value	Description
31	RO	0x0	pll_lock PLL lock status 1'b0: unlock 1'b1: lock
30:24	RO	0x0	reserved
23:0	RW	0x00031f	fracdiv Fractional part of feedback divide (fraction = FRAC/2^24)

CRU_DPLL_CON3
Address: Operational Base + offset (0x004c)
DPLL configuration register3

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:10	RO	0x0	reserved
9:8	RW	0x0	pll_work_mode PLL work mode select 2'b00: Slow mode, clock from external 24MHz/26MHz OSC (default) 2'b01: Normal mode, clock from PLL output 2'b10: Deep slow mode, clock from external 32.768kHz
7	RO	0x0	reserved
6	RW	0x0	fout4phasepd Power down 4-phase clocks and 2X, 3X, 4X clocks 1'b0: no power down 1'b1: power down
5	RW	0x0	foutvcopd Power down buffered VCO clock 1'b0: no power down 1'b1: power down

Bit	Attr	Reset Value	Description
4	RW	0x0	foutpostdivpd Power down all outputs except for buffered VCO clock 1'b0: no power down 1'b1: power down
3	RW	0x1	dsmpd PLL saturation behavior enable 1'b0: no power down 1'b1: power down DSMPD = 1'b1 (modulator is disabled, "integer mode")
2	RW	0x0	dacpd Power down quantization noise cancellation DAC 1'b0: no power down 1'b1: power down
1	RW	0x0	bypass PLL Bypass. FREF bypasses PLL to FOUTPOSTDIV 1'b0: no bypass 1'b1: bypass
0	RW	0x0	power_down Global power down 1'b0: no power down 1'b1: power down

CRU_DPLL_CON4
Address: Operational Base + offset (0x0050)
DPLL configuration register4

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:13	RO	0x0	reserved
12:8	RW	0x00	ssmod_spread spread amplitude % = 0.1 * SPREAD[4:0]
7:4	RW	0x0	ssmod_divval Divider required to set the modulation frequency Divider required to set the modulation frequency
3	RW	0x0	ssmod_downspread Selects center spread or downs pread 1'b0: center spread 1'b1: down spread
2	RW	0x1	ssmod_reset Reset modulator state 1'b0: no reset 1'b1: reset

Bit	Attr	Reset Value	Description
1	RW	0x1	ssmod_disable_sscg Bypass SSMOD by module 1'b0: no bypass 1'b1: bypass
0	RW	0x1	ssmod_bp Bypass SSMOD by integration 1'b0: no bypass 1'b1: bypass

CRU_DPLL_CON5
Address: Operational Base + offset (0x0054)
DPLL configuration register5

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:8	RW	0x7f	ssmod_ext_maxaddr External wave table data inputs (0-255)
7:1	RO	0x0	reserved
0	RW	0x0	ssmod_sel_ext_wave select external wave 1'b0: no select ext_wave 1'b1: select ext_wave

CRU_CPLL_CON0
Address: Operational Base + offset (0x0060)
CPLL configuration register0

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:12	RO	0x0	reserved
11:0	RW	0x0c0	fbdiv Feedback Divide Value Valid divider settings are: [16, 3200] in integer mode [20, 320] in fractional mode Tips: no plus one operation

CRU_CPLL_CON1
Address: Operational Base + offset (0x0064)
CPLL configuration register1

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RO	0x0	reserved
14:12	RW	0x1	postdiv2 Second Post Divide Value (1-7)
11	RO	0x0	reserved
10:8	RW	0x3	postdiv1 First Post Divide Value (1-7)
7:6	RO	0x0	reserved
5:0	RW	0x02	refdiv Reference Clock Divide Value (1-63)

CRU_CPLL_CON2
Address: Operational Base + offset (0x0068)
CPLL configuration register2

Bit	Attr	Reset Value	Description
31	RO	0x0	pll_lock PLL lock status 1'b0: unlock 1'b1: lock
30:24	RO	0x0	reserved
23:0	RW	0x00031f	fracdiv Fractional part of feedback divide (fraction = FRAC/2^24)

CRU_CPLL_CON3
Address: Operational Base + offset (0x006c)
CPLL configuration register3

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:10	RO	0x0	reserved
9:8	RW	0x0	pll_work_mode PLL work mode select 2'b00: Slow mode, clock from external 24MHz/26MHz OSC (default) 2'b01: Normal mode, clock from PLL output 2'b10: Deep slow mode, clock from external 32.768kHz

Bit	Attr	Reset Value	Description
7	RO	0x0	reserved
6	RW	0x0	fout4phasepd Power down 4-phase clocks and 2X, 3X, 4X clocks 1'b0: no power down 1'b1: power down
5	RW	0x0	foutvcopd Power down buffered VCO clock 1'b0: no power down 1'b1: power down
4	RW	0x0	foutpostdivpd Power down all outputs except for buffered VCO clock 1'b0: no power down 1'b1: power down
3	RW	0x1	dsmpd PLL saturation behavior enable 1'b0: no power down 1'b1: power down DSMPD = 1'b1 (modulator is disabled, "integer mode")
2	RW	0x0	dacpd Power down quantization noise cancellation DAC 1'b0: no power down 1'b1: power down
1	RW	0x0	bypass PLL Bypass. FREF bypasses PLL to FOUTPOSTDIV 1'b0: no bypass 1'b1: bypass
0	RW	0x0	power_down Global power down 1'b0: no power down 1'b1: power down

CRU_CPLL_CON4
Address: Operational Base + offset (0x0070)
CPLL configuration register4

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:13	RO	0x0	reserved
12:8	RW	0x00	ssmod_spread spread amplitude % = 0.1 * SPREAD[4:0]

Bit	Attr	Reset Value	Description
7:4	RW	0x0	ssmod_divval Divider required to set the modulation frequency Divider required to set the modulation frequency
3	RW	0x0	ssmod_downspread Selects center spread or downs pread 1'b0: center spread 1'b1: down spread
2	RW	0x1	ssmod_reset Reset modulator state 1'b0: no reset 1'b1: reset
1	RW	0x1	ssmod_disable_sscg Bypass SSMOD by module 1'b0: no bypass 1'b1: bypass
0	RW	0x1	ssmod_bp Bypass SSMOD by integration 1'b0: no bypass 1'b1: bypass

CRU_CPLL_CON5
Address: Operational Base + offset (0x0074)
CPLL configuration register5

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:8	RW	0x7f	ssmod_ext_maxaddr External wave table data inputs (0-255)
7:1	RO	0x0	reserved
0	RW	0x0	ssmod_sel_ext_wave select external wave 1'b0: no select ext_wave 1'b1: select ext_wave

CRU_GPLL_CON0
Address: Operational Base + offset (0x0080)
GPLL configuration register0

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit

Bit	Attr	Reset Value	Description
15:12	RO	0x0	reserved
11:0	RW	0x0c6	fbdiv Feedback Divide Value Valid divider settings are: [16, 3200] in integer mode [20, 320] in fractional mode Tips: no plus one operation

CRU_GPLL_CON1
Address: Operational Base + offset (0x0084)
GPLL configuration register1

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RO	0x0	reserved
14:12	RW	0x2	postdiv2 Second Post Divide Value (1-7)
11	RO	0x0	reserved
10:8	RW	0x2	postdiv1 First Post Divide Value (1-7)
7:6	RO	0x0	reserved
5:0	RW	0x02	refdiv Reference Clock Divide Value (1-63)

CRU_GPLL_CON2
Address: Operational Base + offset (0x0088)
GPLL configuration register2

Bit	Attr	Reset Value	Description
31	RO	0x0	pll_lock PLL lock status 1'b0: unlock 1'b1: lock
30:24	RO	0x0	reserved
23:0	RW	0x00031f	fracdiv Fractional part of feedback divide (fraction = FRAC/2^24)

CRU_GPLL_CON3
Address: Operational Base + offset (0x008c)
GPLL configuration register3

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:10	RO	0x0	reserved
9:8	RW	0x0	pll_work_mode PLL work mode select 2'b00: Slow mode, clock from external 24MHz/26MHz OSC (default) 2'b01: Normal mode, clock from PLL output 2'b10: Deep slow mode, clock from external 32.768kHz
7	RO	0x0	reserved
6	RW	0x0	fout4phasepd Power down 4-phase clocks and 2X, 3X, 4X clocks 1'b0: no power down 1'b1: power down
5	RW	0x0	foutvcopd Power down buffered VCO clock 1'b0: no power down 1'b1: power down
4	RW	0x0	foutpostdivpd Power down all outputs except for buffered VCO clock 1'b0: no power down 1'b1: power down
3	RW	0x1	dsmpd PLL saturation behavior enable 1'b0: no power down 1'b1: power down DSMPD = 1'b1 (modulator is disabled, "integer mode")
2	RW	0x0	dacpd Power down quantization noise cancellation DAC 1'b0: no power down 1'b1: power down
1	RW	0x0	bypass PLL Bypass. FREF bypasses PLL to FOUTPOSTDIV 1'b0: no bypass 1'b1: bypass
0	RW	0x0	power_down Global power down 1'b0: no power down 1'b1: power down

CRU_GPLL_CON4
Address: Operational Base + offset (0x0090)
GPLL configuration register4

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:13	RO	0x0	reserved
12:8	RW	0x00	ssmod_spread spread amplitude % = 0.1 * SPREAD[4:0]
7:4	RW	0x0	ssmod_divval Divider required to set the modulation frequency Divider required to set the modulation frequency
3	RW	0x0	ssmod_downspread Selects center spread or downs pread 1'b0: center spread 1'b1: down spread
2	RW	0x1	ssmod_reset Reset modulator state 1'b0: no reset 1'b1: reset
1	RW	0x1	ssmod_disable_sscg Bypass SSMOD by module 1'b0: no bypass 1'b1: bypass
0	RW	0x1	ssmod_bp Bypass SSMOD by integration 1'b0: no bypass 1'b1: bypass

CRU_GPLL_CON5
Address: Operational Base + offset (0x0094)
GPLL configuration register5

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:8	RW	0x7f	ssmod_ext_maxaddr External wave table data inputs (0-255)
7:1	RO	0x0	reserved
0	RW	0x0	ssmod_sel_ext_wave select external wave 1'b0: no select ext_wave 1'b1: select ext_wave

CRU_NPLL_CON0
Address: Operational Base + offset (0x00a0)
NPLL configuration register0

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:12	RO	0x0	reserved
11:0	RW	0x0fa	fbdiv Feedback Divide Value Valid divider settings are: [16, 3200] in integer mode [20, 320] in fractional mode Tips: no plus one operation

CRU_NPLL_CON1
Address: Operational Base + offset (0x00a4)
NPLL configuration register1

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RO	0x0	reserved
14:12	RW	0x1	postdiv2 Second Post Divide Value (1-7)
11	RO	0x0	reserved
10:8	RW	0x2	postdiv1 First Post Divide Value (1-7)
7:6	RO	0x0	reserved
5:0	RW	0x03	refdiv Reference Clock Divide Value (1-63)

CRU_NPLL_CON2
Address: Operational Base + offset (0x00a8)
NPLL configuration register2

Bit	Attr	Reset Value	Description
31	RO	0x0	pll_lock PLL lock status 1'b0: unlock 1'b1: lock
30:24	RO	0x0	reserved

Bit	Attr	Reset Value	Description
23:0	RW	0x00031f	fracdiv Fractional part of feedback divide (fraction = FRAC/2^24)

CRU_NPLL_CON3
Address: Operational Base + offset (0x00ac)
NPLL configuration register3

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:10	RO	0x0	reserved
9:8	RW	0x0	pll_work_mode PLL work mode select 2'b00: Slow mode, clock from external 24MHz/26MHz OSC (default) 2'b01: Normal mode, clock from PLL output 2'b10: Deep slow mode, clock from external 32.768kHz
7	RO	0x0	reserved
6	RW	0x0	fout4phasepd Power down 4-phase clocks and 2X, 3X, 4X clocks 1'b0: no power down 1'b1: power down
5	RW	0x0	foutvcopd Power down buffered VCO clock 1'b0: no power down 1'b1: power down
4	RW	0x0	foutpostdivpd Power down all outputs except for buffered VCO clock 1'b0: no power down 1'b1: power down
3	RW	0x1	dsmpd PLL saturation behavior enable 1'b0: no power down 1'b1: power down DSMPD = 1'b1 (modulator is disabled, "integer mode")
2	RW	0x0	dacpd Power down quantization noise cancellation DAC 1'b0: no power down 1'b1: power down
1	RW	0x0	bypass PLL Bypass. FREF bypasses PLL to FOUTPOSTDIV 1'b0: no bypass 1'b1: bypass

Bit	Attr	Reset Value	Description
0	RW	0x0	power_down Global power down 1'b0: no power down 1'b1: power down

CRU_NPLL_CON4
Address: Operational Base + offset (0x00b0)
NPLL configuration register4

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:13	RO	0x0	reserved
12:8	RW	0x00	ssmod_spread spread amplitude % = 0.1 * SPREAD[4:0]
7:4	RW	0x0	ssmod_divval Divider required to set the modulation frequency Divider required to set the modulation frequency
3	RW	0x0	ssmod_downspread Selects center spread or downs pread 1'b0: center spread 1'b1: down spread
2	RW	0x1	ssmod_reset Reset modulator state 1'b0: no reset 1'b1: reset
1	RW	0x1	ssmod_disable_sscg Bypass SSMOD by module 1'b0: no bypass 1'b1: bypass
0	RW	0x1	ssmod_bp Bypass SSMOD by integration 1'b0: no bypass 1'b1: bypass

CRU_NPLL_CON5
Address: Operational Base + offset (0x00b4)
NPLL configuration register5

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit

Bit	Attr	Reset Value	Description
15:8	RW	0x7f	ssmod_ext_maxaddr External wave table data inputs (0-255)
7:1	RO	0x0	reserved
0	RW	0x0	ssmod_sel_ext_wave select external wave 1'b0: no select ext_wave 1'b1: select ext_wave

CRU_VPLL_CON0
Address: Operational Base + offset (0x00c0)
VPLL configuration register0

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:12	RO	0x0	reserved
11:0	RW	0x0c6	fbdiv Feedback Divide Value Valid divider settings are: [16, 3200] in integer mode [20, 320] in fractional mode Tips: no plus one operation

CRU_VPLL_CON1
Address: Operational Base + offset (0x00c4)
VPLL configuration register1

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RO	0x0	reserved
14:12	RW	0x1	postdiv2 Second Post Divide Value (1-7)
11	RO	0x0	reserved
10:8	RW	0x2	postdiv1 First Post Divide Value (1-7)
7:6	RO	0x0	reserved
5:0	RW	0x02	refdiv Reference Clock Divide Value (1-63)

CRU_VPLL_CON2
Address: Operational Base + offset (0x00c8)
VPLL configuration register2

Bit	Attr	Reset Value	Description
31	RO	0x0	pll_lock PLL lock status 1'b0: unlock 1'b1: lock
30:24	RO	0x0	reserved
23:0	RW	0x00031f	fracdiv Fractional part of feedback divide (fraction = FRAC/2^24)

CRU_VPLL_CON3
Address: Operational Base + offset (0x00cc)
VPLL configuration register3

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:10	RO	0x0	reserved
9:8	RW	0x0	pll_work_mode PLL work mode select 2'b00: Slow mode, clock from external 24MHz/26MHz OSC (default) 2'b01: Normal mode, clock from PLL output 2'b10: Deep slow mode, clock from external 32.768kHz
7	RO	0x0	reserved
6	RW	0x0	fout4phasepd Power down 4-phase clocks and 2X, 3X, 4X clocks 1'b0: no power down 1'b1: power down
5	RW	0x0	foutvcopd Power down buffered VCO clock 1'b0: no power down 1'b1: power down
4	RW	0x0	foutpostdivpd Power down all outputs except for buffered VCO clock 1'b0: no power down 1'b1: power down

Bit	Attr	Reset Value	Description
3	RW	0x1	dsmpd PLL saturation behavior enable 1'b0: no power down 1'b1: power down DSMPD = 1'b1 (modulator is disabled, "integer mode")
2	RW	0x0	dacpd Power down quantization noise cancellation DAC 1'b0: no power down 1'b1: power down
1	RW	0x0	bypass PLL Bypass. FREF bypasses PLL to FOUTPOSTDIV 1'b0: no bypass 1'b1: bypass
0	RW	0x0	power_down Global power down 1'b0: no power down 1'b1: power down

CRU_VPLL_CON4
Address: Operational Base + offset (0x00d0)
VPLL configuration register4

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:13	RO	0x0	reserved
12:8	RW	0x00	ssmod_spread spread amplitude % = 0.1 * SPREAD[4:0]
7:4	RW	0x0	ssmod_divval Divider required to set the modulation frequency Divider required to set the modulation frequency
3	RW	0x0	ssmod_downspread Selects center spread or downs pread 1'b0: center spread 1'b1: down spread
2	RW	0x1	ssmod_reset Reset modulator state 1'b0: no reset 1'b1: reset
1	RW	0x1	ssmod_disable_sscg Bypass SSMOD by module 1'b0: no bypass 1'b1: bypass

Bit	Attr	Reset Value	Description
0	RW	0x1	ssmod_bp Bypass SSMOD by integration 1'b0: no bypass 1'b1: bypass

CRU_VPLL_CON5
Address: Operational Base + offset (0x00d4)
VPLL configuration register5

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:8	RW	0x7f	ssmod_ext_maxaddr External wave table data inputs (0-255)
7:1	RO	0x0	reserved
0	RW	0x0	ssmod_sel_ext_wave select external wave 1'b0: no select ext_wave 1'b1: select ext_wave

CRU_CLKSEL_CON0
Address: Operational Base + offset (0x0100)
Internal clock select and divide register0

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:13	RO	0x0	reserved
12:8	RW	0x01	aclkm_core_l_div_con aclkm_core_l divider control register clk=clk_src/(div_con+1)
7:6	RW	0x0	clk_core_l_pll_sel clk_core_l clock source select control register 2'b00:LPLL 2'b01:BPLL 2'b10:DPLL 2'b11:GPLL
5	RO	0x0	reserved
4:0	RW	0x01	clk_core_l_div_con clk_core_l divider control register clk=clk_src/(div_con+1)

CRU_CLKSEL_CON1
Address: Operational Base + offset (0x0104)
Internal clock select and divide register1

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:13	RO	0x0	reserved
12:8	RW	0x03	pclk_dbg_l_div_con pclk_dbg_l divider control register clk=clk_src/(div_con+1)
7:5	RO	0x0	reserved
4:0	RW	0x03	atclk_core_l_div_con atclk_core_l divider control register clk=clk_src/(div_con+1)

CRU_CLKSEL_CON2
Address: Operational Base + offset (0x0108)
Internal clock select and divide register2

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:13	RO	0x0	reserved
12:8	RW	0x01	aclkm_core_b_div_con aclkm_core_b divider control register clk=clk_src/(div_con+1)
7:6	RW	0x1	clk_core_b_pll_sel clk_core_b clock source select control register 2'b00:LPLL 2'b01:BPLL 2'b10:DPLL 2'b11:GPLL
5	RO	0x0	reserved
4:0	RW	0x01	clk_core_b_div_con clk_core_b divider control register clk=clk_src/(div_con+1)

CRU_CLKSEL_CON3
Address: Operational Base + offset (0x010c)
Internal clock select and divide register3

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RO	0x0	reserved
14:13	RW	0x3	pclken_dbg_b_div_con pclken_dbg_b divider control register clk=clk_src/(div_con+1)
12:8	RW	0x03	pclk_dbg_b_div_con pclk_dbg_b divider control register clk=clk_src/(div_con+1)
7:5	RO	0x0	reserved
4:0	RW	0x03	atclk_core_b_div_con atclk_core_b divider control register clk=clk_src/(div_con+1)

CRU_CLKSEL_CON4

Address: Operational Base + offset (0x0110)

Internal clock select and divide register4

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:8	RO	0x0	reserved
7:6	RW	0x1	clk_cs_pll_sel clk_cs clock source select control register 2'b00:CPLL 2'b01:GPLL 2'b1x:NPLL
5	RO	0x0	reserved
4:0	RW	0x01	clk_cs_div_con clk_cs divider control register clk=clk_src/(div_con+1)

CRU_CLKSEL_CON5

Address: Operational Base + offset (0x0114)

Internal clock select and divide register5

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit

Bit	Attr	Reset Value	Description
15	RW	0x1	clk_cci_trace_pll_sel clk_cci_trace clock source select control register 1'b0:CPLL 1'b1:GPLL
14:13	RO	0x0	reserved
12:8	RW	0x03	clk_cci_trace_div_con clk_cci_trace divider control register clk=clk_src/(div_con+1)
7:6	RW	0x1	aclk_cci_pll_sel aclk_cci clock source select control register 2'b00:CPLL 2'b01:GPLL 2'b10:NPLL 2'b11:VPLL
5	RO	0x0	reserved
4:0	RW	0x01	aclk_cci_div_con aclk_cci divider control register clk=clk_src/(div_con+1)

CRU_CLKSEL_CON6
Address: Operational Base + offset (0x0118)
Internal clock select and divide register6

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	pclk_ddr_pll_sel pclk_ddr clock source select control register 1'b0:CPLL 1'b1:GPLL
14:13	RO	0x0	reserved
12:8	RW	0x03	pclk_ddr_div_con pclk_ddr divider control register clk=clk_src/(div_con+1)
7:6	RO	0x0	reserved
5:4	RW	0x2	clk_ddrc_pll_sel clk_ddrc clock source select control register 2'b00:LPLL 2'b01:BPLL 2'b10:DPLL 2'b11:GPLL
3	RO	0x0	reserved

Bit	Attr	Reset Value	Description
2:0	RW	0x0	clk_ddrc_div_con clk_ddrc divider control register clk=clk_src/(div_con+1)

CRU_CLKSEL_CON7
Address: Operational Base + offset (0x011c)
Internal clock select and divide register7

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:13	RO	0x0	reserved
12:8	RW	0x01	hclk_vcodec_div_con hclk_vcodec divider control register clk=clk_src/(div_con+1)
7:6	RW	0x0	aclk_vcodec_pll_sel aclk_vcodec clock source select control register 2'b00:CPLL 2'b01:GPLL 2'b10:NPLL 2'b11:PPLL
5	RO	0x0	reserved
4:0	RW	0x01	aclk_vcodec_div_con aclk_vcodec divider control register clk=clk_src/(div_con+1)

CRU_CLKSEL_CON8
Address: Operational Base + offset (0x0120)
Internal clock select and divide register8

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:13	RO	0x0	reserved
12:8	RW	0x01	hclk_vdu_div_con hclk_vdu divider control register clk=clk_src/(div_con+1)
7:6	RW	0x0	aclk_vdu_pll_sel aclk_vdu clock source select control register 2'b00:CPLL 2'b01:GPLL 2'b10:NPLL 2'b11:PPLL

Bit	Attr	Reset Value	Description
5	RO	0x0	reserved
4:0	RW	0x01	aclk_vdu_div_con aclk_vdu divider control register clk=clk_src/(div_con+1)

CRU_CLKSEL_CON9
Address: Operational Base + offset (0x0124)
Internal clock select and divide register9

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:14	RW	0x1	clk_vdu_ca_pll_sel clk_vdu_ca clock source select control register 2'b00:CPLL 2'b01:GPLL 2'b1x:NPLL
13	RO	0x0	reserved
12:8	RW	0x01	clk_vdu_ca_div_con clk_vdu_ca divider control register clk=clk_src/(div_con+1)
7:6	RW	0x1	clk_vdu_core_pll_sel clk_vdu_core clock source select control register 2'b00:CPLL 2'b01:GPLL 2'b1x:NPLL
5	RO	0x0	reserved
4:0	RW	0x01	clk_vdu_core_div_con clk_vdu_core divider control register clk=clk_src/(div_con+1)

CRU_CLKSEL_CON10
Address: Operational Base + offset (0x0128)
Internal clock select and divide register10

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:13	RO	0x0	reserved
12:8	RW	0x01	hclk_iep_div_con hclk_iep divider control register clk=clk_src/(div_con+1)

Bit	Attr	Reset Value	Description
7:6	RW	0x0	aclk_iep_pll_sel aclk_iep clock source select control register 2'b00:CPLL 2'b01:GPLL 2'b10:NPLL 2'b11:PPLL
5	RO	0x0	reserved
4:0	RW	0x01	aclk_iep_div_con aclk_iep divider control register clk=clk_src/(div_con+1)

CRU_CLKSEL_CON11
Address: Operational Base + offset (0x012c)
Internal clock select and divide register11

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:13	RO	0x0	reserved
12:8	RW	0x01	hclk_rga_div_con hclk_rga divider control register clk=clk_src/(div_con+1)
7:6	RW	0x0	aclk_rga_pll_sel aclk_rga clock source select control register 2'b00:CPLL 2'b01:GPLL 2'b10:NPLL 2'b11:PPLL
5	RO	0x0	reserved
4:0	RW	0x01	aclk_rga_div_con aclk_rga divider control register clk=clk_src/(div_con+1)

CRU_CLKSEL_CON12
Address: Operational Base + offset (0x0130)
Internal clock select and divide register12

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit

Bit	Attr	Reset Value	Description
15:14	RW	0x0	aclk_center_pll_sel aclk_center clock source select control register 2'b00:CPLL 2'b01:GPLL 2'b1x:NPLL
13	RO	0x0	reserved
12:8	RW	0x01	aclk_center_div_con aclk_center divider control register clk=clk_src/(div_con+1)
7:6	RW	0x0	clk_rga_core_pll_sel clk_rga_core clock source select control register 2'b00:CPLL 2'b01:GPLL 2'b10:NPLL 2'b11:PPLL
5	RO	0x0	reserved
4:0	RW	0x00	clk_rga_core_div_con clk_rga_core divider control register clk=clk_src/(div_con+1)

CRU_CLKSEL_CON13

Address: Operational Base + offset (0x0134)

Internal clock select and divide register13

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	hclk_sd_src_sel hclk_sd clock source select control register 1'b0:CPLL 1'b1:GPLL
14:13	RO	0x0	reserved
12:8	RW	0x03	hclk_sd_div_con hclk_sd divider control register clk=clk_src/(div_con+1)
7:5	RW	0x3	aclk_gpu_pll_sel aclk_gpu clock source select control register 3'b000:PPLL 3'b001:CPLL 3'b010:GPLL 3'b011:NPLL 3'b100:USB_480M

Bit	Attr	Reset Value	Description
4:0	RW	0x01	aclk_gpu_div_con aclk_gpu divider control register clk=clk_src/(div_con+1)

CRU_CLKSEL_CON14
Address: Operational Base + offset (0x0138)
Internal clock select and divide register14

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	clk_usbpll_480m_sel clk_usbpll_480m_sel clock select control register 1'b0:xin_24m 1'b1:clk_usbphy_480m
14:12	RW	0x3	pclk_perihp_div_con pclk_perihp divider control register clk=aclk_perihp/(div_con+1)
11:10	RO	0x0	reserved
9:8	RW	0x1	hclk_perihp_div_con hclk_perihp divider control register clk=aclk_perihp/(div_con+1)
7	RW	0x1	aclk_perihp_pll_sel aclk_perihp clock source select control register 1'b0:CPLL 1'b1:GPLL
6	RW	0x0	clk_usbphy_480m_ch_sel clk_usbphy_480m clock channel select control register 1'b0:usb_phy0_480m 1'b1:usb_phy1_480m
5	RO	0x0	reserved
4:0	RW	0x01	aclk_perihp_div_con aclk_perihp divider control register clk=clk_src/(div_con+1)

CRU_CLKSEL_CON15
Address: Operational Base + offset (0x013c)
Internal clock select and divide register15

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:11	RO	0x0	reserved

Bit	Attr	Reset Value	Description
10:8	RW	0x5	clk_sdio_pll_sel clk_sdio clock source select control register 3'b000:CPLL 3'b001:GPLL 3'b010:NPLL 3'b011:PPLL 3'b100:USB_480M 3'b101:xin_24m
7	RO	0x0	reserved
6:0	RW	0x00	clk_sdio_div_con clk_sdio divider control register clk=clk_src/(div_con+1)

CRU_CLKSEL_CON16
Address: Operational Base + offset (0x0140)
Internal clock select and divide register16

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:11	RO	0x0	reserved
10:8	RW	0x5	clk_sdmmc_pll_sel clk_sdmmc clock source select control register 3'b000:CPLL 3'b001:GPLL 3'b010:NPLL 3'b011:PPLL 3'b100:USB_480M 3'b101:xin_24m
7	RO	0x0	reserved
6:0	RW	0x00	clk_sdmmc_div_con clk_sdmmc divider control register clk=clk_src/(div_con+1)

CRU_CLKSEL_CON17
Address: Operational Base + offset (0x0144)
Internal clock select and divide register17

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:11	RO	0x0	reserved

Bit	Attr	Reset Value	Description
10:8	RW	0x3	clk_pcie_pm_pll_sel clk_pcie_pm clock source select control register 3'b000:CPLL 3'b001:GPLL 3'b010:NPLL 3'b011:xin_24m 3'b1xx:reserved
7	RO	0x0	reserved
6:0	RW	0x00	clk_pcie_pm_div_con clk_pcie_pm divider control register clk=clk_src/(div_con+1)

CRU_CLKSEL_CON18
Address: Operational Base + offset (0x0148)
Internal clock select and divide register18

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:11	RW	0x09	clk_pciephy_ref100m_div_con clk_pciephy_ref100m divider control register clk=clk_src/(div_con+1)
10	RW	0x0	clk_pciephy_ref_sel clk_pciephy_ref clock select control register 1'b0:clk_pcie_ref24m 1'b1:clk_pcie_ref100m
9:8	RW	0x2	clk_pcie_core_pll_sel clk_pcie_core clock source select control register 2'b00:CPLL 2'b01:GPLL 2'b1x:NPLL
7	RW	0x1	clk_pcie_core_clk_sel clk_pcie_core clock select control register 1'b0:clk_pcie_core 1'b1:pipe_clk_pcie from PCIE PHY
6:0	RW	0x07	clk_pcie_core_div_con clk_pcie_core divider control register clk=clk_src/(div_con+1)

CRU_CLKSEL_CON19
Address: Operational Base + offset (0x014c)
Internal clock select and divide register19

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:11	RO	0x0	reserved
10:8	RW	0x1	pclk_gmac_div_con pclk_gmac divider control register clk=clk_src/(div_con+1)
7:5	RO	0x0	reserved
4	RW	0x0	clk_rmii_src_sel clk_rmii_src clock select control register 1'b0:clk_mac_divout 1'b1:rmii_clkin from IO
3:2	RO	0x0	reserved
1:0	RW	0x0	clk_hsicphy_pll_sel clk_hsicphy clock source select control register 2'b00:CPLL 2'b01:GPLL 2'b10:NPLL 2'b11:USB_480M

CRU_CLKSEL_CON20
Address: Operational Base + offset (0x0150)
Internal clock select and divide register20

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:14	RW	0x2	clk_gmac_pll_sel clk_gmac clock source select control register 2'b00:CPLL 2'b01:GPLL 2'b1x:NPLL
13	RO	0x0	reserved
12:8	RW	0x13	clk_gmac_div_con clk_gmac divider control register clk=clk_src/(div_con+1)
7	RW	0x0	aclk_gmac_pll_sel aclk_gmac clock source select control register 1'b0:CPLL 1'b1:GPLL
6:5	RO	0x0	reserved

Bit	Attr	Reset Value	Description
4:0	RW	0x03	aclk_gmac_div_con aclk_gmac divider control register clk=clk_src/(div_con+1)

CRU_CLKSEL_CON21
Address: Operational Base + offset (0x0154)
Internal clock select and divide register21

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:8	RO	0x0	reserved
7	RW	0x0	aclk_emmc_pll_sel aclk_emmc clock source select control register 1'b0:CPLL 1'b1:GPLL
6:5	RO	0x0	reserved
4:0	RW	0x03	aclk_emmc_div_con aclk_emmc divider control register clk=clk_src/(div_con+1)

CRU_CLKSEL_CON22
Address: Operational Base + offset (0x0158)
Internal clock select and divide register22

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:11	RO	0x0	reserved
10:8	RW	0x4	clk_emmc_pll_sel clk_emmc clock source select control register 3'b000:CPLL 3'b001:GPLL 3'b010:NPLL 3'b011:USB_480M 3'b1xx:xin_24m
7	RO	0x0	reserved
6:0	RW	0x00	clk_emmc_div_con clk_emmc divider control register clk=clk_src/(div_con+1)

CRU_CLKSEL_CON23
Address: Operational Base + offset (0x015c)

Internal clock select and divide register23

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RO	0x0	reserved
14:12	RW	0x3	pclk_perilp0_div_con perilp0_pclk divider control register clk=aclk_perilp0/(div_con+1)
11:10	RO	0x0	reserved
9:8	RW	0x1	hclk_perilp0_div_con perilp0_hclk divider control register clk=aclk_perilp0/(div_con+1)
7	RW	0x1	aclk_perilp0_pll_sel aclk_perilp0 clock source select control register 1'b0:CPLL 1'b1:GPLL
6:5	RO	0x0	reserved
4:0	RW	0x01	aclk_perilp0_div_con aclk_perilp0 divider control register clk=clk_src/(div_con+1)

CRU_CLKSEL_CON24

Address: Operational Base + offset (0x0160)

Internal clock select and divide register24

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x1	fclk_cm0s_pll_sel fclk_cm0s clock source select control register 1'b0:CPLL 1'b1:GPLL
14:13	RO	0x0	reserved
12:8	RW	0x01	fclk_cm0s_div_con fclk_cm0s divider control register clk=clk_src/(div_con+1)
7:6	RW	0x0	clk_crypto0_pll_sel clk_crypto0 clock source select control register 2'b00:CPLL 2'b01:GPLL 2'b10:PPLL
5	RO	0x0	reserved

Bit	Attr	Reset Value	Description
4:0	RW	0x03	clk_crypto0_div_con clk_crypto0 divider control register clk=clk_src/(div_con+1)

CRU_CLKSEL_CON25
Address: Operational Base + offset (0x0164)
Internal clock select and divide register25

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:11	RO	0x0	reserved
10:8	RW	0x1	pclk_perilp1_div_con pclk_perilp1 divider control register clk=hclk_perilp1/(div_con+1)
7	RW	0x1	hclk_perilp1_pll_sel hclk_perilp1 clock source select control register 1'b0:CPLL 1'b1:GPLL
6:5	RO	0x0	reserved
4:0	RW	0x03	hclk_perilp1_div_con hclk_perilp1 divider control register clk=clk_src/(div_con+1)

CRU_CLKSEL_CON26
Address: Operational Base + offset (0x0168)
Internal clock select and divide register26

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:8	RW	0x01	clk_saradc_div_con clk_saradc divider control register clk=clk_src/(div_con+1)
7:6	RW	0x0	clk_crypto1_pll_sel clk_crypto1 clock source select control register 2'b00:CPLL 2'b01:GPLL 2'b10:PPLL
5	RO	0x0	reserved
4:0	RW	0x03	clk_crypto1_div_con clk_crypto1 divider control register clk=clk_src/(div_con+1)

CRU_CLKSEL_CON27
Address: Operational Base + offset (0x016c)
Internal clock select and divide register27

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	clk_tsadc_sel clk tsadc clock select control register 1'b0:xin_24m 1'b1:clk_32k
14:10	RO	0x0	reserved
9:0	RW	0x2dc	clk_tsadc_div_con clk tsadc divider control register clk=clk_src/(div_con+1)

CRU_CLKSEL_CON28
Address: Operational Base + offset (0x0170)
Internal clock select and divide register28

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:10	RO	0x0	reserved
9:8	RW	0x3	clk_i2s0_sel clk_i2s0 clock select control register 2'b00:clk_i2s0_divout 2'b01:clk_i2s0_frac 2'b10:clkin_i2s from IO 2'b11:clk_12m
7	RW	0x0	clk_i2s0_pll_sel clk_i2s0 clock source select control register 1'b0:CPLL 1'b1:GPLL
6:0	RW	0x00	clk_i2s0_div_con clk_i2s0 divider control register clk=clk_src/(div_con+1)

CRU_CLKSEL_CON29
Address: Operational Base + offset (0x0174)
Internal clock select and divide register29

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:10	RO	0x0	reserved
9:8	RW	0x3	clk_i2s1_sel clk_i2s1 clock select control register 2'b00:clk_i2s1_divout 2'b01:clk_i2s1_frac 2'b10:clkin_i2s from IO 2'b11:clk_12m
7	RW	0x0	clk_i2s1_pll_sel clk_i2s1 clock source select control register 1'b0:CPLL 1'b1:GPLL
6:0	RW	0x00	clk_i2s1_div_con clk_i2s1 divider control register clk=clk_src/(div_con+1)

CRU_CLKSEL_CON30
Address: Operational Base + offset (0x0178)
Internal clock select and divide register30

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:10	RO	0x0	reserved
9:8	RW	0x3	clk_i2s2_sel clk_i2s2 clock select control register 2'b00:clk_i2s2_divout 2'b01:clk_i2s2_frac 2'b10:clkin_i2s2 from IO 2'b11:clk_12m
7	RW	0x0	clk_i2s2_pll_sel clk_i2s2 clock source select control register 1'b0:CPLL 1'b1:GPLL
6:0	RW	0x00	clk_i2s2_div_con clk_i2s2 divider control register clk=clk_src/(div_con+1)

CRU_CLKSEL_CON31
Address: Operational Base + offset (0x017c)
Internal clock select and divide register31

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:3	RO	0x0	reserved
2	RW	0x0	clk_i2sout_sel clk_i2sout clock select control register 1'b0:clk_i2s 1'b1:clk_12m
1:0	RW	0x0	clk_i2s_ch_sel clk_i2s_ch clock select control register 2'b00:clk_i2s0 2'b01:clk_i2s1 2'b10:clk_i2s2

CRU_CLKSEL_CON32
Address: Operational Base + offset (0x0180)
Internal clock select and divide register32

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	clk_dptx_spdif_rec_pll_sel clk_dptx_spdif_rec clock source select control register 1'b0:CPLL 1'b1:GPLL
14:13	RW	0x3	clk_spdif_8ch_clk_sel clk_spdif_8ch clock select control register 2'b00:clk_spdif_divout 2'b01:clk_spdif_frac 2'b10:clkin_spdif from IO SAME AS clkin_i2s 2'b11:clk_12m
12:8	RW	0x03	clk_dptx_spdif_rec_div_con clk_dptx_spdif_rec divider control register clk=clk_src/(div_con+1)
7	RW	0x0	clk_spdif_8ch_pll_sel clk_spdif_8ch clock source select control register 1'b0:CPLL 1'b1:GPLL
6:0	RW	0x00	clk_spdif_8ch_pll_div_con clk_spdif_8ch_pll divider control register clk=clk_src/(div_con+1)

CRU_CLKSEL_CON33

Address: Operational Base + offset (0x0184)
Internal clock select and divide register33

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	clk_uart_pll_sel clk_uart clock source select control register 1'b0:CPLL 1'b1:GPLL
14	RO	0x0	reserved
13:12	RW	0x0	clk_uart0_src_sel clk_uart0_src clock select control register 2'b00:CPLL 2'b01:GPLL 2'b1x:USB_480M
11:10	RO	0x0	reserved
9:8	RW	0x2	clk_uart0_sel clk_uart0 clock select control register 2'b00:clk_uart0_divout 2'b01:clk_uart0_frac 2'b10:xin_24m
7	RO	0x0	reserved
6:0	RW	0x00	clk_uart0_div_con clk_uart0 divider control register clk=clk_src/(div_con+1)

CRU_CLKSEL_CON34
Address: Operational Base + offset (0x0188)
Internal clock select and divide register34

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	clk_write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:10	RO	0x0	reserved
9:8	RW	0x2	clk_uart1_sel clk_uart1 clock select control register 2'b00:clk_uart1_divout 2'b01:clk_uart1_frac 2'b10:xin_24m
7	RO	0x0	reserved
6:0	RW	0x00	clk_uart1_div_con clk_uart1 divider control register clk=clk_src/(div_con+1)

CRU_CLKSEL_CON35
Address: Operational Base + offset (0x018c)
Internal clock select and divide register35

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:10	RO	0x0	reserved
9:8	RW	0x2	clk_uart2_sel clk_uart2 clock select control register 2'b00:clk_uart2_divout 2'b01:clk_uart2_frac 2'b10:xin_24m
7	RO	0x0	reserved
6:0	RW	0x00	clk_uart2_div_con clk_uart2 divider control register clk=clk_src/(div_con+1)

CRU_CLKSEL_CON36
Address: Operational Base + offset (0x0190)
Internal clock select and divide register36

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:10	RO	0x0	reserved
9:8	RW	0x2	clk_uart3_sel clk_uart3 clock select control register 2'b00:clk_uart3_divout 2'b01:clk_uart3_frac 2'b10:xin_24m
7	RO	0x0	reserved
6:0	RW	0x00	clk_uart3_div_con clk_uart3 divider control register clk=clk_src/(div_con+1)

CRU_CLKSEL_CON38
Address: Operational Base + offset (0x0198)
Internal clock select and divide register38

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit

Bit	Attr	Reset Value	Description
15:14	RW	0x0	clk_testout2_pll_sel clk_testout2 clock source select control register 2'b00:CPLL 2'b01:GPLL 2'b1x:NPLL
13	RW	0x1	clk_testout2_clk_sel clk_testout2 clock select control register 1'b0:clk_testout_src 1'b1:xin_24m
12:8	RW	0x1f	clk_testout2_div_con clk_testout2 divider control register clk=clk_src/(div_con+1)
7:6	RW	0x0	clk_testout1_pll_sel clk_testout1 clock source select control register 2'b00:CPLL 2'b01:GPLL 2'b1x:NPLL
5	RW	0x1	clk_testout1_clk_sel clk_testout1 clock select control register 1'b0:clk_testout_src 1'b1:xin_24m
4:0	RW	0x1f	clk_testout1_div_con clk_testout1 divider control register clk=clk_src/(div_con+1)

CRU_CLKSEL_CON39

Address: Operational Base + offset (0x019c)

Internal clock select and divide register39

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:8	RO	0x0	reserved
7:6	RW	0x1	aclk_usb3_pll_sel aclk_usb3 clock source select control register 2'b00:CPLL 2'b01:GPLL 2'b1x:NPLL
5	RO	0x0	reserved
4:0	RW	0x01	aclk_usb3_div_con aclk_usb3 divider control register clk=clk_src/(div_con+1)

CRU_CLKSEL_CON40

Address: Operational Base + offset (0x01a0)
Internal clock select and divide register40

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	clk_usb3_otg0_suspend_src_sel clk_usb3_otg0_suspend_src clock select control register 1'b0:xin_24m 1'b1:clk_32k
14:10	RO	0x0	reserved
9:0	RW	0x000	clk_usb3_otg0_suspend_div_con clk_usb3_otg0_suspend divider control register clk=clk_src/(div_con+1)

CRU_CLKSEL_CON41
Address: Operational Base + offset (0x01a4)
Internal clock select and divide register41

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	clk_usb3_otg1_suspend_src_sel clk_usb3_otg1_suspend_src clock select control register 1'b0:xin_24m 1'b1:clk_32k
14:10	RO	0x0	reserved
9:0	RW	0x000	clk_usb3_otg1_suspend_div_con clk_usb3_otg1_suspend divider control register clk=clk_src/(div_con+1)

CRU_CLKSEL_CON42
Address: Operational Base + offset (0x01a8)
Internal clock select and divide register42

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:14	RW	0x0	aclk_hdcp_pll_sel aclk_hdcp clock source select control register 2'b00:CPLL 2'b01:GPLL 2'b10:PPLL

Bit	Attr	Reset Value	Description
13	RO	0x0	reserved
12:8	RW	0x01	aclk_hdcp_div_con aclk_hdcp divider control register clk=clk_src/(div_con+1)
7:6	RW	0x0	aclk_vio_pll_sel aclk_vio clock source select control register 2'b00:CPLL 2'b01:GPLL 2'b10:PPLL
5	RO	0x0	reserved
4:0	RW	0x01	aclk_vio_div_con aclk_vio divider control register clk=clk_src/(div_con+1)

CRU_CLKSEL_CON43
Address: Operational Base + offset (0x01ac)
Internal clock select and divide register43

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RO	0x0	reserved
14:10	RW	0x01	pclk_hdcp_div_con pclk_hdcp divider control register clk=clk_src/(div_con+1)
9:5	RW	0x01	hclk_hdcp_div_con hclk_hdcp divider control register clk=clk_src/(div_con+1)
4:0	RW	0x01	pclk_vio_div_con pclk_vio divider control register clk=clk_src/(div_con+1)

CRU_CLKSEL_CON44
Address: Operational Base + offset (0x01b0)
Internal clock select and divide register44

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	pclk_edp_pll_sel pclk_edp clock source select control register 1'b0:CPLL 1'b1:GPLL

Bit	Attr	Reset Value	Description
14	RO	0x0	reserved
13:8	RW	0x07	pclk_edp_div_con pclk_edp divider control register clk=clk_src/(div_con+1)
7:0	RO	0x0	reserved

CRU_CLKSEL_CON45
Address: Operational Base + offset (0x01b4)
Internal clock select and divide register45

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	clk_hdmi_cec_src_sel clk_hdmi_cec_src clock select control register 1'b0:clk_32k 1'b1:xin_24m
14:10	RO	0x0	reserved
9:0	RW	0x2dc	clk_hdmi_cec_div_con clk_hdmi_cec divider control register clk=clk_src/(div_con+1)

CRU_CLKSEL_CON46
Address: Operational Base + offset (0x01b8)
Internal clock select and divide register46

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:8	RO	0x0	reserved
7:6	RW	0x0	clk_dp_core_pll_sel clk_dp_core clock source select control register 2'b00:NPLL 2'b01:CPLL 2'b10:GPLL
5	RO	0x0	reserved
4:0	RW	0x04	clk_dp_core_div_con clk_dp_core divider control register clk=clk_src/(div_con+1)

CRU_CLKSEL_CON47
Address: Operational Base + offset (0x01bc)
Internal clock select and divide register47

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:13	RO	0x0	reserved
12:8	RW	0x01	hclk_vop0_div_con hclk_vop0 divider control register clk=clk_src/(div_con+1)
7:6	RW	0x0	aclk_vop0_pll_sel aclk_vop0 clock source select control register 2'b00:VPLL 2'b01:CPLL 2'b10:GPLL 2'b11:NPLL
5	RO	0x0	reserved
4:0	RW	0x02	aclk_vop0_div_con aclk_vop0 divider control register clk=clk_src/(div_con+1)

CRU_CLKSEL_CON48
Address: Operational Base + offset (0x01c0)
Internal clock select and divide register48

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:13	RO	0x0	reserved
12:8	RW	0x01	hclk_vop1_div_con hclk_vop1 divider control register clk=clk_src/(div_con+1)
7:6	RW	0x0	aclk_vop1_pll_sel aclk_vop1 clock source select control register 2'b00:VPLL 2'b01:CPLL 2'b10:GPLL 2'b11:NPLL
5	RO	0x0	reserved
4:0	RW	0x02	aclk_vop1_div_con aclk_vop1 divider control register clk=clk_src/(div_con+1)

CRU_CLKSEL_CON49
Address: Operational Base + offset (0x01c4)
Internal clock select and divide register49

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:12	RO	0x0	reserved
11	RW	0x0	dclk_vop0_dclk_sel dclk_vop0 clock select control register 1'b0:dclk_vop_divout 1'b1:dclk_vop_frac
10	RO	0x0	reserved
9:8	RW	0x0	dclk_vop0_pll_sel dclk_vop0 clock source select control register 2'b00:VPLL 2'b01:CPLL 2'b1x:GPLL
7:0	RW	0x01	dclk_vop0_div_con dclk_vop0 divider control register clk=clk_src/(div_con+1)

CRU_CLKSEL_CON50
Address: Operational Base + offset (0x01c8)
Internal clock select and divide register50

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:12	RO	0x0	reserved
11	RW	0x0	dclk_vop1_dclk_sel dclk_vop1 clock select control register 1'b0:dclk_vop_divout 1'b1:dclk_vop_frac
10	RO	0x0	reserved
9:8	RW	0x0	dclk_vop1_pll_sel dclk_vop1 clock source select control register 2'b00:VPLL 2'b01:CPLL 2'b1x:GPLL
7:0	RW	0x03	dclk_vop1_div_con vop1 divider control register clk=clk_src/(div_con+1)

CRU_CLKSEL_CON51
Address: Operational Base + offset (0x01cc)
Internal clock select and divide register51

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:8	RO	0x0	reserved
7:6	RW	0x0	clk_vop0_pwm_src_sel vop0_pwm_src clock select control register 2'b00:VPLL 2'b01:CPLL 2'b1x:GPLL
5	RO	0x0	reserved
4:0	RW	0x05	clk_vop0_pwm_div_con vop0_pwm divider control register clk=clk_src/(div_con+1)

CRU_CLKSEL_CON52
Address: Operational Base + offset (0x01d0)
Internal clock select and divide register52

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:8	RO	0x0	reserved
7:6	RW	0x0	clk_vop1_pwm_src_sel vop1_pwm_src clock select control register 2'b00:VPLL 2'b01:CPLL 2'b1x:GPLL
5	RO	0x0	reserved
4:0	RW	0x05	clk_vop1_pwm_div_con vop1_pwm divider control register clk=clk_src/(div_con+1)

CRU_CLKSEL_CON53
Address: Operational Base + offset (0x01d4)
Internal clock select and divide register53

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:13	RO	0x0	reserved

Bit	Attr	Reset Value	Description
12:8	RW	0x01	hclk_isp0_div_con hclk_isp0 divider control register clk=clk_src/(div_con+1)
7:6	RW	0x0	aclk_isp0_pll_sel aclk_isp0 clock source select control register 2'b00:CPLL 2'b01:GPLL 2'b10:PPLL
5	RO	0x0	reserved
4:0	RW	0x01	aclk_isp0_div_con aclk_isp0 divider control register clk=clk_src/(div_con+1)

CRU_CLKSEL_CON54
Address: Operational Base + offset (0x01d8)
Internal clock select and divide register54

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:13	RO	0x0	reserved
12:8	RW	0x01	hclk_isp1_div_con hclk_isp1 divider control register clk=clk_src/(div_con+1)
7:6	RW	0x0	aclk_isp1_pll_sel aclk_isp1 clock source select control register 2'b00:CPLL 2'b01:GPLL 2'b10:PPLL
5	RO	0x0	reserved
4:0	RW	0x01	aclk_isp1_div_con aclk_isp1 divider control register clk=clk_src/(div_con+1)

CRU_CLKSEL_CON55
Address: Operational Base + offset (0x01dc)
Internal clock select and divide register55

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit

Bit	Attr	Reset Value	Description
15:14	RW	0x2	clk_isp1_pll_sel clk_isp1 clock source select control register 2'b00:CPLL 2'b01:GPLL 2'b1x:NPLL
13	RO	0x0	reserved
12:8	RW	0x01	clk_isp1_div_con clk_isp1 divider control register clk=clk_src/(div_con+1)
7:6	RW	0x2	clk_isp0_pll_sel clk_isp0 clock source select control register 2'b00:CPLL 2'b01:GPLL 2'b1x:NPLL
5	RO	0x0	reserved
4:0	RW	0x01	clk_isp0_div_con clk_isp0 divider control register clk=clk_src/(div_con+1)

CRU_CLKSEL_CON56
Address: Operational Base + offset (0x01e0)
Internal clock select and divide register56

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	aclk_gic_pll_sel aclk_gic source select control register 1'b0:CPLL 1'b1:GPLL
14:13	RO	0x0	reserved
12:8	RW	0x03	aclk_gic_div_con aclk_gic divider control register clk=clk_src/(div_con+1)
7:6	RW	0x0	clk_cif_pll_sel clk_cif clock source select control register 2'b00:CPLL 2'b01:GPLL 2'b1x:NPLL
5	RW	0x1	clk_cif_clk_sel clk_cif clock select control register 1'b0:clk_cif_src 1'b1:xin_24m

Bit	Attr	Reset Value	Description
4:0	RW	0x00	clk_cif_div_con clk_cif divider control register clk=clk_src/(div_con+1)

CRU_CLKSEL_CON57
Address: Operational Base + offset (0x01e4)
Internal clock select and divide register57

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:6	RW	0x000	clkout_24m_div_con clkout_24m divider control register clk=clk_src/(div_con+1)
5	RO	0x0	reserved
4:0	RW	0x05	pclk_alive_div_con pclk_alive divider control register clk=clk_src/(div_con+1)

CRU_CLKSEL_CON58
Address: Operational Base + offset (0x01e8)
Internal clock select and divide register58

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	clk_spi5_pll_sel clk_spi5 clock source select control register 1'b0:CPLL 1'b1:GPLL
14:8	RW	0x07	clk_spi5_div_con spi5 divider control register clk=clk_src/(div_con+1)
7	RW	0x0	clk_testfrac_pll_sel clk_frac clock source select control register 1'b0:CPLL 1'b1:GPLL
6:5	RO	0x0	reserved
4:0	RW	0x1f	clk_test_div_con test divider control register clk=clk_src/(div_con+1)

CRU_CLKSEL_CON59
Address: Operational Base + offset (0x01ec)
Internal clock select and divide register59

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	clk_spi1_pll_sel clk_spi1 clock source select control register 1'b0:CPLL 1'b1:GPLL
14:8	RW	0x07	clk_spi1_div_con spi1 divider control register clk=clk_src/(div_con+1)
7	RW	0x0	clk_spi0_pll_sel clk_spi0 clock source select control register 1'b0:CPLL 1'b1:GPLL
6:0	RW	0x07	clk_spi0_div_con spi0 divider control register clk=clk_src/(div_con+1)

CRU_CLKSEL_CON60
Address: Operational Base + offset (0x01f0)
Internal clock select and divide register60

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	clk_spi4_pll_sel clk_spi4 clock source select control register 1'b0:CPLL 1'b1:GPLL
14:8	RW	0x07	clk_spi4_div_con spi4 divider control register clk=clk_src/(div_con+1)
7	RW	0x0	clk_spi2_pll_sel clk_spi2 clock source select control register 1'b0:CPLL 1'b1:GPLL
6:0	RW	0x07	clk_spi2_div_con spi2 divider control register clk=clk_src/(div_con+1)

CRU_CLKSEL_CON61
Address: Operational Base + offset (0x01f4)
Internal clock select and divide register61

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	clk_i2c5_pll_sel clk_i2c5 clock source select control register 1'b0:CPLL 1'b1:GPLL
14:8	RW	0x03	clk_i2c5_div_con i2c5 divider control register clk=clk_src/(div_con+1)
7	RW	0x0	clk_i2c1_pll_sel clk_i2c1 clock source select control register 1'b0:CPLL 1'b1:GPLL
6:0	RW	0x03	clk_i2c1_div_con i2c1 divider control register clk=clk_src/(div_con+1)

CRU_CLKSEL_CON62
Address: Operational Base + offset (0x01f8)
Internal clock select and divide register62

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	clk_i2c6_pll_sel clk_i2c6 clock source select control register 1'b0:CPLL 1'b1:GPLL
14:8	RW	0x03	clk_i2c6_div_con i2c6 divider control register clk=clk_src/(div_con+1)
7	RW	0x0	clk_i2c2_pll_sel clk_i2c2 clock source select control register 1'b0:CPLL 1'b1:GPLL
6:0	RW	0x03	clk_i2c2_div_con i2c2 divider control register clk=clk_src/(div_con+1)

CRU_CLKSEL_CON63
Address: Operational Base + offset (0x01fc)
Internal clock select and divide register63

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	clk_i2c7_pll_sel clk_i2c7 clock source select control register 1'b0:CPLL 1'b1:GPLL
14:8	RW	0x03	clk_i2c7_div_con i2c7 divider control register clk=clk_src/(div_con+1)
7	RW	0x0	clk_i2c3_pll_sel clk_i2c3 clock source select control register 1'b0:CPLL 1'b1:GPLL
6:0	RW	0x03	clk_i2c3_div_con i2c3 divider control register clk=clk_src/(div_con+1)

CRU_CLKSEL_CON64
Address: Operational Base + offset (0x0200)
Internal clock select and divide register64

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	clk_uphy0_tcpdphy_ref_clk_sel clk_uphy0_tcpdphy_ref clock select control register 1'b0:xin_24m 1'b1:clk_32k
14:13	RO	0x0	reserved
12:8	RW	0x00	clk_uphy0_tcpdphy_ref_div_con clk_uphy0_tcpdphy_ref divider control register clk=clk_src/(div_con+1)
7:6	RW	0x3	clk_uphy0_tcpdcore_clk_sel clk_uphy0_tcpdcore clock select control register 2'b00:xin_24m 2'b01:clk_32k 2'b10:cpll 2'b11:gpll
5	RO	0x0	reserved

Bit	Attr	Reset Value	Description
4:0	RW	0x05	clk_uphy0_tcpdcore_div_con clk_uphy0_tcpdcore divider control register clk=clk_src/(div_con+1)

CRU_CLKSEL_CON65
Address: Operational Base + offset (0x0204)
Internal clock select and divide register65

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	clk_uphy1_tcpdphy_ref_clk_sel clk_uphy1_tcpdphy_ref clock select control register 1'b0:xin_24m 1'b1:clk_32k
14:13	RO	0x0	reserved
12:8	RW	0x00	clk_uphy1_tcpdphy_ref_div_con clk_uphy1_tcpdphy_ref divider control register clk=clk_src/(div_con+1)
7:6	RW	0x3	clk_uphy1_tcpdcore_clk_sel clk_uphy1_tcpdcore clock select control register 2'b00:xin_24m 2'b01:clk_32k 2'b10:cpll 2'b11:gpll
5	RO	0x0	reserved
4:0	RW	0x05	clk_uphy1_tcpdcore_div_con clk_uphy1_tcpdcore divider control register clk=clk_src/(div_con+1)

CRU_CLKSEL_CON96
Address: Operational Base + offset (0x0280)
Internal clock select and divide register80

Bit	Attr	Reset Value	Description
31:0	RW	0x0bb8ea60	clk_i2s0_frac_div_con clk_i2s0_frac divider control register Fout = Fsrc*numerator/denominator High 16-bit for numerator Low 16-bit for denominator

CRU_CLKSEL_CON97
Address: Operational Base + offset (0x0284)
Internal clock select and divide register81

Bit	Attr	Reset Value	Description
31:0	RW	0x0bb8ea60	clk_i2s1_frac_div_con clk_i2s1_frac divider control register Fout = Fsrc*numerator/denominator High 16-bit for numerator Low 16-bit for denominator

CRU_CLKSEL_CON98
Address: Operational Base + offset (0x0288)
Internal clock select and divide register82

Bit	Attr	Reset Value	Description
31:0	RW	0x0bb8ea60	clk_i2s2_frac_div_con clk_i2s2_frac divider control register Fout = Fsrc*numerator/denominator High 16-bit for numerator Low 16-bit for denominator

CRU_CLKSEL_CON99
Address: Operational Base + offset (0x028c)
Internal clock select and divide register83

Bit	Attr	Reset Value	Description
31:0	RW	0x0bb8ea60	clk_spdif_8ch_frac_div_con spdif_8ch_frac divider control register Fout = Fsrc*numerator/denominator High 16-bit for numerator Low 16-bit for denominator

CRU_CLKSEL_CON100
Address: Operational Base + offset (0x0290)
Internal clock select and divide register84

Bit	Attr	Reset Value	Description
31:0	RW	0x0bb8ea60	clk_uart0_frac_div_con uart0_frac divider control register Fout = Fsrc*numerator/denominator High 16-bit for numerator Low 16-bit for denominator

CRU_CLKSEL_CON101
Address: Operational Base + offset (0x0294)
Internal clock select and divide register85

Bit	Attr	Reset Value	Description
31:0	RW	0x0bb8ea60	clk_uart1_frac_div_con uart1_frac divider control register Fout = Fsrc*numerator/denominator High 16-bit for numerator Low 16-bit for denominator

CRU_CLKSEL_CON102
Address: Operational Base + offset (0x0298)
Internal clock select and divide register86

Bit	Attr	Reset Value	Description
31:0	RW	0x0bb8ea60	clk_uart2_frac_div_con uart2_frac divider control register Fout = Fsrc*numerator/denominator High 16-bit for numerator Low 16-bit for denominator

CRU_CLKSEL_CON103
Address: Operational Base + offset (0x029c)
Internal clock select and divide register87

Bit	Attr	Reset Value	Description
31:0	RW	0x0bb8ea60	clk_uart3_frac_div_con uart3_frac divider control register Fout = Fsrc*numerator/denominator High 16-bit for numerator Low 16-bit for denominator

CRU_CLKSEL_CON105
Address: Operational Base + offset (0x02a4)
Internal clock select and divide register89

Bit	Attr	Reset Value	Description
31:0	RW	0x0bb8ea60	clk_testfrac_frac_div_con clk_testfrac frac divider control register Fout = Fsrc*numerator/denominator High 16-bit for numerator Low 16-bit for denominator

CRU_CLKSEL_CON106
Address: Operational Base + offset (0x02a8)
Internal clock select and divide register90

Bit	Attr	Reset Value	Description
31:0	RW	0x0bb8ea60	dclk_vop0_frac_div_con dclk_vop0_frac divider control register Fout = Fsrc*numerator/denominator High 16-bit for numerator Low 16-bit for denominator

CRU_CLKSEL_CON107
Address: Operational Base + offset (0x02ac)
Internal clock select and divide register91

Bit	Attr	Reset Value	Description
31:0	RW	0x0bb8ea60	dclk_vop1_frac_div_con dclk_vop1_frac divider control register Fout = Fsrc*numerator/denominator High 16-bit for numerator Low 16-bit for denominator

CRU_CLKGATE_CON0
Address: Operational Base + offset (0x0300)
Internal clock gating register0

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:8	RO	0x0	reserved
7	RW	0x0	clk_pvtm_core_l_en clk_pvtm_core_l clock disable bit When HIGH, disable clock
6	RW	0x0	pclk_coredbg_l_en pclk_coredbg_l clock disable bit When HIGH, disable clock
5	RW	0x0	atclk_core_l_en atclk_core_l clock disable bit When HIGH, disable clock
4	RW	0x0	aclkm_core_l_en aclkm_core_l clock disable bit When HIGH, disable clock
3	RW	0x0	clk_core_l_gpll_src_en clk_core_l_gpll clock disable bit When HIGH, disable clock
2	RW	0x0	clk_core_l_dpll_src_en clk_core_l_dpll clock disable bit When HIGH, disable clock
1	RW	0x0	clk_core_l_bpll_src_en clk_core_l_bpll clock disable bit When HIGH, disable clock
0	RW	0x0	clk_core_l_lpll_src_en clk_core_l_lpll clock disable bit When HIGH, disable clock

CRU_CLKGATE_CON1
Address: Operational Base + offset (0x0304)
Internal clock gating register1

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:8	RO	0x0	reserved
7	RW	0x0	clk_pvtm_core_b_en clk_pvtm_core_b clock disable bit When HIGH, disable clock
6	RW	0x0	pclk_coredbg_b_en pclk_coredbg_b clock disable bit When HIGH, disable clock
5	RW	0x0	atclk_core_b_en atclk_core_b clock disable bit When HIGH, disable clock
4	RW	0x0	aclkm_core_b_en aclkm_core_b clock disable bit When HIGH, disable clock
3	RW	0x0	clk_core_b_gpll_src_en clk_core_b_gpll clock disable bit When HIGH, disable clock
2	RW	0x0	clk_core_b_dpll_src_en clk_core_b_dpll clock disable bit When HIGH, disable clock
1	RW	0x0	clk_core_b_bpll_src_en clk_core_b_bpll clock disable bit When HIGH, disable clock
0	RW	0x0	clk_core_b_lpll_src_en clk_core_b_lpll clock disable bit When HIGH, disable clock

CRU_CLKGATE_CON2

Address: Operational Base + offset (0x0308)

Internal clock gating register2

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:11	RO	0x0	reserved
10	RW	0x0	cs_npll_clk_en cs_npll_clk clock disable bit When HIGH, disable clock
9	RW	0x0	cs_gpll_clk_en cs_gpll_clk clock disable bit When HIGH, disable clock

Bit	Attr	Reset Value	Description
8	RW	0x0	cs_cpll_clk_en cs_cpll_clk clock disable bit When HIGH, disable clock
7	RW	0x0	clk_cci_trace_en clk_cci_trace clock disable bit When HIGH, disable clock
6	RW	0x0	clk_cci_trace_gpll_src_en clk_cci_trace_gpll clock disable bit When HIGH, disable clock
5	RW	0x0	clk_cci_trace_cpll_src_en clk_cci_trace_cpll clock disable bit When HIGH, disable clock
4	RW	0x0	aclk_cci_src_en aclk_cci_src clock disable bit When HIGH, disable clock
3	RW	0x0	aclk_cci_vpll_src_en aclk_cci_vpll clock disable bit When HIGH, disable clock
2	RW	0x0	aclk_cci_npll_src_en aclk_cci_npll clock disable bit When HIGH, disable clock
1	RW	0x0	aclk_cci_gpll_src_en aclk_cci_gpll clock disable bit When HIGH, disable clock
0	RW	0x0	aclk_cci_cpll_src_en aclk_cci_cpll clock disable bit When HIGH, disable clock

CRU_CLKGATE_CON3

Address: Operational Base + offset (0x030c)

Internal clock gating register3

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:8	RO	0x0	reserved
7	RW	0x0	aclk_center_src_en aclk_center_src clock disable bit When HIGH, disable clock
6	RW	0x0	reserved
5	RW	0x0	reserved
4	RW	0x0	pclk_ddr_en pclk_ddr clock disable bit When HIGH, disable clock

Bit	Attr	Reset Value	Description
3	RW	0x0	clk_ddrc_gpll_src_en clk_ddrc_gpll clock disable bit When HIGH, disable clock
2	RW	0x0	clk_ddrc_dpll_src_en clk_ddrc_dpll clock disable bit When HIGH, disable clock
1	RW	0x0	clk_ddrc_bpll_src_en clk_ddrc_bpll clock disable bit When HIGH, disable clock
0	RW	0x0	clk_ddrc_lpll_src_en clk_ddrc_lpll clock disable bit When HIGH, disable clock

CRU_CLKGATE_CON4
Address: Operational Base + offset (0x0310)
Internal clock gating register4

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:12	RO	0x0	reserved
11	RW	0x0	clk_pvtm_ddr_en clk_pvtm_ddr clock disable bit When HIGH, disable clock
10	RW	0x0	clk_rga_core_src_en clk_rga_core_src clock disable bit When HIGH, disable clock
9	RW	0x0	hclk_rga_src_en hclk_rga_src clock disable bit When HIGH, disable clock
8	RW	0x0	aclk_rga_src_en aclk_rga_src clock disable bit When HIGH, disable clock
7	RW	0x0	hclk_iep_src_en hclk_iep_src clock disable bit When HIGH, disable clock
6	RW	0x0	aclk_iep_src_en aclk_iep_src clock disable bit When HIGH, disable clock
5	RW	0x0	clk_vdu_ca_src_en clk_vdu_ca_src clock disable bit When HIGH, disable clock

Bit	Attr	Reset Value	Description
4	RW	0x0	clk_vdu_core_src_en clk_vdu_core_src clock disable bit When HIGH, disable clock
3	RW	0x0	hclk_vdu_src_en hclk_vdu_src clock disable bit When HIGH, disable clock
2	RW	0x0	aclk_vdu_src_en aclk_vdu_src clock disable bit When HIGH, disable clock
1	RW	0x0	hclk_vcodec_src_en hclk_vcodec_src clock disable bit When HIGH, disable clock
0	RW	0x0	aclk_vcodec_src_en aclk_vcodec_src clock disable bit When HIGH, disable clock

CRU_CLKGATE_CON5

Address: Operational Base + offset (0x0314)

Internal clock gating register5

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:10	RO	0x0	reserved
9	RW	0x0	clk_mac_tx_en clk_mac_tx clock disable bit When HIGH, disable clock
8	RW	0x0	clk_mac_rx_en clk_mac_rx clock disable bit When HIGH, disable clock
7	RW	0x0	clk_mac_refout_en clk_mac_refout clock disable bit When HIGH, disable clock
6	RW	0x0	clk_mac_ref_en clk_mac_ref clock disable bit When HIGH, disable clock
5	RW	0x0	clk_gmac_src_en clk_gmac_src clock disable bit When HIGH, disable clock
4	RW	0x0	pclk_perihp_en pclk_perihp clock disable bit When HIGH, disable clock

Bit	Attr	Reset Value	Description
3	RW	0x0	hclk_perihp_en hclk_perihp clock disable bit When HIGH, disable clock
2	RW	0x0	aclk_perihp_en aclk_perihp clock disable bit When HIGH, disable clock
1	RW	0x0	aclk_perihp_cpll_src_en aclk_perihp_cpll clock disable bit When HIGH, disable clock
0	RW	0x0	aclk_perihp_gpll_src_en aclk_perihp_gpll clock disable bit When HIGH, disable clock

CRU_CLKGATE_CON6
Address: Operational Base + offset (0x0318)
Internal clock gating register6

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RO	0x0	reserved
14	RW	0x0	clk_emmc_src_en clk_emmc_src clock disable bit When HIGH, disable clock
13	RW	0x0	aclk_emmc_cpll_src_en aclk_emmc_cpll clock disable bit When HIGH, disable clock
12	RW	0x0	aclk_emmc_gpll_src_en aclk_emmc_gpll clock disable bit When HIGH, disable clock
11	RW	0x0	pclk_gmac_en pclk_gmac clock disable bit When HIGH, disable clock
10	RW	0x0	aclk_gmac_en aclk_gmac clock disable bit When HIGH, disable clock
9	RW	0x0	aclk_gmac_cpll_src_en aclk_gmac_cpll clock disable bit When HIGH, disable clock
8	RW	0x0	aclk_gmac_gpll_src_en aclk_gmac_gpll clock disable bit When HIGH, disable clock
7	RO	0x0	reserved

Bit	Attr	Reset Value	Description
6	RW	0x0	clk_usb2phy1_ref_en clk_usb2phy1_ref clock disable bit When HIGH, disable clock
5	RW	0x0	clk_usb2phy0_ref_en clk_usb2phy0_ref clock disable bit When HIGH, disable clock
4	RW	0x0	clk_hsicphy_en clk_hsicphy clock disable bit When HIGH, disable clock
3	RW	0x0	clk_pcie_core_src_en clk_pcie_core_src clock disable bit When HIGH, disable clock
2	RW	0x0	clk_pcie_pm_src_en clk_pcie_pm_src clock disable bit When HIGH, disable clock
1	RW	0x0	clk_sdmmc_src_en clk_sdmmc_src clock disable bit When HIGH, disable clock
0	RW	0x0	clk_sdio_src_en clk_sdio_src clock disable bit When HIGH, disable clock

CRU_CLKGATE_CON7

Address: Operational Base + offset (0x031c)

Internal clock gating register7

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:10	RO	0x0	reserved
9	RW	0x0	fclk_cm0s_en fclk_cm0s clock disable bit When HIGH, disable clock
8	RW	0x0	clk_crypto1_en clk_crypto1 clock disable bit When HIGH, disable clock
7	RW	0x0	clk_crypto0_en clk_crypto0 clock disable bit When HIGH, disable clock
6	RW	0x0	clk_cm0s_cpll_src_en clk_cm0s_cpll clock disable bit When HIGH, disable clock

Bit	Attr	Reset Value	Description
5	RW	0x0	clk_cm0s_gpll_src_en clk_cm0s_gpll clock disable bit When HIGH, disable clock
4	RW	0x0	pclk_perilp0_en pclk_perilp0 clock disable bit When HIGH, disable clock
3	RW	0x0	hclk_perilp0_en hclk_perilp0 clock disable bit When HIGH, disable clock
2	RW	0x0	aclk_perilp0_en aclk_perilp0 clock disable bit When HIGH, disable clock
1	RW	0x0	aclk_perilp0_cpll_src_en aclk_perilp0_cpll clock disable bit When HIGH, disable clock
0	RW	0x0	aclk_perilp0_gpll_src_en aclk_perilp0_gpll clock disable bit When HIGH, disable clock

CRU_CLKGATE_CON8

Address: Operational Base + offset (0x0320)

Internal clock gating register8

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	clk_spdif_8ch_en clk_spdif_8ch clock disable bit When HIGH, disable clock
14	RW	0x0	clk_spdif_8ch_frac_src_en clk_spdif_8ch_frac_src clock disable bit When HIGH, disable clock
13	RW	0x0	clk_spdif_8ch_src_en clk_spdif_8ch_src clock disable bit When HIGH, disable clock
12	RW	0x0	clk_i2s_out_en clk_i2s_out clock disable bit When HIGH, disable clock
11	RW	0x0	clk_i2s2_en clk_i2s2 clock disable bit When HIGH, disable clock
10	RW	0x0	clk_i2s2_frac_src_en clk_i2s2_frac_src clock disable bit When HIGH, disable clock

Bit	Attr	Reset Value	Description
9	RW	0x0	clk_i2s2_src_en clk_i2s2_src clock disable bit When HIGH, disable clock
8	RW	0x0	clk_i2s1_en clk_i2s1 clock disable bit When HIGH, disable clock
7	RW	0x0	clk_i2s1_frac_src_en clk_i2s1_frac_src clock disable bit When HIGH, disable clock
6	RW	0x0	clk_i2s1_src_en clk_i2s1_src clock disable bit When HIGH, disable clock
5	RW	0x0	clk_i2s0_en clk_i2s0 clock disable bit When HIGH, disable clock
4	RW	0x0	clk_i2s0_frac_src_en clk_i2s0_frac_src clock disable bit When HIGH, disable clock
3	RW	0x0	clk_i2s0_src_en clk_i2s0_src clock disable bit When HIGH, disable clock
2	RW	0x0	pclk_perilp1_en pclk_perilp1 clock disable bit When HIGH, disable clock
1	RW	0x0	hclk_perilp1_cpll_src_en hclk_perilp1_cpll clock disable bit When HIGH, disable clock
0	RW	0x0	hclk_perilp1_gpll_src_en hclk_perilp1_gpll clock disable bit When HIGH, disable clock

CRU_CLKGATE_CON9

Address: Operational Base + offset (0x0324)

Internal clock gating register9

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	clk_spi4_src_en clk_spi4_src clock disable bit When HIGH, disable clock
14	RW	0x0	clk_spi2_src_en clk_spi2_src clock disable bit When HIGH, disable clock

Bit	Attr	Reset Value	Description
13	RW	0x0	clk_spi1_src_en clk_spi1_src clock disable bit When HIGH, disable clock
12	RW	0x0	clk_spi0_src_en clk_spi0_src clock disable bit When HIGH, disable clock
11	RW	0x0	clk_saradc_src_en clk_saradc_src clock disable bit When HIGH, disable clock
10	RW	0x0	clk_tsadc_src_en clk_tsadc_src clock disable bit When HIGH, disable clock
9:8	RO	0x0	reserved
7	RW	0x0	clk_uart3_frac_src_en clk_uart3_frac_src clock disable bit When HIGH, disable clock
6	RW	0x0	clk_uart3_src_en clk_uart3_src clock disable bit When HIGH, disable clock
5	RW	0x0	clk_uart2_frac_src_en clk_uart2_frac_src clock disable bit When HIGH, disable clock
4	RW	0x0	clk_uart2_src_en clk_uart2_src clock disable bit When HIGH, disable clock
3	RW	0x0	clk_uart1_frac_src_en clk_uart1_frac_src clock disable bit When HIGH, disable clock
2	RW	0x0	clk_uart1_src_en clk_uart1_src clock disable bit When HIGH, disable clock
1	RW	0x0	clk_uart0_frac_src_en clk_uart0_frac_src clock disable bit When HIGH, disable clock
0	RW	0x0	clk_uart0_src_en clk_uart0_src clock disable bit When HIGH, disable clock

CRU_CLKGATE_CON10
Address: Operational Base + offset (0x0328)
Internal clock gating register10

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	clk_vop1_pwm_en clk_vop1_pwm clock disable bit When HIGH, disable clock
14	RW	0x0	clk_vop0_pwm_en clk_vop0_pwm clock disable bit When HIGH, disable clock
13	RW	0x0	dclk_vop1_src_en dclk_vop1_src clock disable bit When HIGH, disable clock
12	RW	0x0	dclk_vop0_src_en dclk_vop0_src clock disable bit When HIGH, disable clock
11	RW	0x0	hclk_vop1_pre_en hclk_vop1_pre clock disable bit When HIGH, disable clock
10	RW	0x0	aclk_vop1_pre_src_en aclk_vop1_pre_src clock disable bit When HIGH, disable clock
9	RW	0x0	hclk_vop0_pre_en hclk_vop0_pre clock disable bit When HIGH, disable clock
8	RW	0x0	aclk_vop0_pre_src_en aclk_vop0_pre_src clock disable bit When HIGH, disable clock
7	RW	0x0	clk_cif_out_src_en clk_cif_out_src clock disable bit When HIGH, disable clock
6	RW	0x0	clk_dptx_spdif_rec_src_en clk_dptx_spdif_rec_src clock disable bit When HIGH, disable clock
5	RW	0x0	clk_i2c7_src_en clk_i2c7_src clock disable bit When HIGH, disable clock
4	RW	0x0	clk_i2c3_src_en clk_i2c3_src clock disable bit When HIGH, disable clock
3	RW	0x0	clk_i2c6_src_en clk_i2c6_src clock disable bit When HIGH, disable clock

Bit	Attr	Reset Value	Description
2	RW	0x0	clk_i2c2_src_en clk_i2c2_src clock disable bit When HIGH, disable clock
1	RW	0x0	clk_i2c5_src_en clk_i2c5_src clock disable bit When HIGH, disable clock
0	RW	0x0	clk_i2c1_src_en clk_i2c1_src clock disable bit When HIGH, disable clock

CRU_CLKGATE_CON11

Address: Operational Base + offset (0x032c)

Internal clock gating register11

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	clk_mipidphy_cfg_en clk_mipidphy_cfg clock disable bit When HIGH, disable clock
14	RW	0x0	clk_mipidphy_ref_en clk_mipidphy_ref clock disable bit When HIGH, disable clock
13:12	RO	0x0	reserved
11	RW	0x0	pclk_edp_en pclk_edp clock disable bit When HIGH, disable clock
10	RW	0x0	pclk_hdcp_en pclk_hdcp clock disable bit When HIGH, disable clock
9	RO	0x0	reserved
8	RW	0x0	clk_dp_core_src_en clk_dp_core_src clock disable bit When HIGH, disable clock
7	RW	0x0	clk_hdmi_cec_en clk_hdmi_cec clock disable bit When HIGH, disable clock
6	RW	0x0	clk_hdmi_sfr_en clk_hdmi_sfr clock disable bit When HIGH, disable clock
5	RW	0x0	clk_isp1_en clk_isp1 clock disable bit When HIGH, disable clock

Bit	Attr	Reset Value	Description
4	RW	0x0	clk_isp0_en clk_isp0 clock disable bit When HIGH, disable clock
3	RW	0x0	hclk_hdcp_en hclk_hdcp clock disable bit When HIGH, disable clock
2	RW	0x0	aclk_hdcp_src_en aclk_hdcp_src clock disable bit When HIGH, disable clock
1	RW	0x0	pclk_vio_en pclk_vio clock disable bit When HIGH, disable clock
0	RW	0x0	aclk_vio_src_en aclk_vio_src clock disable bit When HIGH, disable clock

CRU_CLKGATE_CON12

Address: Operational Base + offset (0x0330)

Internal clock gating register12

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:14	RO	0x0	reserved
13	RW	0x0	hclk_sd_src_en hclk_sd_src clock disable bit When HIGH, disable clock
12	RW	0x0	aclk_gic_src_en aclk_gic_src clock disable bit When HIGH, disable clock
11	RW	0x0	hclk_isp1_en hclk_isp1 clock disable bit When HIGH, disable clock
10	RW	0x0	aclk_isp1_src_en aclk_isp1_src clock disable bit When HIGH, disable clock
9	RW	0x0	hclk_isp0_en hclk_isp0 clock disable bit When HIGH, disable clock
8	RW	0x0	aclk_isp0_src_en aclk_isp0_src clock disable bit When HIGH, disable clock
7	RO	0x0	reserved

Bit	Attr	Reset Value	Description
6	RW	0x0	clk_pciephy_ref100m_en clk_pciephy_ref100m clock disable bit When HIGH, disable clock
5	RO	0x0	reserved
4	RW	0x0	clk_usb3_otg1_suspend_en clk_usb3_otg1_suspend clock disable bit When HIGH, disable clock
3	RW	0x0	clk_usb3_otg0_suspend_en clk_usb3_otg0_suspend clock disable bit When HIGH, disable clock
2	RW	0x0	clk_usb3_otg1_ref_en clk_usb3_otg1_ref clock disable bit When HIGH, disable clock
1	RW	0x0	clk_usb3_otg0_ref_en clk_usb3_otg0_ref clock disable bit When HIGH, disable clock
0	RW	0x0	aclk_usb3_src_en aclk_usb3_src clock disable bit When HIGH, disable clock

CRU_CLKGATE_CON13
Address: Operational Base + offset (0x0334)
Internal clock gating register13

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	clk_testout2_src_en clk_testout2_src clock disable bit When HIGH, disable clock
14	RW	0x0	clk_testout1_src_en clk_testout1_src clock disable bit When HIGH, disable clock
13	RW	0x0	clk_spi5_src_en clk_spi5_src clock disable bit When HIGH, disable clock
12	RW	0x0	clk_usb480m_en clk_usb480m clock disable bit When HIGH, disable clock
11	RW	0x0	testclk_en testclk clock disable bit When HIGH, disable clock
10	RO	0x0	reserved

Bit	Attr	Reset Value	Description
9	RW	0x0	clk_wifi_en clk_wifi clock disable bit When HIGH, disable clock
8	RO	0x0	reserved
7	RW	0x0	clk_uphy1_tcpdcore_en clk_uphy1_tcpdcore clock disable bit When HIGH, disable clock
6	RW	0x0	clk_uphy1_tcpdphyref_en clk_uphy1_tcpdphyref clock disable bit When HIGH, disable clock
5	RW	0x0	clk_uphy0_tcpdcore_en clk_uphy0_tcpdcore clock disable bit When HIGH, disable clock
4	RW	0x0	clk_uphy0_tcpdphyref_en clk_uphy0_tcpdphyref clock disable bit When HIGH, disable clock
3:2	RO	0x0	reserved
1	RW	0x0	clk_pvtm_gpu_en clk_pvtm_gpu clock disable bit When HIGH, disable clock
0	RW	0x0	aclk_gpu_pll_src_en aclk_gpu_pll_src clock disable bit When HIGH, disable clock

CRU_CLKGATE_CON14

Address: Operational Base + offset (0x0338)

Internal clock gating register14

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:14	RO	0x0	reserved
13	RW	0x0	aclk_perf_core_l_en aclk_perf_core_l clock disable bit When HIGH, disable clock
12	RW	0x0	aclk_core_adb400_core_l_2_cci500_en aclk_core_adb400_core_l_2_cci500 clock disable bit When HIGH, disable clock
11	RW	0x0	aclk_core_adb400_core_l_2_gic_en aclk_core_adb400_core_l_2_gic clock disable bit When HIGH, disable clock
10	RW	0x0	aclk_core_adb400_gic_2_core_l_en aclk_core_adb400_gic_2_core_l clock disable bit When HIGH, disable clock

Bit	Attr	Reset Value	Description
9	RW	0x0	clk_dbg_pd_core_l_en clk_dbg_pd_core_l clock disable bit When HIGH, disable clock
8:7	RO	0x0	reserved
6	RW	0x0	aclk_perf_core_b_en aclk_perf_core_b clock disable bit When HIGH, disable clock
5	RW	0x0	aclk_core_adb400_core_b_2_cci500_en aclk_core_adb400_core_b_2_cci500 clock disable bit When HIGH, disable clock
4	RW	0x0	aclk_core_adb400_core_b_2_gic_en aclk_core_adb400_core_b_2_gic clock disable bit When HIGH, disable clock
3	RW	0x0	aclk_core_adb400_gic_2_core_b_en aclk_core_adb400_gic_2_core_b clock disable bit When HIGH, disable clock
2	RW	0x0	pclk_dbg_cxcs_pd_core_b_en pclk_dbg_cxcs_pd_core_b clock disable bit When HIGH, disable clock
1	RW	0x0	clk_dbg_pd_core_b_en clk_dbg_pd_core_b clock disable bit When HIGH, disable clock
0	RO	0x0	reserved

CRU_CLKGATE_CON15

Address: Operational Base + offset (0x033c)

Internal clock gating register15

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:8	RO	0x0	reserved
7	RW	0x0	aclk_cci_grf_en aclk_cci_grf clock disable bit When HIGH, disable clock Suggest always on
6	RW	0x0	clk_dbg_noc_en clk_dbg_noc clock disable bit When HIGH, disable clock Suggest always on
5	RW	0x0	clk_dbg_cxcs_en clk_dbg_cxcs clock disable bit When HIGH, disable clock

Bit	Attr	Reset Value	Description
4	RW	0x0	aclk_cci_noc1_en aclk_cci_noc1 clock disable bit When HIGH, disable clock Suggest always on
3	RW	0x0	aclk_cci_noc0_en aclk_cci_noc0 clock disable bit When HIGH, disable clock Suggest always on
2	RW	0x0	aclk_cci_en aclk_cci clock disable bit When HIGH, disable clock
1	RW	0x0	aclk_adb400m_pd_core_b_en aclk_adb400m_pd_core_b clock disable bit When HIGH, disable clock
0	RW	0x0	aclk_adb400m_pd_core_l_en aclk_adb400m_pd_core_l clock disable bit When HIGH, disable clock

CRU_CLKGATE_CON16

Address: Operational Base + offset (0x0340)

Internal clock gating register16

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:12	RO	0x0	reserved
11	RW	0x0	hclk_rga_noc_en hclk_rga_noc clock disable bit When HIGH, disable clock Suggest always on
10	RW	0x0	hclk_rga_en hclk_rga clock disable bit When HIGH, disable clock
9	RW	0x0	aclk_rga_noc_en aclk_rga_noc clock disable bit When HIGH, disable clock Suggest always on
8	RW	0x0	aclk_rga_en aclk_rga clock disable bit When HIGH, disable clock
7:4	RO	0x0	reserved

Bit	Attr	Reset Value	Description
3	RW	0x0	hclk_iep_noc_en hclk_iep_noc clock disable bit When HIGH, disable clock Suggest always on
2	RW	0x0	hclk_iep_en hclk_iep clock disable bit When HIGH, disable clock
1	RW	0x0	aclk_iep_noc_en aclk_iep_noc clock disable bit When HIGH, disable clock Suggest always on
0	RW	0x0	aclk_iep_en aclk_iep clock disable bit When HIGH, disable clock

CRU_CLKGATE_CON17
Address: Operational Base + offset (0x0344)
Internal clock gating register17

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:12	RO	0x0	reserved
11	RW	0x0	hclk_vdu_noc_en hclk_vdu_noc clock disable bit When HIGH, disable clock Suggest always on
10	RW	0x0	hclk_vdu_en hclk_vdu clock disable bit When HIGH, disable clock
9	RW	0x0	aclk_vdu_noc_en aclk_vdu_noc clock disable bit When HIGH, disable clock Suggest always on
8	RW	0x0	aclk_vdu_en aclk_vdu clock disable bit When HIGH, disable clock
7:4	RO	0x0	reserved
3	RW	0x0	hclk_vcodec_noc_en hclk_vcodec_noc clock disable bit When HIGH, disable clock Suggest always on

Bit	Attr	Reset Value	Description
2	RW	0x0	hclk_vcodec_en hclk_vcodec clock disable bit When HIGH, disable clock
1	RW	0x0	aclk_vcodec_noc_en aclk_vcodec_noc clock disable bit When HIGH, disable clock Suggest always on
0	RW	0x0	aclk_vcodec_en aclk_vcodec clock disable bit When HIGH, disable clock

CRU_CLKGATE_CON18

Address: Operational Base + offset (0x0348)

Internal clock gating register18

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	pclk_cic_en pclk_cic clock disable bit When HIGH, disable clock
14	RW	0x0	clk_ddr_mon_timer_en clk_ddr_mon_timer clock disable bit When HIGH, disable clock
13	RW	0x0	clk_ddr_mon_en clk_ddr_mon clock disable bit When HIGH, disable clock
12	RW	0x0	pclk_ddr_mon_en pclk_ddr_mon clock disable bit When HIGH, disable clock
11	RW	0x0	clk_ddr_cic_en clk_ddr_cic clock disable bit When HIGH, disable clock
10	RW	0x0	pclk_center_main_noc_en pclk_center_main_noc clock disable bit When HIGH, disable clock Suggest always on
9	RW	0x0	clk_ddrcfg_msch1_en clk_ddrcfg_msch1 clock disable bit When HIGH, disable clock
8	RW	0x0	clk_ddrphy1_en clk_ddrphy1 clock disable bit When HIGH, disable clock

Bit	Attr	Reset Value	Description
7	RW	0x0	clk_ddrphy_ctrl1_en clk_ddrphy_ctrl1 clock disable bit When HIGH, disable clock
6	RW	0x0	clk_ddrc1_en clk_ddrc1 clock disable bit When HIGH, disable clock
5	RW	0x0	clk_ddr1_msch_en clk_ddr1_msch clock disable bit When HIGH, disable clock
4	RW	0x0	clk_ddrcfg_msch0_en clk_ddrcfg_msch0 clock disable bit When HIGH, disable clock
3	RW	0x0	clk_ddrphy0_en clk_ddrphy0 clock disable bit When HIGH, disable clock
2	RW	0x0	clk_ddrphy_ctrl0_en clk_ddrphy_ctrl0 clock disable bit When HIGH, disable clock
1	RW	0x0	clk_ddrc0_en clk_ddrc0 clock disable bit When HIGH, disable clock
0	RW	0x0	clk_ddr0_msch_en clk_ddr0_msch clock disable bit When HIGH, disable clock

CRU_CLKGATE_CON19
Address: Operational Base + offset (0x034c)
Internal clock gating register19

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:3	RO	0x0	reserved
2	RW	0x0	pclk_ddr_sgrf_en pclk_ddr_sgrf clock disable bit When HIGH, disable clock Suggest always on
1	RW	0x0	aclk_center_peri_noc_en aclk_center_peri_noc clock disable bit When HIGH, disable clock Suggest always on

Bit	Attr	Reset Value	Description
0	RW	0x0	aclk_center_main_noc_en aclk_center_main_noc clock disable bit When HIGH, disable clock Suggest always on

CRU_CLKGATE_CON20
Address: Operational Base + offset (0x0350)
Internal clock gating register20

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	hclk_ahb1tom_en hclk_ahb1tom clock disable bit When HIGH, disable clock
14	RW	0x0	pclk_perihp_noc_en pclk_perihp_noc clock disable bit When HIGH, disable clock Suggest always on
13	RW	0x0	hclk_perihp_noc_en hclk_perihp_noc clock disable bit When HIGH, disable clock Suggest always on
12	RW	0x0	aclk_perihp_noc_en aclk_perihp_noc clock disable bit When HIGH, disable clock Suggest always on
11	RW	0x0	pclk_pcie_en pclk_pcie clock disable bit When HIGH, disable clock
10	RW	0x0	aclk_pcie_en aclk_pcie clock disable bit When HIGH, disable clock
9	RW	0x0	hclk_hsic_en hclk_hsic clock disable bit When HIGH, disable clock
8	RW	0x0	hclk_host1_arb_en hclk_host1_arb clock disable bit When HIGH, disable clock
7	RW	0x0	hclk_host1_en hclk_host1 clock disable bit When HIGH, disable clock

Bit	Attr	Reset Value	Description
6	RW	0x0	hclk_host0_arb_en hclk_host0_arb clock disable bit When HIGH, disable clock
5	RW	0x0	hclk_host0_en hclk_host0 clock disable bit When HIGH, disable clock
4	RW	0x0	pclk_perihp_grf_en pclk_perihp_grf clock disable bit When HIGH, disable clock Suggest always on
3	RO	0x0	reserved
2	RW	0x0	aclk_perf_pcie_en aclk_perf_pcie clock disable bit When HIGH, disable clock
1:0	RO	0x0	reserved

CRU_CLKGATE_CON21
Address: Operational Base + offset (0x0354)
Internal clock gating register21

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:10	RO	0x0	reserved
9	RW	0x0	uphy1_pclk_tcpd_gate_en uphy1_pclk_tcpd disable bit When HIGH, disable clock
8	RW	0x0	uphy1_pclk_tcphy_gate_en uphy1_pclk_tcphy clock disable bit When HIGH, disable clock
7	RO	0x0	reserved
6	RW	0x0	uphy0_pclk_tcpd_gate_en uphy0_pclk_tcpd clock disable bit When HIGH, disable clock
5	RW	0x0	uphy0_pclk_tcphy_gate_en uphy0_pclk_tcphy clock disable bit When HIGH, disable clock
4	RW	0x0	uphy_pclk_mux_gate_en uphy_pclk_mux clock disable bit When HIGH, disable clock
3	RW	0x0	dphy_rx0_cfgclk_en dphy_rx0_cfg clock disable bit When HIGH, disable clock

Bit	Attr	Reset Value	Description
2	RW	0x0	dphy_tx1rx1_cfgclk_en dphy_tx1rx1_cfg clock disable bit When HIGH, disable clock
1	RW	0x0	dphy_tx0_cfgclk_en dphy_tx0_cfg clock disable bit When HIGH, disable clock
0	RW	0x0	dphy_pllclk_en dphy_pll clock disable bit When HIGH, disable clock

CRU_CLKGATE_CON22

Address: Operational Base + offset (0x0358)

Internal clock gating register22

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	pclk_efuse1024s_en pclk_efuse1024s clock disable bit When HIGH, disable clock
14	RW	0x0	pclk_efuse1024ns_en pclk_efuse1024ns clock disable bit When HIGH, disable clock
13	RW	0x0	pclk_tsadc_en pclk_tsadc clock disable bit When HIGH, disable clock
12	RW	0x0	pclk_saradc_en pclk_saradc clock disable bit When HIGH, disable clock
11	RW	0x0	pclk_mailbox0_en pclk_mailbox0 clock disable bit When HIGH, disable clock
10	RW	0x0	pclk_rki2c3_en pclk_rki2c3 clock disable bit When HIGH, disable clock
9	RW	0x0	pclk_rki2c2_en pclk_rki2c2 clock disable bit When HIGH, disable clock
8	RW	0x0	pclk_rki2c6_en pclk_rki2c6 clock disable bit When HIGH, disable clock
7	RW	0x0	pclk_rki2c5pad_en pclk_rki2c5pad clock disable bit When HIGH, disable clock

Bit	Attr	Reset Value	Description
6	RW	0x0	pclk_rki2c1cam_en pclk_rki2c1cam clock disable bit When HIGH, disable clock
5	RW	0x0	pclk_rki2c7_en pclk_rki2c7 clock disable bit When HIGH, disable clock
4	RO	0x0	reserved
3	RW	0x0	pclk_uart3_en pclk_uart3 clock disable bit When HIGH, disable clock
2	RW	0x0	pclk_uart2_en pclk_uart2 clock disable bit When HIGH, disable clock
1	RW	0x0	pclk_uart1_en pclk_uart1 clock disable bit When HIGH, disable clock
0	RW	0x0	pclk_uart0_en pclk_uart0 clock disable bit When HIGH, disable clock

CRU_CLKGATE_CON23
Address: Operational Base + offset (0x035c)
Internal clock gating register23

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:14	RO	0x0	reserved
13	RW	0x0	pclk_spi4_en pclk_spi4 clock disable bit When HIGH, disable clock
12	RW	0x0	pclk_spi2_en pclk_spi2 clock disable bit When HIGH, disable clock
11	RW	0x0	pclk_spi1_en pclk_spi1 clock disable bit When HIGH, disable clock
10	RW	0x0	pclk_spi0codec_en pclk_spi0codec clock disable bit When HIGH, disable clock
9	RW	0x0	pclk_dcf_en pclk_dcf clock disable bit When HIGH, disable clock

Bit	Attr	Reset Value	Description
8	RW	0x0	aclk_dcf_en aclk_dcf clock disable bit When HIGH, disable clock
7	RW	0x0	clk_intmem5_en clk_intmem5 clock disable bit When HIGH, disable clock
6	RW	0x0	clk_intmem4_en clk_intmem4 clock disable bit When HIGH, disable clock
5	RW	0x0	clk_intmem3_en clk_intmem3 clock disable bit When HIGH, disable clock
4	RW	0x0	clk_intmem2_en clk_intmem2 clock disable bit When HIGH, disable clock
3	RW	0x0	clk_intmem1_en clk_intmem1 clock disable bit When HIGH, disable clock
2	RW	0x0	clk_intmem0_en clk_intmem0 clock disable bit When HIGH, disable clock
1	RW	0x0	aclk_tzma_en aclk_tzma clock disable bit When HIGH, disable clock
0	RW	0x0	aclk_intmem_en aclk_intmem clock disable bit When HIGH, disable clock

CRU_CLKGATE_CON24

Address: Operational Base + offset (0x0360)

Internal clock gating register24

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	hclk_s_crypto1_en hclk_s_crypto1 clock disable bit When HIGH, disable clock
14	RW	0x0	hclk_m_crypto1_en hclk_m_crypto1 clock disable bit When HIGH, disable clock

Bit	Attr	Reset Value	Description
13	RW	0x0	pclk_perilp_sgrf_en pclk_perilp_sgrf clock disable bit When HIGH, disable clock Suggest always on
12	RO	0x0	reserved
11	RW	0x0	clk_m0_perilp_dec_en clk_m0_perilp_dec clock disable bit When HIGH, disable clock
10	RW	0x0	dclk_m0_perilp_en dclk_m0_perilp clock disable bit When HIGH, disable clock
9	RW	0x0	hclk_m0_perilp_en hclk_m0_perilp clock disable bit When HIGH, disable clock
8	RW	0x0	sclk_m0_perilp_en sclk_m0_perilp clock disable bit When HIGH, disable clock
7	RO	0x0	reserved
6	RW	0x0	hclk_s_crypto0_en hclk_s_crypto0 clock disable bit When HIGH, disable clock
5	RW	0x0	hclk_m_crypto0_en hclk_m_crypto0 clock disable bit When HIGH, disable clock
4	RW	0x0	hclk_rom_en hclk_rom clock disable bit When HIGH, disable clock
3:0	RO	0x0	reserved

CRU_CLKGATE_CON25

Address: Operational Base + offset (0x0364)

Internal clock gating register25

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:13	RO	0x0	reserved
12	RW	0x0	hclk_sdio_noc_en hclk_sdio_noc clock disable bit When HIGH, disable clock Suggest always on

Bit	Attr	Reset Value	Description
11	RW	0x0	hclk_m0_perilp_noc_en hclk_m0_perilp_noc clock disable bit When HIGH, disable clock Suggest always on
10	RW	0x0	pclk_perilp1_noc_en pclk_perilp1_noc clock disable bit When HIGH, disable clock Suggest always on
9	RW	0x0	hclk_perilp1_noc_en hclk_perilp1_noc clock disable bit When HIGH, disable clock Suggest always on
8	RW	0x0	hclk_perilp0_noc_en hclk_perilp0_noc clock disable bit When HIGH, disable clock Suggest always on
7	RW	0x0	aclk_perilp0_noc_en aclk_perilp0_noc clock disable bit When HIGH, disable clock Suggest always on
6	RW	0x0	aclk_dmac1_perilp_en aclk_dmac1_perilp clock disable bit When HIGH, disable clock
5	RW	0x0	aclk_dmac0_perilp_en aclk_dmac0_perilp clock disable bit When HIGH, disable clock
4:0	RO	0x0	reserved

CRU_CLKGATE_CON26

Address: Operational Base + offset (0x0368)

Internal clock gating register26

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:12	RO	0x0	reserved
11	RW	0x0	clk_timer11_en clk_timer11 clock disable bit When HIGH, disable clock
10	RW	0x0	clk_timer10_en clk_timer10 clock disable bit When HIGH, disable clock

Bit	Attr	Reset Value	Description
9	RW	0x0	clk_timer9_en clk_timer9 clock disable bit When HIGH, disable clock
8	RW	0x0	clk_timer8_en clk_timer8 clock disable bit When HIGH, disable clock
7	RW	0x0	clk_timer7_en clk_timer7 clock disable bit When HIGH, disable clock
6	RW	0x0	clk_timer6_en clk_timer6 clock disable bit When HIGH, disable clock
5	RW	0x0	clk_timer5_en clk_timer5 clock disable bit When HIGH, disable clock
4	RW	0x0	clk_timer4_en clk_timer4 clock disable bit When HIGH, disable clock
3	RW	0x0	clk_timer3_en clk_timer3 clock disable bit When HIGH, disable clock
2	RW	0x0	clk_timer2_en clk_timer2 clock disable bit When HIGH, disable clock
1	RW	0x0	clk_timer1_en clk_timer1 clock disable bit When HIGH, disable clock
0	RW	0x0	clk_timer0_en clk_timer0 clock disable bit When HIGH, disable clock

CRU_CLKGATE_CON27

Address: Operational Base + offset (0x036c)

Internal clock gating register27

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:9	RO	0x0	reserved
8	RW	0x0	aclk_isp1_wrapper_en aclk_isp1_wrapper clock disable bit When HIGH, disable clock

Bit	Attr	Reset Value	Description
7	RW	0x0	hclk_isp1_wrapper_en hclk_isp1_wrapper clock disable bit When HIGH, disable clock
6	RW	0x0	pclkin_isp1_wrapper_en pclkin_isp1_wrapper clock disable bit When HIGH, disable clock
5	RW	0x0	aclk_isp0_wrapper_en aclk_isp0_wrapper clock disable bit When HIGH, disable clock
4	RW	0x0	hclk_isp0_wrapper_en hclk_isp0_wrapper clock disable bit When HIGH, disable clock
3	RW	0x0	aclk_isp1_noc_en aclk_isp1_noc clock disable bit When HIGH, disable clock Suggest always on
2	RW	0x0	hclk_isp1_noc_en hclk_isp1_noc clock disable bit When HIGH, disable clock Suggest always on
1	RW	0x0	aclk_isp0_noc_en aclk_isp0_noc clock disable bit When HIGH, disable clock Suggest always on
0	RW	0x0	hclk_isp0_noc_en hclk_isp0_noc clock disable bit When HIGH, disable clock Suggest always on

CRU_CLKGATE_CON28

Address: Operational Base + offset (0x0370)

Internal clock gating register28

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:8	RO	0x0	reserved
7	RW	0x0	aclk_vopb_en aclk_vopb clock disable bit When HIGH, disable clock
6	RW	0x0	hclk_vopb_en hclk_vopb clock disable bit When HIGH, disable clock

Bit	Attr	Reset Value	Description
5	RW	0x0	aclk_vopb_noc_en aclk_vopb_noc clock disable bit When HIGH, disable clock Suggest always on
4	RW	0x0	hclk_vopb_noc_en hclk_vopb_noc clock disable bit When HIGH, disable clock Suggest always on
3	RW	0x0	aclk_vop0_en aclk_vop0 clock disable bit When HIGH, disable clock
2	RW	0x0	hclk_vop0_en hclk_vop0 clock disable bit When HIGH, disable clock
1	RW	0x0	aclk_vop0_noc_en aclk_vop0_noc clock disable bit When HIGH, disable clock Suggest always on
0	RW	0x0	hclk_vop0_noc_en hclk_vop0_noc clock disable bit When HIGH, disable clock Suggest always on

CRU_CLKGATE_CON29
Address: Operational Base + offset (0x0374)
Internal clock gating register29

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:13	RO	0x0	reserved
12	RW	0x0	pclk_vio_grf_en pclk_vio_grf clock disable bit When HIGH, disable clock Suggest always on
11	RW	0x0	pclk_gasket_en pclk_gasket clock disable bit When HIGH, disable clock
10	RW	0x0	aclk_hdcp22_en aclk_hdcp22 clock disable bit When HIGH, disable clock
9	RW	0x0	hclk_hdcp22_en hclk_hdcp22 clock disable bit When HIGH, disable clock

Bit	Attr	Reset Value	Description
8	RW	0x0	pclk_hdcp22_en pclk_hdcp22 clock disable bit When HIGH, disable clock
7	RW	0x0	pclk_dp_ctrl_en pclk_dp_ctrl clock disable bit When HIGH, disable clock
6	RW	0x0	pclk_hdmi_ctrl_en pclk_hdmi_ctrl clock disable bit When HIGH, disable clock
5	RW	0x0	hclk_hdcpnoc_en hclk_hdcpnoc clock disable bit When HIGH, disable clock Suggest always on
4	RW	0x0	aclk_hdcpnoc_en aclk_hdcpnoc clock disable bit When HIGH, disable clock Suggest always on
3	RW	0x0	pclk_hdcpnoc_en pclk_hdcpnoc clock disable bit When HIGH, disable clock Suggest always on
2	RW	0x0	pclk_mipi_dsi1_en pclk_mipi_dsi1 clock disable bit When HIGH, disable clock
1	RW	0x0	pclk_mipi_dsi0_en pclk_mipi_dsi0 clock disable bit When HIGH, disable clock
0	RW	0x0	aclk_vio_noc_en aclk_vio_noc clock disable bit When HIGH, disable clock Suggest always on

CRU_CLKGATE_CON30
Address: Operational Base + offset (0x0378)
Internal clock gating register30

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:12	RO	0x0	reserved
11	RW	0x0	aclk_gpu_grf_en aclk_gpu_grf clock disable bit When HIGH, disable clock Suggest always on

Bit	Attr	Reset Value	Description
10	RW	0x0	aclk_perf_gpu_en aclk_perf_gpu clock disable bit When HIGH, disable clock
9	RO	0x0	reserved
8	RW	0x0	aclk_gpu_en aclk_gpu clock disable bit When HIGH, disable clock
7:5	RO	0x0	reserved
4	RW	0x0	aclk_usb3_grf_en aclk_usb3_grf clock disable bit When HIGH, disable clock Suggest always on
3	RW	0x0	aclk_usb3_rksoc_axi_perf_en aclk_usb3_rksoc_axi_perf clock disable bit When HIGH, disable clock
2	RW	0x0	aclk_usb3otg1_en aclk_usb3otg1 clock disable bit When HIGH, disable clock
1	RW	0x0	aclk_usb3otg0_en aclk_usb3otg0 clock disable bit When HIGH, disable clock
0	RW	0x0	aclk_usb3_noc_en aclk_usb3_noc clock disable bit When HIGH, disable clock Suggest always on

CRU_CLKGATE_CON31

Address: Operational Base + offset (0x037c)

Internal clock gating register31

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:11	RO	0x0	reserved
10	RW	0x0	pclk_sgrf_en pclk_sgrf clock disable bit When HIGH, disable clock Suggest always on
9	RW	0x0	pclk_pmu_intr_arb_en pclk_pmu_intr_arb clock disable bit When HIGH, disable clock
8	RW	0x0	pclk_hsicphy_en pclk_hsicphy clock disable bit When HIGH, disable clock

Bit	Attr	Reset Value	Description
7	RW	0x0	pclk_timer1_en pclk_timer1 clock disable bit When HIGH, disable clock
6	RW	0x0	pclk_timer0_en pclk_timer0 clock disable bit When HIGH, disable clock
5	RW	0x0	pclk_gpio4_en pclk_gpio4 clock disable bit When HIGH, disable clock
4	RW	0x0	pclk_gpio3_en pclk_gpio3 clock disable bit When HIGH, disable clock
3	RW	0x0	pclk_gpio2_en pclk_gpio2 clock disable bit When HIGH, disable clock
2	RW	0x0	pclk_intr_arb_en pclk_intr_arb clock disable bit When HIGH, disable clock
1	RW	0x0	pclk_grf_en pclk_grf clock disable bit When HIGH, disable clock Suggest always on
0	RO	0x0	reserved

CRU_CLKGATE_CON32

Address: Operational Base + offset (0x0380)

Internal clock gating register32

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:14	RO	0x0	reserved
13	RW	0x0	pclk_edp_ctrl_en pclk_edp_ctrl clock disable bit When HIGH, disable clock
12	RW	0x0	pclk_edp_noc_en pclk_edp_noc clock disable bit When HIGH, disable clock Suggest always on
11	RO	0x0	reserved
10	RW	0x0	aclk_emmc_grf_en aclk_emmc_grf clock disable bit When HIGH, disable clock Suggest always on

Bit	Attr	Reset Value	Description
9	RW	0x0	aclk_emmc_noc_en aclk_emmc_noc clock disable bit When HIGH, disable clock Suggest always on
8	RW	0x0	aclk_emmc_core_en aclk_emmc_core clock disable bit When HIGH, disable clock
7:5	RO	0x0	reserved
4	RW	0x0	aclk_perf_gmac_en aclk_perf_gmac clock disable bit When HIGH, disable clock
3	RW	0x0	pclk_gmac_noc_en pclk_gmac_noc clock disable bit When HIGH, disable clock Suggest always on
2	RW	0x0	pclk_gmac_en pclk_gmac clock disable bit When HIGH, disable clock
1	RW	0x0	aclk_gmac_noc_en aclk_gmac_noc clock disable bit When HIGH, disable clock Suggest always on
0	RW	0x0	aclk_gmac_en aclk_gmac clock disable bit When HIGH, disable clock

CRU_CLKGATE_CON33

Address: Operational Base + offset (0x0384)

Internal clock gating register33

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:10	RO	0x0	reserved
9	RW	0x0	hclk_sd_noc_en hclk_sd_noc clock disable bit When HIGH, disable clock Suggest always on
8	RW	0x0	hclk_sdmmc_en hclk_sdmmc clock disable bit When HIGH, disable clock
7:6	RO	0x0	reserved

Bit	Attr	Reset Value	Description
5	RW	0x0	aclk_gic_adb400_gic_2_core_b_en aclk_gic_adb400_gic_2_core_b clock disable bit When HIGH, disable clock
4	RW	0x0	aclk_gic_adb400_gic_2_core_l_en aclk_gic_adb400_gic_2_core_l clock disable bit When HIGH, disable clock
3	RW	0x0	aclk_gic_adb400_core_b_2_gic_en aclk_gic_adb400_core_b_2_gic clock disable bit When HIGH, disable clock
2	RW	0x0	aclk_gic_adb400_core_l_2_gic_en aclk_gic_adb400_core_l_2_gic clock disable bit When HIGH, disable clock
1	RW	0x0	aclk_gic_noc_en aclk_gic_noc clock disable bit When HIGH, disable clock Suggest always on
0	RW	0x0	aclk_gic_en aclk_gic clock disable bit When HIGH, disable clock

CRU_CLKGATE_CON34
Address: Operational Base + offset (0x0388)
Internal clock gating register34

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:7	RO	0x0	reserved
6	RW	0x0	hclk_sdioaudio_noc_en hclk_sdioaudio_noc clock disable bit When HIGH, disable clock Suggest always on
5	RW	0x0	pclk_spi5_en pclk_spi5 clock disable bit When HIGH, disable clock
4	RW	0x0	hclk_sdio_en hclk_sdio clock disable bit When HIGH, disable clock
3	RW	0x0	hclk_spdif_en hclk_spdif clock disable bit When HIGH, disable clock
2	RW	0x0	hclk_i2s2_en hclk_i2s2 clock disable bit When HIGH, disable clock

Bit	Attr	Reset Value	Description
1	RW	0x0	hclk_i2s1_en hclk_i2s1 clock disable bit When HIGH, disable clock
0	RW	0x0	hclk_i2s0_en hclk_i2s0 clock disable bit When HIGH, disable clock

CRU_SOFTRST_CON0
Address: Operational Base + offset (0x0400)
Internal software reset control register0

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:12	RO	0x0	reserved
11	R/WSC	0x0	resetn_dbg_noc_req resetn_dbg_noc request bit When HIGH, reset relative logic
10	R/WSC	0x0	aresetn_ccim1_noc_req aresetn_ccim1_noc request bit When HIGH, reset relative logic
9	R/WSC	0x0	aresetn_ccim0_noc_req aresetn_ccim0_noc request bit When HIGH, reset relative logic
8	R/WSC	0x0	aresetn_cci_req aresetn_cci request bit When HIGH, reset relative logic
7	R/WSC	0x0	adb_b_srstn_req adb_b_srstn request bit When HIGH, reset relative logic
6	R/WSC	0x0	adb_l_srstn_req adb_l_srstn request bit When HIGH, reset relative logic
5	R/WSC	0x0	l2_b_srstn_req l2_b_srstn request bit When HIGH, reset relative logic
4	R/WSC	0x0	l2_l_srstn_req l2_l_srstn request bit When HIGH, reset relative logic
3	R/WSC	0x0	corepo0_b_srstn_req corepo0_b_srstn request bit When HIGH, reset relative logic

Bit	Attr	Reset Value	Description
2	R/W SC	0x0	corepo0_l_srstn_req corepo0_l_srstn request bit When HIGH, reset relative logic
1	R/W SC	0x0	core0_b_srstn_req core0_b_srstn request bit When HIGH, reset relative logic
0	R/W SC	0x0	core0_l_srstn_req core0_l_srstn request bit When HIGH, reset relative logic

CRU_SOFTRST_CON1

Address: Operational Base + offset (0x0404)

Internal software reset control register1

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	pvtm_core_l_srstn_req pvtm_core_l_srstn request bit When HIGH, reset relative logic
14	RW	0x0	rkperf_l_arstn_req rkperf_l_arstn request bit When HIGH, reset relative logic
13	RW	0x0	adb_l_srstn_req_t adb_l_srstn request bit When HIGH, reset relative logic
12	RW	0x0	l2_l_srstn_req_t l2_l_srstn request bit When HIGH, reset relative logic
11	RO	0x0	reserved
10	RW	0x0	prstn_dbg_l_req prstn_dbg_l request bit When HIGH, reset relative logic
9	RW	0x0	arstn_adb400_corel2gic_req arstn_adb400_corel2gic request bit When HIGH, reset relative logic
8	RW	0x0	arstn_adb400_gic2corel_req arstn_adb400_gic2corel request bit When HIGH, reset relative logic
7	RW	0x0	corepo3_l_srstn_req corepo3_l_srstn request bit When HIGH, reset relative logic

Bit	Attr	Reset Value	Description
6	RW	0x0	corepo2_l_srstn_req corepo2_l_srstn request bit When HIGH, reset relative logic
5	RW	0x0	corepo1_l_srstn_req corepo1_l_srstn request bit When HIGH, reset relative logic
4	RW	0x0	corepo0_l_srstn_req_t corepo0_l_srstn request bit When HIGH, reset relative logic
3	RW	0x0	core3_l_srstn_req core3_l_srstn request bit When HIGH, reset relative logic
2	RW	0x0	core2_l_srstn_req core2_l_srstn request bit When HIGH, reset relative logic
1	RW	0x0	core1_l_srstn_req core1_l_srstn request bit When HIGH, reset relative logic
0	RW	0x0	core0_l_srstn_req_t core0_l_srstn request bit When HIGH, reset relative logic

CRU_SOFTRST_CON2
Address: Operational Base + offset (0x0408)
Internal software reset control register2

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	pvtm_core_b_srstn_req pvtm_core_b_srstn request bit When HIGH, reset relative logic
14	RW	0x0	rkperf_b_arstn_req rkperf_b_arstn request bit When HIGH, reset relative logic
13	RW	0x0	adb_b_srstn_req_t adb_b_srstn request bit When HIGH, reset relative logic
12	RW	0x0	l2_b_srstn_req_t l2_b_srstn request bit When HIGH, reset relative logic
11	RO	0x0	reserved

Bit	Attr	Reset Value	Description
10	RW	0x0	prstn_dbg_b_req prstn_dbg_b request bit When HIGH, reset relative logic
9	RW	0x0	arstn_adb400_coreb2gic_req arstn_adb400_coreb2gic request bit When HIGH, reset relative logic
8	RW	0x0	arstn_adb400_gic2coreb_req arstn_adb400_gic2coreb request bit When HIGH, reset relative logic
7:6	RO	0x0	reserved
5	RW	0x0	corepo1_b_srstn_req corepo1_b_srstn request bit When HIGH, reset relative logic
4	RW	0x0	corepo0_b_srstn_req_t corepo0_b_srstn request bit When HIGH, reset relative logic
3:2	RO	0x0	reserved
1	RW	0x0	core1_b_srstn_req core1_b_srstn request bit When HIGH, reset relative logic
0	RW	0x0	core0_b_srstn_req_t core0_b_srstn request bit When HIGH, reset relative logic

CRU_SOFTRST_CON3

Address: Operational Base + offset (0x040c)

Internal software reset control register3

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:11	RO	0x0	reserved
10	RW	0x0	presetn_cci_grf_req presetn_cci_grf request bit When HIGH, reset relative logic
9	RW	0x0	resetn_cci_trace_req resetn_cci_trace request bit When HIGH, reset relative logic
8	RW	0x0	resetn_dbg_cxcs_req resetn_dbg_cxcs request bit When HIGH, reset relative logic
7	RW	0x0	resetn_dbg_noc_req_t resetn_dbg_noc request bit When HIGH, reset relative logic

Bit	Attr	Reset Value	Description
6	RW	0x0	aresetn_adb400m_pd_core_l_req_t aresetn_adb400m_pd_core_l request bit When HIGH, reset relative logic
5	RW	0x0	aresetn_adb400m_pd_core_b_req_t aresetn_adb400m_pd_core_b request bit When HIGH, reset relative logic
4	RO	0x1	aresetn_ccim1_noc_req_t aresetn_ccim1_noc request bit When HIGH, reset relative logic
3	RW	0x0	aresetn_ccim0_noc_req_t aresetn_ccim0_noc request bit When HIGH, reset relative logic
2	RW	0x0	aresetn_cci_req_t aresetn_cci request bit When HIGH, reset relative logic
1:0	RO	0x0	reserved

CRU_SOFTRST_CON4
Address: Operational Base + offset (0x0410)
Internal software reset control register4

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	resetn_pvtm_ddr_req resetn_pvtm_ddr request bit When HIGH, reset relative logic
14	RW	0x0	resetn_ddr_cic_req resetn_ddr_cic request bit When HIGH, reset relative logic
13	RW	0x0	resetn_ddrphy1_req resetn_ddrphy1 request bit When HIGH, reset relative logic
12	RW	0x0	resetn_ddr1_req resetn_ddr1 request bit When HIGH, reset relative logic
11	RW	0x0	resetn_ddrcfg1_msch_req resetn_ddrcfg1_msch request bit When HIGH, reset relative logic
10	RW	0x0	resetn_ddr1_msch_req resetn_ddr1_msch request bit When HIGH, reset relative logic

Bit	Attr	Reset Value	Description
9	RW	0x0	resetn_ddrphy0_req resetn_ddrphy0 request bit When HIGH, reset relative logic
8	RW	0x0	resetn_ddr0_req resetn_ddr0 request bit When HIGH, reset relative logic
7	RW	0x0	resetn_ddrcfg0_msch_req resetn_ddrcfg0_msch request bit When HIGH, reset relative logic
6	RW	0x0	resetn_ddr0_msch_req resetn_ddr0_msch request bit When HIGH, reset relative logic
5	RW	0x0	presetn_center_sgrf_req presetn_center_sgrf request bit When HIGH, reset relative logic
4	RW	0x0	presetn_cic_req presetn_cic request bit When HIGH, reset relative logic
3	RW	0x0	presetn_ddrmon_req presetn_ddrmon request bit When HIGH, reset relative logic
2	RW	0x0	presetn_center_main_req presetn_center_main request bit When HIGH, reset relative logic
1	RW	0x0	aresetn_center_peri_noc_req aresetn_center_peri_noc request bit When HIGH, reset relative logic
0	RW	0x0	aresetn_center_main_noc_req aresetn_center_main_noc request bit When HIGH, reset relative logic

CRU_SOFTRST_CON5

Address: Operational Base + offset (0x0414)

Internal software reset control register5

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:14	RO	0x0	reserved
13	RW	0x0	resetn_vdu_ca_req resetn_vdu_ca request bit When HIGH, reset relative logic

Bit	Attr	Reset Value	Description
12	RW	0x0	resetn_vdu_core_req resetn_vdu_core request bit When HIGH, reset relative logic
11	RW	0x0	hresetn_vdu_req hresetn_vdu request bit When HIGH, reset relative logic
10	RW	0x0	hresetn_vdu_noc_req hresetn_vdu_noc request bit When HIGH, reset relative logic
9	RW	0x0	aresetn_vdu_req aresetn_vdu request bit When HIGH, reset relative logic
8	RW	0x0	aresetn_vdu_noc_req aresetn_vdu_noc request bit When HIGH, reset relative logic
7:4	RO	0x0	reserved
3	RW	0x0	hresetn_vcodec_req hresetn_vcodec request bit When HIGH, reset relative logic
2	RW	0x0	hresetn_vcodec_noc_req hresetn_vcodec_noc request bit When HIGH, reset relative logic
1	RW	0x0	aresetn_vcodec_req aresetn_vcodec request bit When HIGH, reset relative logic
0	RW	0x0	aresetn_vcodec_noc_req aresetn_vcodec_noc request bit When HIGH, reset relative logic

CRU_SOFTRST_CON6

Address: Operational Base + offset (0x0418)

Internal software reset control register6

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RO	0x0	reserved
14	RW	0x0	aresetn_emmc_grf_req aresetn_emmc_grf request bit When HIGH, reset relative logic
13	RW	0x0	aresetn_emmc_req aresetn_emmc request bit When HIGH, reset relative logic

Bit	Attr	Reset Value	Description
12	RW	0x0	aresetn_emmc_noc_req aresetn_emmc_noc request bit When HIGH, reset relative logic
11	RO	0x0	reserved
10	RW	0x0	resetn_rga_core_req resetn_rga_core request bit When HIGH, reset relative logic
9	RW	0x0	hresetn_rga_req hresetn_rga request bit When HIGH, reset relative logic
8	RW	0x0	hresetn_rga_noc_req hresetn_rga_noc request bit When HIGH, reset relative logic
7	RW	0x0	aresetn_rga_req aresetn_rga request bit When HIGH, reset relative logic
6	RW	0x0	aresetn_rga_noc_req aresetn_rga_noc request bit When HIGH, reset relative logic
5	RO	0x0	reserved
4	RW	0x0	hresetn_iep_req hresetn_iep request bit When HIGH, reset relative logic
3	RW	0x0	hresetn_iep_noc_req hresetn_iep_noc request bit When HIGH, reset relative logic
2	RW	0x0	aresetn_iep_req aresetn_iep request bit When HIGH, reset relative logic
1	RW	0x0	aresetn_vop_iep_req aresetn_vop_iep request bit When HIGH, reset relative logic
0	RW	0x0	aresetn_iep_noc_req aresetn_iep_noc request bit When HIGH, reset relative logic

CRU_SOFTRST_CON7

Address: Operational Base + offset (0x041c)

Internal software reset control register7

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit

Bit	Attr	Reset Value	Description
15	RW	0x0	presetn_hsicphy_req presetn_hsicphy request bit When HIGH, reset relative logic
14	RW	0x0	presetn_perihp_noc_req presetn_perihp_noc request bit When HIGH, reset relative logic
13	RW	0x0	hresetn_ahb1tom_req hresetn_ahb1tom request bit When HIGH, reset relative logic
12	RW	0x0	hresetn_hsic_aux_req hresetn_hsic_aux request bit When HIGH, reset relative logic
11	RW	0x0	hresetn_hsic_req hresetn_hsic request bit When HIGH, reset relative logic
10	RW	0x0	hresetn_sdmmc_req hresetn_sdmmc request bit When HIGH, reset relative logic
9	RW	0x0	hresetn_sdio0_req hresetn_sdio0 request bit When HIGH, reset relative logic
8	RW	0x0	hresetn_host1_arb_req hresetn_host1_arb request bit When HIGH, reset relative logic
7	RW	0x0	hresetn_hostc1_aux_req hresetn_hostc1_aux request bit When HIGH, reset relative logic
6	RW	0x0	hresetn_usbhost1_req hresetn_usbhost1 request bit When HIGH, reset relative logic
5	RW	0x0	hresetn_host0_arb_req hresetn_host0_arb request bit When HIGH, reset relative logic
4	RW	0x0	hresetn_hostc0_aux_req hresetn_hostc0_aux request bit When HIGH, reset relative logic
3	RW	0x0	hresetn_usbhost0_req hresetn_usbhost0 request bit When HIGH, reset relative logic
2	RW	0x0	hresetn_perihp_noc_req hresetn_perihp_noc request bit When HIGH, reset relative logic
1	RW	0x0	presetn_perihp_grf_req presetn_perihp_grf request bit When HIGH, reset relative logic

Bit	Attr	Reset Value	Description
0	RW	0x0	aresetn_perihp_noc_req aresetn_perihp_noc request bit When HIGH, reset relative logic

CRU_SOFTRST_CON8
Address: Operational Base + offset (0x0420)
Internal software reset control register8

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	hsicphy_utmi_rstn_req hsicphy_utmi_rstn request bit When HIGH, reset relative logic
14	RW	0x0	hsicphy_por_rstn_req hsicphy_por_rstn request bit When HIGH, reset relative logic
13	RO	0x0	reserved
12	RW	0x0	presetn_gmac_grf_req presetn_gmac_grf request bit When HIGH, reset relative logic
11	RO	0x0	reserved
10	RW	0x0	presetn_gmac_noc_req presetn_gmac_noc request bit When HIGH, reset relative logic
9	RW	0x0	aresetn_gmac_req aresetn_gmac request bit When HIGH, reset relative logic
8	RW	0x0	aresetn_gmac_noc_req aresetn_gmac_noc request bit When HIGH, reset relative logic
7	RW	0x1	resetn_pciephy_req resetn_pciephy request bit When HIGH, reset relative logic
6	RW	0x0	resetn_pcie_pm_req resetn_pcie_pm request bit When HIGH, reset relative logic
5	RW	0x1	resetn_pcie_pipe_req resetn_pcie_pipe request bit When HIGH, reset relative logic
4	RW	0x1	resetn_pcie_mgmt_sticky_req resetn_pcie_mgmt_sticky request bit When HIGH, reset relative logic

Bit	Attr	Reset Value	Description
3	RW	0x1	resetn_pcie_mgmt_req resetn_pcie_mgmt request bit When HIGH, reset relative logic
2	RW	0x1	resetn_pcie_core_req resetn_pcie_core request bit When HIGH, reset relative logic
1	RW	0x0	presetn_pcie_req presetn_pcie request bit When HIGH, reset relative logic
0	RW	0x0	aresetn_pcie_req aresetn_pcie request bit When HIGH, reset relative logic

CRU_SOFTRST_CON9

Address: Operational Base + offset (0x0424)

Internal software reset control register9

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RO	0x0	reserved
14	RW	0x0	resetn_uphy1_tcpdpwrup_req resetn_uphy1_tcpdpwrup request bit When HIGH, reset relative logic
13	RW	0x0	resetn_uphy1_req resetn_uphy1 request bit When HIGH, reset relative logic
12	RW	0x0	resetn_uphy1_pipe_l00_req resetn_uphy1_pipe_l00 request bit When HIGH, reset relative logic
11	RW	0x0	resetn_usb2phy1_ehciphy_req resetn_usb2phy1_ehciphy request bit When HIGH, reset relative logic
10	RW	0x0	resetn_usb2phy1_utmi_port1_req resetn_usb2phy1_utmi_port1 request bit When HIGH, reset relative logic
9	RW	0x0	resetn_usb2phy1_utmi_port0_req resetn_usb2phy1_utmi_port0 request bit When HIGH, reset relative logic
8	RW	0x0	resetn_usb2phy1_por_req resetn_usb2phy1_por request bit When HIGH, reset relative logic
7	RO	0x0	reserved

Bit	Attr	Reset Value	Description
6	RW	0x0	resetn_uphy0_tcpdpwrup_req resetn_uphy0_tcpdpwrup request bit When HIGH, reset relative logic
5	RW	0x0	resetn_uphy0_req resetn_uphy0 request bit When HIGH, reset relative logic
4	RW	0x0	resetn_uphy0_pipe_l00_req resetn_uphy0_pipe_l00 request bit When HIGH, reset relative logic
3	RW	0x0	resetn_usb2phy0_ehciphy_req resetn_usb2phy0_ehciphy request bit When HIGH, reset relative logic
2	RW	0x0	resetn_usb2phy0_utmi_port1_req resetn_usb2phy0_utmi_port1 request bit When HIGH, reset relative logic
1	RW	0x0	resetn_usb2phy0_utmi_port0_req resetn_usb2phy0_utmi_port0 request bit When HIGH, reset relative logic
0	RW	0x0	resetn_usb2phy0_por_req resetn_usb2phy0_por request bit When HIGH, reset relative logic

CRU_SOFTRST_CON10

Address: Operational Base + offset (0x0428)

Internal software reset control register10

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	hresetn_crypto0_m_req hresetn_crypto0_m request bit When HIGH, reset relative logic
14	RW	0x0	hresetn_crypto0_s_req hresetn_crypto0_s request bit When HIGH, reset relative logic
13	RW	0x0	hresetn_rom_req hresetn_rom request bit When HIGH, reset relative logic
12	RW	0x0	hresetn_perilp0_noc_req hresetn_perilp0_noc request bit When HIGH, reset relative logic
11	RW	0x0	hresetn_perilp0_req hresetn_perilp0 request bit When HIGH, reset relative logic

Bit	Attr	Reset Value	Description
10	RW	0x0	aresetn_adb400_slv1_req aresetn_adb400_slv1 request bit When HIGH, reset relative logic
9	RW	0x0	aresetn_adb400_slv0_req aresetn_adb400_slv0 request bit When HIGH, reset relative logic
8	RW	0x0	aresetn_adb400_mst1_req aresetn_adb400_mst1 request bit When HIGH, reset relative logic
7	RW	0x0	aresetn_adb400_mst0_req aresetn_adb400_mst0 request bit When HIGH, reset relative logic
6	RW	0x0	aresetn_intmem_req aresetn_intmem request bit When HIGH, reset relative logic
5	RW	0x0	aresetn_tzma_req aresetn_tzma request bit When HIGH, reset relative logic
4	RW	0x0	aresetn_dmac1_perilp0_req aresetn_dmac1_perilp0 request bit When HIGH, reset relative logic
3	RW	0x0	aresetn_dmac0_perilp0_req aresetn_dmac0_perilp0 request bit When HIGH, reset relative logic
2	RW	0x0	aresetn_gic500_req aresetn_gic500 request bit When HIGH, reset relative logic
1	RW	0x0	aresetn_dcf_req aresetn_dcf request bit When HIGH, reset relative logic
0	RW	0x0	aresetn_perilp0_noc_req aresetn_perilp0_noc request bit When HIGH, reset relative logic

CRU_SOFTRST_CON11

Address: Operational Base + offset (0x042c)

Internal software reset control register11

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RO	0x0	reserved

Bit	Attr	Reset Value	Description
14	RW	0x0	hresetn_sdioaudio_brg_req hresetn_sdioaudio_brg request bit When HIGH, reset relative logic
13	RW	0x0	hresetn_sd_noc_req hresetn_sd_noc request bit When HIGH, reset relative logic
12	RW	0x0	aresetn_gic_noc_req aresetn_gic_noc request bit When HIGH, reset relative logic
11	RO	0x0	reserved
10	RW	0x0	resetn_crypto1_req resetn_crypto1 request bit When HIGH, reset relative logic
9	RW	0x0	hresetn_crypto1_m_req hresetn_crypto1_m request bit When HIGH, reset relative logic
8	RW	0x0	hresetn_crypto1_s_req hresetn_crypto1_s request bit When HIGH, reset relative logic
7	RW	0x0	presetn_perilp1_grf_req presetn_perilp1_grf request bit When HIGH, reset relative logic
6	RW	0x0	presetn_perilp1_sgrf_req presetn_perilp1_sgrf request bit When HIGH, reset relative logic
5	RW	0x0	resetn_crypto0_req resetn_crypto0 request bit When HIGH, reset relative logic
4	RW	0x1	poresetn_cm0s_req poresetn_cm0s request bit When HIGH, reset relative logic
3	RW	0x0	dbgresetn_cm0s_req dbgresetn_cm0s request bit When HIGH, reset relative logic
2	RW	0x1	hresetn_cm0s_req hresetn_cm0s request bit When HIGH, reset relative logic
1	RW	0x0	hresetn_cm0s_noc_req hresetn_cm0s_noc request bit When HIGH, reset relative logic
0	RW	0x0	presetn_dcf_req presetn_dcf request bit When HIGH, reset relative logic

CRU_SOFTRST_CON12

Address: Operational Base + offset (0x0430)
Internal software reset control register12

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	presetn_mailbox0_req presetn_mailbox0 request bit When HIGH, reset relative logic
14	RW	0x0	presetn_i2c7_req presetn_i2c7 request bit When HIGH, reset relative logic
13	RW	0x0	presetn_i2c3_req presetn_i2c3 request bit When HIGH, reset relative logic
12	RW	0x0	presetn_i2c6_req presetn_i2c6 request bit When HIGH, reset relative logic
11	RW	0x0	presetn_i2c2_req presetn_i2c2 request bit When HIGH, reset relative logic
10	RW	0x0	presetn_i2c5_req presetn_i2c5 request bit When HIGH, reset relative logic
9	RW	0x0	presetn_i2c1_req presetn_i2c1 request bit When HIGH, reset relative logic
8	RW	0x0	presetn_efuse_1024s_req presetn_efuse_1024s request bit When HIGH, reset relative logic
7	RW	0x0	presetn_efuse_1024_req presetn_efuse_1024 request bit When HIGH, reset relative logic
6	RW	0x0	presetn_perilp1_noc_req presetn_perilp1_noc request bit When HIGH, reset relative logic
5	RW	0x0	hresetn_spdif_8ch_req hresetn_spdif_8ch request bit When HIGH, reset relative logic
4	RW	0x0	hresetn_i2s2_req hresetn_i2s2 request bit When HIGH, reset relative logic
3	RW	0x0	hresetn_i2s1_req hresetn_i2s1 request bit When HIGH, reset relative logic

Bit	Attr	Reset Value	Description
2	RW	0x0	hresetn_i2s0_req hresetn_i2s0 request bit When HIGH, reset relative logic
1	RW	0x0	hresetn_perilp1_noc_req hresetn_perilp1_noc request bit When HIGH, reset relative logic
0	RW	0x0	hresetn_perilp1_req hresetn_perilp1 request bit When HIGH, reset relative logic

CRU_SOFTRST_CON13
Address: Operational Base + offset (0x0434)
Internal software reset control register13

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	resetn_spi5_req resetn_spi5 request bit When HIGH, reset relative logic
14	RW	0x0	resetn_spi4_req resetn_spi4 request bit When HIGH, reset relative logic
13	RW	0x0	resetn_spi2_req resetn_spi2 request bit When HIGH, reset relative logic
12	RW	0x0	resetn_spi1_req resetn_spi1 request bit When HIGH, reset relative logic
11	RW	0x0	resetn_spi0_req resetn_spi0 request bit When HIGH, reset relative logic
10	RW	0x0	presetn_spi5_req presetn_spi5 request bit When HIGH, reset relative logic
9	RW	0x0	presetn_spi4_req presetn_spi4 request bit When HIGH, reset relative logic
8	RW	0x0	presetn_spi2_req presetn_spi2 request bit When HIGH, reset relative logic
7	RW	0x0	presetn_spi1_req presetn_spi1 request bit When HIGH, reset relative logic

Bit	Attr	Reset Value	Description
6	RW	0x0	presetn_spi0_req presetn_spi0 request bit When HIGH, reset relative logic
5	RW	0x0	presetn_tsadc_req presetn_tsadc request bit When HIGH, reset relative logic
4	RW	0x0	presetn_saradc_req presetn_saradc request bit When HIGH, reset relative logic
3	RW	0x0	presetn_uart3_req presetn_uart3 request bit When HIGH, reset relative logic
2	RW	0x0	presetn_uart2_req presetn_uart2 request bit When HIGH, reset relative logic
1	RW	0x0	presetn_uart1_req presetn_uart1 request bit When HIGH, reset relative logic
0	RW	0x0	presetn_uart0_req presetn_uart0 request bit When HIGH, reset relative logic

CRU_SOFTRST_CON14
Address: Operational Base + offset (0x0438)
Internal software reset control register14

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	hresetn_sdioaudio_noc_req hresetn_sdioaudio_noc request bit When HIGH, reset relative logic
14	RW	0x0	resetn_i2c7_req resetn_i2c7 request bit When HIGH, reset relative logic
13	RW	0x0	resetn_i2c3_req resetn_i2c3 request bit When HIGH, reset relative logic
12	RW	0x0	resetn_i2c6_req resetn_i2c6 request bit When HIGH, reset relative logic
11	RW	0x0	resetn_i2c2_req resetn_i2c2 request bit When HIGH, reset relative logic

Bit	Attr	Reset Value	Description
10	RW	0x0	resetn_i2c5_req resetn_i2c5 request bit When HIGH, reset relative logic
9	RW	0x0	resetn_i2c1_req resetn_i2c1 request bit When HIGH, reset relative logic
8	RW	0x0	resetn_tsadc_req resetn_tsadc request bit When HIGH, reset relative logic
7	RW	0x0	resetn_uart3_req resetn_uart3 request bit When HIGH, reset relative logic
6	RW	0x0	resetn_uart2_req resetn_uart2 request bit When HIGH, reset relative logic
5	RW	0x0	resetn_uart1_req resetn_uart1 request bit When HIGH, reset relative logic
4	RW	0x0	resetn_uart0_req resetn_uart0 request bit When HIGH, reset relative logic
3	RW	0x0	resetn_spdif_8ch_req resetn_spdif_8ch request bit When HIGH, reset relative logic
2	RW	0x0	resetn_i2s2_req resetn_i2s2 request bit When HIGH, reset relative logic
1	RW	0x0	resetn_i2s1_req resetn_i2s1 request bit When HIGH, reset relative logic
0	RW	0x0	resetn_i2s0_req resetn_i2s0 request bit When HIGH, reset relative logic

CRU_SOFTRST_CON15

Address: Operational Base + offset (0x043c)

Internal software reset control register15

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RO	0x0	reserved

Bit	Attr	Reset Value	Description
14	RW	0x0	resetn_dp_i2s_req resetn_dp_i2s request bit When HIGH, reset relative logic
13	RW	0x0	resetn_dp_core_req resetn_dp_core request bit When HIGH, reset relative logic
12	RW	0x0	presetn_mipi_dsi1_req presetn_mipi_dsi1 request bit When HIGH, reset relative logic
11	RW	0x0	presetn_mipi_dsi0_req presetn_mipi_dsi0 request bit When HIGH, reset relative logic
10	RW	0x0	cresetn_dp_ctrl_req cresetn_dp_ctrl request bit When HIGH, reset relative logic
9	RW	0x0	sresetn_dp_ctrl_req sresetn_dp_ctrl request bit When HIGH, reset relative logic
8	RW	0x0	presetn_dp_ctrl_req presetn_dp_ctrl request bit When HIGH, reset relative logic
7	RW	0x0	presetn_hdmi_ctrl_req presetn_hdmi_ctrl request bit When HIGH, reset relative logic
6	RW	0x0	presetn_hdcp_req presetn_hdcp request bit When HIGH, reset relative logic
5	RW	0x0	presetn_hdcp_noc_req presetn_hdcp_noc request bit When HIGH, reset relative logic
4	RW	0x0	hresetn_hdcp_req hresetn_hdcp request bit When HIGH, reset relative logic
3	RW	0x0	hresetn_hdcp_noc_req hresetn_hdcp_noc request bit When HIGH, reset relative logic
2	RW	0x0	aresetn_hdcp_req aresetn_hdcp request bit When HIGH, reset relative logic
1	RW	0x0	aresetn_hdcp_noc_req aresetn_hdcp_noc request bit When HIGH, reset relative logic
0	RW	0x0	aresetn_vio_noc_req aresetn_vio_noc request bit When HIGH, reset relative logic

CRU_SOFTRST_CON16
Address: Operational Base + offset (0x0440)
Internal software reset control register16

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	resetn_isp1_req resetn_isp1 request bit When HIGH, reset relative logic
14	RW	0x0	resetn_isp0_req resetn_isp0 request bit When HIGH, reset relative logic
13	RW	0x0	hresetn_isp1_req hresetn_isp1 request bit When HIGH, reset relative logic
12	RW	0x0	hresetn_isp0_req hresetn_isp0 request bit When HIGH, reset relative logic
11	RW	0x0	hresetn_isp1_noc_req hresetn_isp1_noc request bit When HIGH, reset relative logic
10	RW	0x0	hresetn_isp0_noc_req hresetn_isp0_noc request bit When HIGH, reset relative logic
9:8	RO	0x0	reserved
7	RW	0x0	aresetn_isp1_noc_req aresetn_isp1_noc request bit When HIGH, reset relative logic
6	RW	0x0	aresetn_isp0_noc_req aresetn_isp0_noc request bit When HIGH, reset relative logic
5	RW	0x0	resetn_hdcp_ctrl_req resetn_hdcp_ctrl request bit When HIGH, reset relative logic
4	RW	0x0	resetn_hdmi_ctrl_req resetn_hdmi_ctrl request bit When HIGH, reset relative logic
3	RW	0x0	resetn_dptx_spdif_rec_req resetn_dptx_spdif_rec request bit When HIGH, reset relative logic
2	RW	0x0	presetn_vio_grf_req presetn_vio_grf request bit When HIGH, reset relative logic

Bit	Attr	Reset Value	Description
1	RO	0x0	reserved
0	RW	0x0	presetn_gasket_req presetn_gasket request bit When HIGH, reset relative logic

CRU_SOFTRST_CON17
Address: Operational Base + offset (0x0444)
Internal software reset control register17

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:14	RO	0x0	reserved
13	RW	0x0	presetn_edp_ctrl_req presetn_edp_ctrl request bit When HIGH, reset relative logic
12	RW	0x0	presetn_edp_noc_req presetn_edp_noc request bit When HIGH, reset relative logic
11	RW	0x0	resetn_vop1_pwm_req resetn_vop1_pwm request bit When HIGH, reset relative logic
10	RW	0x0	resetn_vop0_pwm_req resetn_vop0_pwm request bit When HIGH, reset relative logic
9	RW	0x0	dresetn_vop1_req dresetn_vop1 request bit When HIGH, reset relative logic
8	RW	0x0	dresetn_vop0_req dresetn_vop0 request bit When HIGH, reset relative logic
7	RW	0x0	hresetn_vop1_req hresetn_vop1 request bit When HIGH, reset relative logic
6	RW	0x0	hresetn_vop0_req hresetn_vop0 request bit When HIGH, reset relative logic
5	RW	0x0	hresetn_vop1_noc_req hresetn_vop1_noc request bit When HIGH, reset relative logic
4	RW	0x0	hresetn_vop0_noc_req hresetn_vop0_noc request bit When HIGH, reset relative logic

Bit	Attr	Reset Value	Description
3	RW	0x0	aresetn_vop1_req aresetn_vop1 request bit When HIGH, reset relative logic
2	RW	0x0	aresetn_vop0_req aresetn_vop0 request bit When HIGH, reset relative logic
1	RW	0x0	aresetn_vop1_noc_req aresetn_vop1_noc request bit When HIGH, reset relative logic
0	RW	0x0	aresetn_vop0_noc_req aresetn_vop0_noc request bit When HIGH, reset relative logic

CRU_SOFTRST_CON18

Address: Operational Base + offset (0x0448)

Internal software reset control register18

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:9	RO	0x0	reserved
8	RW	0x0	pmu_srstn_req pmu_srstn request bit When HIGH, reset relative logic
7	RW	0x0	aresetn_usb3_grf_req aresetn_usb3_grf request bit When HIGH, reset relative logic
6	RW	0x0	aresetn_usb3_otg1_req aresetn_usb3_otg1 request bit When HIGH, reset relative logic
5	RW	0x0	aresetn_usb3_otg0_req aresetn_usb3_otg0 request bit When HIGH, reset relative logic
4	RW	0x0	aresetn_usb3_noc_req aresetn_usb3_noc request bit When HIGH, reset relative logic
3	RW	0x0	resetn_pvtm_gpu_req resetn_pvtm_gpu request bit When HIGH, reset relative logic
2	RW	0x0	aresetn_gpu_grf_req aresetn_gpu_grf request bit When HIGH, reset relative logic

Bit	Attr	Reset Value	Description
1	RW	0x0	aresetn_gpu_noc_req aresetn_gpu_noc request bit When HIGH, reset relative logic
0	RW	0x0	aresetn_gpu_req aresetn_gpu request bit When HIGH, reset relative logic

CRU_SOFTRST_CON19

Address: Operational Base + offset (0x044c)

Internal software reset control register19

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	presetn_alive_sgrf_req presetn_alive_sgrf request bit When HIGH, reset relative logic
14	RW	0x0	presetn_intr_arb_pmu_req presetn_intr_arb_pmu request bit When HIGH, reset relative logic
13	RW	0x0	timer11_srstn_req timer11_srstn request bit When HIGH, reset relative logic
12	RW	0x0	timer10_srstn_req timer10_srstn request bit When HIGH, reset relative logic
11	RW	0x0	timer9_srstn_req timer9_srstn request bit When HIGH, reset relative logic
10	RW	0x0	timer8_srstn_req timer8_srstn request bit When HIGH, reset relative logic
9	RW	0x0	timer7_srstn_req timer7_srstn request bit When HIGH, reset relative logic
8	RW	0x0	timer6_srstn_req timer6_srstn request bit When HIGH, reset relative logic
7	RW	0x0	timer_6_11_psrstn_req timer_6_11_psrstn request bit When HIGH, reset relative logic
6	RW	0x0	timer5_srstn_req timer5_srstn request bit When HIGH, reset relative logic

Bit	Attr	Reset Value	Description
5	RW	0x0	timer4_srstn_req timer4_srstn request bit When HIGH, reset relative logic
4	RW	0x0	timer3_srstn_req timer3_srstn request bit When HIGH, reset relative logic
3	RW	0x0	timer2_srstn_req timer2_srstn request bit When HIGH, reset relative logic
2	RW	0x0	timer1_srstn_req timer1_srstn request bit When HIGH, reset relative logic
1	RW	0x0	timer0_srstn_req timer0_srstn request bit When HIGH, reset relative logic
0	RW	0x0	timer_0_5_psrstn_req timer_0_5_psrstn request bit When HIGH, reset relative logic

CRU_SOFTRST_CON20
Address: Operational Base + offset (0x0450)
Internal software reset control register20

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	presetn_uphy1_tcpdctrl_req presetn_uphy1_tcpdctrl request bit When HIGH, reset relative logic
14	RW	0x0	presetn_uphy0_tcpdctrl_req presetn_uphy0_tcpdctrl request bit When HIGH, reset relative logic
13	RW	0x0	presetn_uphy1_tcphy_req presetn_uphy1_tcphy request bit When HIGH, reset relative logic
12	RW	0x0	presetn_uphy0_tcphy_req presetn_uphy0_tcphy request bit When HIGH, reset relative logic
11	RO	0x0	reserved
10	RW	0x0	presetn_uphy0_apb_req presetn_uphy0_apb request bit When HIGH, reset relative logic
9	RO	0x0	reserved

Bit	Attr	Reset Value	Description
8	RW	0x0	presetn_uphy0_dptx_req presetn_uphy0_dptx request bit When HIGH, reset relative logic
7	RW	0x0	presetn_intr_arb_req presetn_intr_arb request bit When HIGH, reset relative logic
6	RW	0x0	presetn_wdt1_req presetn_wdt1 request bit When HIGH, reset relative logic
5	RW	0x0	presetn_wdt0_req presetn_wdt0 request bit When HIGH, reset relative logic
4	RW	0x0	presetn_alive_noc_req presetn_alive_noc request bit When HIGH, reset relative logic
3	RW	0x0	presetn_grf_req presetn_grf request bit When HIGH, reset relative logic
2	RW	0x0	presetn_gpio4_req presetn_gpio4 request bit When HIGH, reset relative logic
1	RW	0x0	presetn_gpio3_req presetn_gpio3 request bit When HIGH, reset relative logic
0	RW	0x0	presetn_gpio2_req presetn_gpio2 request bit When HIGH, reset relative logic

CRU_GLB_SRST_FST_VALUE
Address: Operational Base + offset (0x0500)
The first global software reset config value

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	glb_srst_fst_value The first global software reset config value If config 0xfdb9, it will generate first global software reset

CRU_GLB_SRST_SND_VALUE
Address: Operational Base + offset (0x0504)
The second global software reset config value

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	glb_srst_snd_value The second global software reset config value If config 0xeca8, it will generate second global software reset

CRU_GLB_CNT_TH
Address: Operational Base + offset (0x0508)
Global soft reset counter threshold

Bit	Attr	Reset Value	Description
31:10	RO	0x0	reserved
9:0	RW	0x000	glb_rst_cnt_th global reset wait counter threshold wait cycles n(at xin_24m)

CRU_MISC_CON
Address: Operational Base + offset (0x050c)
Output clock selection for test

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:9	RO	0x0	reserved
8	RW	0x0	core_dbgrst_wfien A53/A72 dbg reset wait for STANDBYWFI enable 1'b0 : A53 dbg reset has no relation to STANDBYWFI status 1'b1 : A53 dgb reset is asserted after STANDBYWFI valid
7	RW	0x0	core_wrst_wifen A53/A72 warm reset wait for STANDBYWFI enable 1'b0 : A53/A72 warm reset has no relation to STANDBYWFI status 1'b1 : A53/A72 warm reset is asserted after STANDBYWFI valid
6	RW	0x0	core_srst_wfien A53/A72 software reset wait for STANDBYWFI enable 1'b0 : A53/A72 software reset has no relation to STANDBYWFI status 1'b1 : A53/A72 software reset is asserted after STANDBYWFI valid
5	RW	0x0	dbgrstn_en A53/A72 DBGRSTN reset enable 1'b0 : disable A53/A72 DBGRSTN reset 1'b1 : enable A53/A72 DBGRSTN reset
4	RW	0x0	warmrstn_en A53/A72 warm reset enable 1'b0 : disable A53/A72 warm reset 1'b1 : enable A53/A72 warm reset

Bit	Attr	Reset Value	Description
3:0	RW	0x0	testclk_sel Output clock selection for test 4'h0: clk_core_b_2wrap 4'h1: clk_core_l_2wrap 4'h2: aclk_cci_2wrap 4'h3: aclk_perihp_2wrap 4'h4: aclk_perilp0_2wrap 4'h5: hclk_perilp1_2wrap 4'h6: aclk_center_2wrap 4'h7: clk_ddrc_2wrap 4'h8: aclk_gpu_2wrap 4'h9: clk_rga_core_2wrap 4'ha: clk_vdu_core_2wrap 4'hb: clk_pciephy_ref100m 4'hc: dclk_vop0_2wrap 4'hd: clk_rtc 4'he: clkout_24m 4'hf: clk_wifi

CRU_GLB_RST_CON
Address: Operational Base + offset (0x0510)
Global reset trigger select

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4	RW	0x0	pmu_glbrst_wdt_ctrl if pmu reset by wdt resetn src select 1'b0: pmu reset by wdt rstn 1'b1: pmu does not reset by wdt rstn
3:2	RW	0x0	pmu_glb_srst_ctrl pmu reset by global soft reset select 2'b00: pmu reset by first global soft reset 2'b01: pmu reset by second global soft reset 2'b10: pmu not reset by any global soft reset
1	RW	0x0	wdt_glb_srst_ctrl watch_dog trigger global soft reset select 1'b0: watch_dog trigger second global reset 1'b1: watch_dog trigger first global reset
0	RW	0x0	tsadc_glb_srst_ctrl TSADC trigger global soft reset select 1'b0: tsadc trigger second global reset 1'b1: tsadc trigger first global reset

CRU_GLB_RST_ST
Address: Operational Base + offset (0x0514)
Global reset status

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5	W1C	0x0	snd_glb_wdt_rst_st second global watch_dog triggered reset flag 1'b0: last hot reset is not second global watch_dog triggered reset 1'b1: last hot reset is second global watch_dog triggered reset
4	W1C	0x0	fst_glb_wdt_rst_st first global watch_dog triggered reset flag 1'b0: last hot reset is not first global watch_dog triggered reset 1'b1: last hot reset is first global watch_dog triggered reset
3	W1C	0x0	snd_glb_tsadc_rst_st second global TSADC triggered reset flag 1'b0: last hot reset is not second global TSADC triggered reset 1'b1: last hot reset is second global TSADC triggered reset
2	W1C	0x0	fst_glb_tsadc_rst_st first global TSADC triggered reset flag 1'b0: last hot reset is not first global TSADC triggered reset 1'b1: last hot reset is first global TSADC triggered reset
1	W1C	0x0	snd_glb_rst_st second global rst flag 1'b0: last hot reset is not second global reset 1'b1: last hot reset is second global reset
0	W1C	0x0	fst_glb_rst_st first global rst flag 1'b0: last hot reset is not first global reset 1'b1: last hot reset is first global reset

CRU_SDMMC_CON0
Address: Operational Base + offset (0x0580)
sdmmc control0

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:0	WO	0x0004	sdmmc_con0 sdmmc con0 register refer to chapter SDMMC

CRU_SDMMC_CON1
Address: Operational Base + offset (0x0584)
sdmmc control1

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:0	WO	0x0000	sdmmc_con1 sdmmc con1 register refer to chapter SDMMC

CRU_SDIO0_CON0
Address: Operational Base + offset (0x0588)
sdio0 control0

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:0	WO	0x0004	sdio_con0 sdio_con0 register refer to chapter SDIO

CRU_SDIO0_CON1
Address: Operational Base + offset (0x058c)
sdio0 control1

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:0	WO	0x0000	sdio_con1 sdio_con1 register refer to chapter SDIO

3.8 Timing Diagram

Power on reset timing is shown as follow:

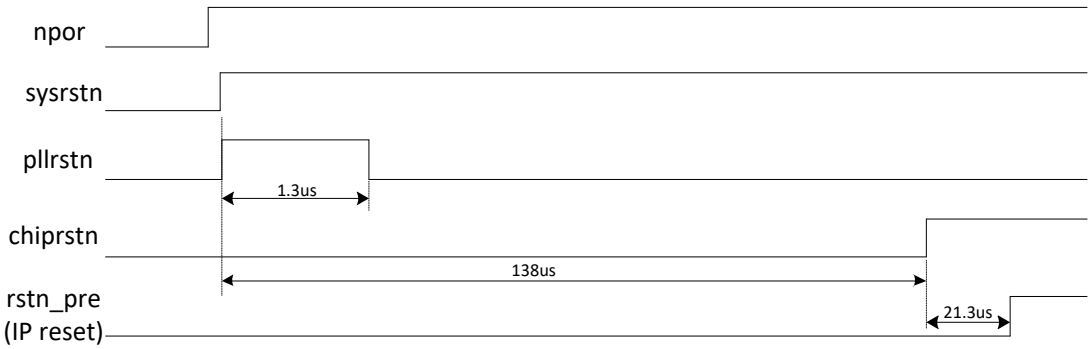


Fig. 3-7 Chip Power On Reset Timing Diagram

NPOR is hardware reset signal from out-chip, which is filtered glitch to obtain signal sysrstn. To make PLLs work normally, the PLL reset signal (pllrstn) must maintain high for more than 1us, and PLLs start to lock when pllrstn deassert, andthe PLL max lock time is 500 PLL REFCLK cycles. And then the system will wait about 138us, and then deactive reset signal chiprstn. The signal chiprstn is used to generate output clocks in CRU. After CRU start output clocks, the system waits again for 512cycles (21.3us) to deactive signal rstn_pre, which is used to generate power on reset of all IP.

3.9 Application Notes

3.9.1 PLL usage

The chip uses 3.2GHz for all 8 PLLs.

A. PLL output frequency configuration

FBDIV can be configured by programming CRU_*PLL_CON0;
REFDIV, POSTDIV1, POSTDIV2 can be configured by programming CRU_*PLL_CON1.
FRAC can be configured by programming CRU_*PLL_CON2.
BYPASS, PLL_WORK_MODE, POWER_DOWN, DSMPD can be configured by programming CRU_*PLL_CON3

If DSMPD = 1 (DSM is disabled, "integer mode")
FOUTVCO = FREF / REFDIV * FBDIV
FOUTPOSTDIV = FOUTVCO / POSTDIV1 / POSTDIV2
When FREF is 24MHz, and if 700MHz FOUTPOSTDIV is needed. The configuration can be:

DSMPD = 1
REFDIV = 6
FBDIV = 200
POSTDIV1=1
POSTDIV2=1

And then
FOUTVCO = FREF / REFDIV * FBDIV = 24/6*200=800
FOUTPOSTDIV = FOUTVCO / POSTDIV1 / POSTDIV2=800/1/1=800

If DSMPD = 0 (DSM is enabled, "fractional mode")
FOUTVCO = FREF / REFDIV * (FBDIV + FRAC / 224)
FOUTPOSTDIV = FOUTVCO / POSTDIV1 / POSTDIV2
When FREF is 24MHz, and if 491.52MHz FOUTPOSTDIV is needed. The configuration can be:

DSMPD = 0
REFDIV = 1
FBDIV = 40
FRAC = 24'hf5c28f
POSTDIV1=2
POSTDIV2=1

And then
FOUTVCO = FREF / REFDIV * (FBDIV + FRAC / 224) = 24/1*(40+24'hf5c28f /224)=

983.04
FOUTPOSTDIV = FOUTVCO / POSTDIV1 / POSTDIV2=983.04/2/1=491.52

B. PLL frequency range requirement

All the value range requirements are as follow.

FREF(Input Frequency Range in Integer Mode):	1MHz to 1200MHz
FREF(Input Frequency Range in Fractional Mode):	10MHz to 1200MHz
FREF/REFDIV(The divided reference frequency Integer Mode):	1 to FVCO/16
FREF/REFDIV(The divided reference frequency Fractional Mode):	10 to FVCO/16
FOUTVCO:	800MHz to 3.2GHz

C. PLL setting consideration

- If the POSTDIV value is changed during operation a short pulse (glitch) may occur on FOUTPOSTDIV. The minimum width of the short pulse will be equal to twice the period of the VCO. Therefore, if the circuitry clocked by the PLL is sensitive to short pulses, the new divide value should be re-timed so that it is synchronous with the rising edge of the output clock (FOUTPOSTDIV). Glitches cannot occur on any of the other outputs.
- For lowest power operation, the minimum VCO and FREF frequencies should be used. For minimum jitter operation, the highest VCO and FREF frequencies should be used. The normal operating range for the VCO is described above in .
- The supply rejection will be worse at the low end of the VCO range so care should be taken to keep the supply clean for low power applications.
- The feedback divider is not capable of dividing by all possible settings due to the use of a power-saving architecture. The following settings are valid for FBDIV:
- DSMPD=1 (Integer Mode):
- 12,13,14,16-4095 (practical value is limited to 3200, 2400, or 1600 (FVCOMAX / FREFMIN))
- DSMPD=0 (Fractional Mode):
- 19-4091 (practical value is limited to 320, 240, or 160 (FVCOMAX / FREFMIN))
- The PD input places the PLL into the lowest power mode. In this case, all analog circuits are turned off and FREF will be "ignored". The FOUTPOSTDIV and FOUTVCO pins are forced to logic low (0V).
- The BYPASS pin controls a mux which selects FREF to be passed to the FOUTPOSTDIV when active high. However, the PLL continues to run as it normally would if bypass were low. This is a useful feature for PLL testing since the clock path can be verified without the PLL being required to work. Also, the effect that the PLL induced supply noise has on the output buffering can be evaluated. It is not recommended to switch between BYPASS mode and normal mode for regular chip operation since this may result in a glitch. Also, FOUTPOSTDIVPD should be set low if the PLL is to be used in BYPASS mode.

3.9.2 PLL frequency change and lock check

The PLL programming support changed on-the-fly and the PLL will simply slew to the new frequency.

PLL lock state can be check in CRU_*PLL_CON3[31] register. The lock state is high when both original hardware PLL lock and PLL counter lock are high. The PLL counter lock initial value is CRU_GLB_CNT_TH[31:16].

The max delay time is 500 (REF_CLK / REFDIV) cycles.

Frequency change.

- Assert PD.
- Program the PLL to a valid setting that runs the VCO within the specified ranges.
- Release PD after no less than 1us from the time it was asserted. This will allow the power-down switch enough time to pull the loop filter voltage from rail-to-rail, which ensures the PLL will be completely powered down from any state.

PLL locking consists of three phases.

- Phase 1 is control voltage slewing. During this phase one of the clocks (reference or divide) is much faster than the other, and the PLL frequency adjusts almost

continuously. When locking from power down, the divide clock is initially very slow and steadily increases frequency. Slew time is about 2~5s. It will take slightly longer for faster VCO settings when locking from power down, since the PLL must slew further.

- Phase 2 is small signal phase acquisition. During this phase, the internal up/down signals alternate semi-chaotically as the phase slowly adjusts until the two signals are aligned. The duration of this phase depends on the loop bandwidth and is faster with higher bandwidth. Bandwidth can be estimated as FREF / REFDIV / 20 for integer mode and FREF /REFDIV / 40 for fractional mode. The duration of small signal locking is about 1/Bandwidth.
- Phase 3 is the digital cycle count. After the last cycle slip is detected, an internal counter waits 500 FREF / REFDIV cycles before the lock signal goes high. This is frequently the dominant factor in lock time – especially for slower reference clock signals or large reference divide settings. This time can be calculated as 500*REFDIV/FREF.

3.9.3 Fractional divider usage

To get specific frequency, clocks of I2S, SPDIF, UART can be generated by fractional divider. Generally you must set that denominator is 20 times larger than numerator to generate precise clock frequency. So the fractional divider applies only to generate low frequency clock like I2S, UART.

3.9.4 Global software reset

Two global software resets are designed in the chip, you can program CRU_GLB_SRST_FST_VALUE[15:0] as 0xfdb9 to assert the first global software reset glb_srstn_1 and program CRU_GLB_SRST_SND_VALUE[15:0] as 0xeca8 to assert the second global software reset glb_srstn_2. These two software resets are self-deasserted by hardware.

Glb_srstn_1 resets almost all logic.

Glb_srstn_2 resets almost all logic except GRF and GPIOs.

Chapter 4 General Register Files (GRF)

4.1 Overview

The general register file will be used to do static set by software, which is composed of many registers for system control. The GRF is divided into twosections,

- GRF, used for generalnon-secure system,
- PMUGRF, used for always on sysyem

4.2 Function Description

The function of general register file is：

- IOMUX control
- Control the state of GPIOin power-down mode
- GPIO PAD pulldown and pullup control
- Used for common system control
- Used to record the system state

4.3 GRF Register description

4.3.1 Register Summary

Name	Offset	Size	Reset Value	Description
GRF_USB3_PERF_CON0	0x02000	W	0x00000000	usb3 performance monitor control register
GRF_USB3_PERF_CON1	0x02004	W	0x00000000	usb3 performance monitor control register
GRF_USB3_PERF_CON2	0x02008	W	0x00000000	usb3 performance monitor control register
GRF_USB3_PERF_RD_MAX_LATENCY_NUM	0x0200c	W	0x00000000	usb3 performance monitor status register
GRF_USB3_PERF_RD_LATENCY_SAMP_NUM	0x02010	W	0x00000000	usb3 performance monitor status register
GRF_USB3_PERF_RD_LATENCY_ACC_NUM	0x02014	W	0x00000000	usb3 performance monitor status register
GRF_USB3_PERF_RD_AXI_TOTAL_BYTE	0x02018	W	0x00000000	usb3 performance monitor status register
GRF_USB3_PERF_WR_AXI_TOTAL_BYTE	0x0201c	W	0x00000000	usb3 performance monitor status register
GRF_USB3_PERF_WORKING_CNT	0x02020	W	0x00000000	usb3 performance monitor status register
GRF_USB3OTG0_CON0	0x02430	W	0x00002000	USB3 OTG0 GRF Register0

Name	Offset	Size	Reset Value	Description
GRF_USB3OTG0_CON1	0x02434	W	0x00001100	USB3 OTG0 GRF Register1
GRF_USB3OTG1_CON0	0x02440	W	0x00002000	USB3 OTG1 GRF Register0
GRF_USB3OTG1_CON1	0x02444	W	0x00001100	USB3 OTG1 GRF Register1
GRF_USB3OTG0_STATUS_LAT0	0x02450	W	0x00000000	USB3 OTG0 status register
GRF_USB3OTG0_STATUS_LAT1	0x02454	W	0x00000000	USB3 OTG1 status register
GRF_USB3OTG0_STATUS_CB	0x02458	W	0x00000000	USB3 OTG0 status register
GRF_USB3OTG1_STATUS_LAT0	0x02460	W	0x00000000	USB3 OTG1 status register
GRF_USB3OTG1_STATUS_LAT1	0x02464	W	0x00000000	USB3 OTG1 status register
GRF_USB3OTG1_STATUS_CB	0x02468	W	0x00000000	USB3 OTG1 status register
GRF_PCIE_PERF_CON0	0x04000	W	0x00000000	pcie performance monitor control register
GRF_PCIE_PERF_CON1	0x04004	W	0x00000000	pcie performance monitor control register
GRF_PCIE_PERF_CON2	0x04008	W	0x00000000	pcie performance monitor control register
GRF_PCIE_PERF_RD_MAX_LATENCY_NUM	0x0400c	W	0x00000000	pcieperformance monitor status register
GRF_PCIE_PERF_RD_LATENCY_SAMP_NUM	0x04010	W	0x00000000	pcie performance monitor status register
GRF_PCIE_PERF_RD_LATENCY_ACC_NUM	0x04014	W	0x00000000	pcie performance monitor status register
GRF_PCIE_PERF_RD_AXI_TOTAL_BYTE	0x04018	W	0x00000000	pcie performance monitor status register
GRF_PCIE_PERF_WR_AXI_TOTAL_BYTE	0x0401c	W	0x00000000	pcie performance monitor status register
GRF_PCIE_PERF_WORKING_CNT	0x04020	W	0x00000000	pcie performance monitor status register
GRF_USB20_HOST0_CON0	0x04100	W	0x000023e0	USB20 Host0 GRF register0
GRF_USB20_HOST0_CON1	0x04104	W	0x00000820	USB20 Host0 GRF register1

Name	Offset	Size	Reset Value	Description
GRF_USB20_HOST1_CON0	0x04110	W	0x000023e0	USB20 Host1 GRF register0
GRF_USB20_HOST1_CON1	0x04114	W	0x00000820	USB20 Host1 GRF register1
GRF_HSIC_CON0	0x04120	W	0x000002f0	HSIC controller GRF register 0
GRF_HSIC_CON1	0x04124	W	0x00000820	HSIC controller GRF register1
GRF_GRF_USBHOST0_STATUS	0x04140	W	0x00000000	usb host0 controller status register
GRF_GRF_USBHOST1_STATUS	0x04144	W	0x00000000	usb host1 controller status register
GRF_GRF_HSIC_STATUS	0x04148	W	0x00000000	hsic controller status register
GRF_HSICPHY_CON0	0x04470	W	0x0000004f	HSICPHY GRF control register
GRF_usbphy0_ctrl0	0x04480	W	0x0000850f	usbphy0_ctrl0
GRF_usbphy0_ctrl1	0x04484	W	0x0000e007	usbphy0_ctrl1
GRF_usbphy0_ctrl2	0x04488	W	0x000082e7	usbphy0_ctrl2
GRF_usbphy0_ctrl3	0x0448c	W	0x000002a2	usbphy0_ctrl3
GRF_usbphy0_ctrl4	0x04490	W	0x00005554	usbphy0_ctrl4
GRF_usbphy0_ctrl5	0x04494	W	0x00004555	usbphy0_ctrl5
GRF_usbphy0_ctrl6	0x04498	W	0x00000005	usbphy0_ctrl6
GRF_usbphy0_ctrl7	0x0449c	W	0x000068c8	usbphy0_ctrl7
GRF_usbphy0_ctrl8	0x044a0	W	0x00000000	usbphy0_ctrl8
GRF_usbphy0_ctrl9	0x044a4	W	0x00000000	usbphy0_ctrl9
GRF_usbphy0_ctrl10	0x044a8	W	0x00000000	usbphy0_ctrl10
GRF_usbphy0_ctrl11	0x044ac	W	0x00000000	usbphy0_ctrl11
GRF_usbphy0_ctrl12	0x044b0	W	0x000000a1	usbphy0_ctrl12
GRF_usbphy0_ctrl13	0x044b4	W	0x0000850f	usbphy0_ctrl13
GRF_usbphy0_ctrl14	0x044b8	W	0x0000e007	usbphy0_ctrl14
GRF_usbphy0_ctrl15	0x044bc	W	0x000002e7	usbphy0_ctrl15
GRF_usbphy0_ctrl16	0x044c0	W	0x00000200	usbphy0_ctrl16
GRF_usbphy0_ctrl17	0x044c4	W	0x00005554	usbphy0_ctrl17
GRF_usbphy0_ctrl18	0x044c8	W	0x00004555	usbphy0_ctrl18
GRF_usbphy0_ctrl19	0x044cc	W	0x00000005	usbphy0_ctrl19
GRF_usbphy0_ctrl20	0x044d0	W	0x000068c8	usbphy0_ctrl20
GRF_usbphy0_ctrl21	0x044d4	W	0x00000000	usbphy0_ctrl21
GRF_usbphy0_ctrl22	0x044d8	W	0x00000000	usbphy0_ctrl22
GRF_usbphy0_ctrl23	0x044dc	W	0x00000000	usbphy0_ctrl23
GRF_usbphy0_ctrl24	0x044e0	W	0x00000000	usbphy0_ctrl24
GRF_usbphy0_ctrl25	0x044e4	W	0x00000021	usbphy0_ctrl25
GRF_usbphy1_ctrl0	0x04500	W	0x0000850f	usbphy1_ctrl0

Name	Offset	Size	Reset Value	Description
GRF_usbphy1_ctrl1	0x04504	W	0x0000e007	usbphy1_ctrl1
GRF_usbphy1_ctrl2	0x04508	W	0x000082e7	usbphy1_ctrl2
GRF_usbphy1_ctrl3	0x0450c	W	0x000002a2	usbphy1_ctrl3
GRF_usbphy1_ctrl4	0x04510	W	0x00005554	usbphy1_ctrl4
GRF_usbphy1_ctrl5	0x04514	W	0x00004555	usbphy1_ctrl5
GRF_usbphy1_ctrl6	0x04518	W	0x00000005	usbphy1_ctrl6
GRF_usbphy1_ctrl7	0x0451c	W	0x000068c8	usbphy1_ctrl7
GRF_usbphy1_ctrl8	0x04520	W	0x00000000	usbphy1_ctrl8
GRF_usbphy1_ctrl9	0x04524	W	0x00000000	usbphy1_ctrl9
GRF_usbphy1_ctrl10	0x04528	W	0x00000000	usbphy1_ctrl10
GRF_usbphy1_ctrl11	0x0452c	W	0x00000000	usbphy1_ctrl11
GRF_usbphy1_ctrl12	0x04530	W	0x000000a1	usbphy1_ctrl12
GRF_usbphy1_ctrl13	0x04534	W	0x0000850f	usbphy1_ctrl13
GRF_usbphy1_ctrl14	0x04538	W	0x0000e007	usbphy1_ctrl14
GRF_usbphy1_ctrl15	0x0453c	W	0x000002e7	usbphy1_ctrl15
GRF_usbphy1_ctrl16	0x04540	W	0x00000200	usbphy1_ctrl16
GRF_usbphy1_ctrl17	0x04544	W	0x00005554	usbphy1_ctrl17
GRF_usbphy1_ctrl18	0x04548	W	0x00004555	usbphy1_ctrl18
GRF_usbphy1_ctrl19	0x0454c	W	0x00000005	usbphy1_ctrl19
GRF_usbphy1_ctrl20	0x04550	W	0x000068c8	usbphy1_ctrl20
GRF_usbphy1_ctrl21	0x04554	W	0x00000000	usbphy1_ctrl21
GRF_usbphy1_ctrl22	0x04558	W	0x00000000	usbphy1_ctrl22
GRF_usbphy1_ctrl23	0x0455c	W	0x00000000	usbphy1_ctrl23
GRF_usbphy1_ctrl24	0x04560	W	0x00000000	usbphy1_ctrl24
GRF_usbphy1_ctrl25	0x04564	W	0x00000021	usbphy1_ctrl25
GRF_HDCP22_PERF_CON0	0x06000	W	0x00000000	hdcp performance monitor control register
GRF_HDCP22_PERF_CON1	0x06004	W	0x00000000	hdcp performance monitor control register
GRF_HDCP22_PERF_CON2	0x06008	W	0x00000000	hdcp performance monitor control register
GRF_HDCP22_PERF_RD_MAX_LATENCY_NUM	0x0600c	W	0x00000000	hdcp performance monitor status register
GRF_HDCP22_PERF_RD_LATENCY_SAMP_NUM	0x06010	W	0x00000000	hdcp performance monitor status register
GRF_HDCP22_PERF_RD_LATENCY_ACC_NUM	0x06014	W	0x00000000	hdcp performance monitor status register
GRF_HDCP22_PERF_RD_AXI_TOTAL_BYTE	0x06018	W	0x00000000	hdcp performance monitor status register

Name	Offset	Size	Reset Value	Description
GRF_HDCP22_PERF_WR_AXI_TOTAL_BYTE	0x0601c	W	0x00000000	hdcp performance monitor status register
GRF_HDCP22_PERF_WORKING_CNT	0x06020	W	0x00000000	hdcp performance monitor status register
GRF_SOC_CON9	0x06224	W	0x00000000	SoC control register 9
GRF_SOC_CON20	0x06250	W	0x00000249	SoC control register 20
GRF_SOC_CON21	0x06254	W	0x000002cb	SoC control register 21
GRF_SOC_CON22	0x06258	W	0x000010cb	SoC control register 22
GRF_SOC_CON23	0x0625c	W	0x00000021	SoC control register 23
GRF_SOC_CON24	0x06260	W	0x000039f0	SoC control register 24
GRF_SOC_CON25	0x06264	W	0x0000d45b	SoC control register 25
GRF_SOC_CON26	0x06268	W	0x00000110	SoC control register 26
GRF_GPU_PERF_CON0	0x08000	W	0x00000000	gpu performance monitor control register
GRF_GPU_PERF_CON1	0x08004	W	0x00000000	gpu performance monitor control register
GRF_GPU_PERF_CON2	0x08008	W	0x00000000	gpu performance monitor control register
GRF_GPU_PERF_RD_MAX_LATENCY_NUM	0x0800c	W	0x00000000	gpu performance monitor status register
GRF_GPU_PERF_RD_LATENCY_SAMP_NUM	0x08010	W	0x00000000	gpu performance monitor status register
GRF_GPU_PERF_RD_LATENCY_ACC_NUM	0x08014	W	0x00000000	gpu performance monitor status register
GRF_GPU_PERF_RD_AXI_TOTAL_BYTE	0x08018	W	0x00000000	gpu performance monitor status register
GRF_GPU_PERF_WR_AXI_TOTAL_BYTE	0x0801c	W	0x00000000	gpu performance monitor status register
GRF_GPU_PERF_WORKING_CNT	0x08020	W	0x00000000	gpu performance monitor status register
GRF_CPU_CON0	0x0a000	W	0x0000000b	cpu control register 0
GRF_CPU_CON1	0x0a004	W	0x0000f000	cpu control register 1
GRF_CPU_CON2	0x0a008	W	0x0000000b	cpu control register 2
GRF_CPU_CON3	0x0a00c	W	0x00003110	cpu control register 3
GRF_CPU_STATUS0	0x0a080	W	0x00000000	cpu status register 0
GRF_CPU_STATUS1	0x0a084	W	0x00000000	cpu status register 1
GRF_CPU_STATUS2	0x0a088	W	0x00000000	cpu status register 2
GRF_CPU_STATUS3	0x0a08c	W	0x00000000	cpu status register 3
GRF_CPU_STATUS4	0x0a090	W	0x00000000	cpu status register 4
GRF_CPU_STATUS5	0x0a094	W	0x00000000	cpu status register 5

Name	Offset	Size	Reset Value	Description
GRF_A53_PERF_CON0	0x0a100	W	0x00000000	a53 performance monitor control register
GRF_A53_PERF_CON1	0x0a104	W	0x00000000	a53 performance monitor control register
GRF_A53_PERF_CON2	0x0a108	W	0x00000000	a53 performance monitor control register
GRF_A53_PERF_CON3	0x0a10c	W	0x00000000	a53 performance monitor control register
GRF_A53_PERF_RD_MON_ST	0x0a110	W	0x00000000	performance monitor read start address
GRF_A53_PERF_RD_MON_END	0x0a114	W	0x00000000	performance monitor end address
GRF_A53_PERF_WR_MON_ST	0x0a118	W	0x00000000	performance write monitor start address
GRF_A53_PERF_WR_MON_END	0x0a11c	W	0x00000000	performance monitor write end address
GRF_A53_PERF_RD_MAX_LATENCY_NUM	0x0a120	W	0x00000000	a53 performance monitor status register
GRF_A53_PERF_RD_LATENCY_SAMP_NUM	0x0a124	W	0x00000000	a53 performance monitor status register
GRF_A53_PERF_RD_LATENCY_ACC_NUM	0x0a128	W	0x00000000	a53 performance monitor status register
GRF_A53_PERF_RD_AXI_TOTAL_BYTE	0x0a12c	W	0x00000000	a53 performance monitor status register
GRF_A53_PERF_WR_AXI_TOTAL_BYTE	0x0a130	W	0x00000000	a53 performance monitor status register
GRF_A53_PERF_WORKING_CNT	0x0a134	W	0x00000000	a53 performance monitor status register
GRF_A53_PERF_INT_STATUS	0x0a138	W	0x00000000	a53 performance monitor status register
GRF_A72_PERF_CON0	0x0a200	W	0x00000000	a72 performance monitor control register
GRF_A72_PERF_CON1	0x0a204	W	0x00000000	a72 performance monitor control register
GRF_A72_PERF_CON2	0x0a208	W	0x00000000	a72 performance monitor control register
GRF_A72_PERF_CON3	0x0a20c	W	0x00000000	a72 performance monitor control register
GRF_A72_PERF_RD_MON_ST	0x0a210	W	0x00000000	performance monitor read start address
GRF_A72_PERF_RD_MON_END	0x0a214	W	0x00000000	performance monitor end address

Name	Offset	Size	Reset Value	Description
GRF_A72_PERF_WR_MON_ST	0x0a218	W	0x00000000	performance write monitor start address
GRF_A72_PERF_WR_MON_END	0x0a21c	W	0x00000000	performance monitor write end address
GRF_A72_PERF_RD_MAX_LATENCY_NUM	0x0a220	W	0x00000000	a72 performance monitor status register
GRF_A72_PERF_RD_LATENCY_SAMP_NUM	0x0a224	W	0x00000000	a72 performance monitor status register
GRF_A72_PERF_RD_LATENCY_ACC_NUM	0x0a228	W	0x00000000	a72 performance monitor status register
GRF_A72_PERF_RD_AXI_TOTAL_BYTE	0x0a22c	W	0x00000000	a72 performance monitor status register
GRF_A72_PERF_WR_AXI_TOTAL_BYTE	0x0a230	W	0x00000000	a72 performance monitor status register
GRF_A72_PERF_WORKING_CNT	0x0a234	W	0x00000000	a72 performance monitor status register
GRF_A72_PERF_INT_STATUS	0x0a238	W	0x00000000	a72 performance monitor status register
GRF_GMAC_PERF_CON0	0x0c000	W	0x00000000	gmac performance monitor control register
GRF_GMAC_PERF_CON1	0x0c004	W	0x00000000	gmac performance monitor control register
GRF_GMAC_PERF_CON2	0x0c008	W	0x00000000	gmac performance monitor control register
GRF_GMAC_PERF_RD_MAX_LATENCY_NUM	0x0c00c	W	0x00000000	gmac performance monitor status register
GRF_GMAC_PERF_RD_LATENCY_SAMP_NUM	0x0c010	W	0x00000000	gmac performance monitor status register
GRF_GMAC_PERF_RD_LATENCY_ACC_NUM	0x0c014	W	0x00000000	gmac performance monitor status register
GRF_GMAC_PERF_RD_AXI_TOTAL_BYTE	0x0c018	W	0x00000000	gmac performance monitor status register
GRF_GMAC_PERF_WR_AXI_TOTAL_BYTE	0x0c01c	W	0x00000000	gmac performance monitor status register
GRF_GMAC_PERF_WORKING_CNT	0x0c020	W	0x00000000	gmac performance monitor status register
GRF_SOC_CON5	0x0c214	W	0x00000008	SoC control register 5
GRF_SOC_CON6	0x0c218	W	0x00000000	SoC control register 6
GRF_GPIO2A_IOMUX	0x0e000	W	0x00000000	GPIO2A iomux control
GRF_GPIO2B_IOMUX	0x0e004	W	0x00000000	GPIO2B iomux control

Name	Offset	Size	Reset Value	Description
GRF_GPIO2C_IOMUX	0x0e008	W	0x00000000	GPIO2C iomux control
GRF_GPIO2D_IOMUX	0x0e00c	W	0x00000000	GPIO2D iomux control
GRF_GPIO3A_IOMUX	0x0e010	W	0x00000000	GPIO3A iomux control
GRF_GPIO3B_IOMUX	0x0e014	W	0x00000000	GPIO3B iomux control
GRF_GPIO3C_IOMUX	0x0e018	W	0x00000000	GPIO3C iomux control
GRF_GPIO3D_IOMUX	0x0e01c	W	0x00000000	GPIO3D iomux control
GRF_GPIO4A_IOMUX	0x0e020	W	0x00000000	GPIO4A iomux control
GRF_GPIO4B_IOMUX	0x0e024	W	0x00000000	GPIO4B iomux control
GRF_GPIO4C_IOMUX	0x0e028	W	0x00000000	GPIO4C iomux control
GRF_GPIO4D_IOMUX	0x0e02c	W	0x00000000	GPIO4D iomux control
GRF_GPIO2A_P	0x0e040	W	0x00006aa5	GPIO2A PU/PD control
GRF_GPIO2B_P	0x0e044	W	0x00000155	GPIO2B PU/PD control
GRF_GPIO2C_P	0x0e048	W	0x0000ffff	GPIO2C PU/PD control
GRF_GPIO2D_P	0x0e04c	W	0x0000007f	GPIO2D PU/PD control
GRF_GPIO3A_P	0x0e050	W	0x00005a5a	GPIO3A PU/PD control
GRF_GPIO3B_P	0x0e054	W	0x00005559	GPIO3B PU/PD control
GRF_GPIO3C_P	0x0e058	W	0x00000005	GPIO3C PU/PD control
GRF_GPIO3D_P	0x0e05c	W	0x0000aaaa	GPIO3D PU/PD control
GRF_GPIO4A_P	0x0e060	W	0x0000aa96	GPIO4A PU/PD control
GRF_GPIO4B_P	0x0e064	W	0x00000655	GPIO4B PU/PD control
GRF_GPIO4C_P	0x0e068	W	0x00006965	GPIO4C PU/PD control
GRF_GPIO4D_P	0x0e06c	W	0x00002aa9	GPIO4D PU/PD control
GRF_GPIO2A_SR	0x0e080	W	0x00000000	GPIO2A slew rate control
GRF_GPIO2B_SR	0x0e084	W	0x00000000	GPIO2B slew rate control
GRF_GPIO2C_SR	0x0e088	W	0x00000000	GPIO2C slew rate control
GRF_GPIO2D_SR	0x0e08c	W	0x00000000	GPIO2D slew rate control
GRF_GPIO3D_SR	0x0e09c	W	0x00000000	GPIO3D slew rate control
GRF_GPIO4A_SR	0x0e0a0	W	0x00000000	GPIO4A slew rate control
GRF_GPIO4B_SR	0x0e0a4	W	0x0000003f	GPIO4B slew rate control
GRF_GPIO4C_SR	0x0e0a8	W	0x00000000	GPIO4C slew rate control
GRF_GPIO4D_SR	0x0e0ac	W	0x00000000	GPIO4D slew rate control
GRF_GPIO2A_SMT	0x0e0c0	W	0x00000000	GPIO2A smitter control register
GRF_GPIO2B_SMT	0x0e0c4	W	0x00000000	GPIO2B smitter control register

Name	Offset	Size	Reset Value	Description
GRF_GPIO2C_SMT	0x0e0c8	W	0x00000000	GPIO2C smitter control register
GRF_GPIO2D_SMT	0x0e0cc	W	0x00000000	GPIO2D smitter control register
GRF_GPIO3A_SMT	0x0e0d0	W	0x000000f0	GPIO3A smitter control register
GRF_GPIO3B_SMT	0x0e0d4	W	0x00000000	GPIO3B smitter control register
GRF_GPIO3C_SMT	0x0e0d8	W	0x00000000	GPIO3C smitter control register
GRF_GPIO3D_SMT	0x0e0dc	W	0x00000000	GPIO3D smitter control register
GRF_GPIO4A_SMT	0x0e0e0	W	0x00000000	GPIO4A smitter control register
GRF_GPIO4B_SMT	0x0e0e4	W	0x0000003f	GPIO4B smitter control register
GRF_GPIO4C_SMT	0x0e0e8	W	0x00000000	GPIO4C smitter control register
GRF_GPIO4D_SMT	0x0e0ec	W	0x00000000	GPIO4D smitter control register
GRF_GPIO2A_E	0x0e100	W	0x00000000	GPIO2A drive strength control
GRF_GPIO2B_E	0x0e104	W	0x00000000	GPIO2B drive strength control
GRF_GPIO2C_E	0x0e108	W	0x00000000	GPIO2C drive strength control
GRF_GPIO2D_E	0x0e10c	W	0x00000000	GPIO2D drive strength control
GRF_GPIO3A_E01	0x0e110	W	0x00000000	GPIO3A drive strength control
GRF_GPIO3A_E2	0x0e114	W	0x00000000	GPIO3B drive strength control
GRF_GPIO3B_E01	0x0e118	W	0x00000000	GPIO3B drive strength control
GRF_GPIO3B_E2	0x0e11c	W	0x00000000	GPIO3B drive strength control
GRF_GPIO3C_E01	0x0e120	W	0x00000000	GPIO3C drive strength control
GRF_GPIO3C_E2	0x0e124	W	0x00000000	GPIO3C drive strength control
GRF_GPIO3D_E	0x0e128	W	0x00000000	GPIO3D drive strength control
GRF_GPIO4A_E	0x0e12c	W	0x00000000	GPIO4A drive strength control

Name	Offset	Size	Reset Value	Description
GRF_GPIO4B_E01	0x0e130	W	0x00009249	GPIO4B drive strength control
GRF_GPIO4B_E2	0x0e134	W	0x00000000	GPIO4B drive strength control
GRF_GPIO4C_E	0x0e138	W	0x00000000	GPIO4C drive strength control
GRF_GPIO4D_E	0x0e13c	W	0x00000000	GPIO4D drive strength control
GRF_GPIO2C_HE	0x0e188	W	0x00000000	GPIO2C HE control
GRF_GPIO2D_HE	0x0e18c	W	0x00000000	GPIO2D HE control
GRF_SOC_CON0	0x0e200	W	0x00000000	SoC control register 0
GRF_SOC_CON1	0x0e204	W	0x00000000	SoC control register 2
GRF_SOC_CON2	0x0e208	W	0x00000000	SoC control register 1
GRF_SOC_CON3	0x0e20c	W	0x00000000	SoC control register 3
GRF_SOC_CON4	0x0e210	W	0x0000010f	SoC control register 4
GRF_SOC_CON_5_PCIE	0x0e214	W	0x00000002	SoC control register 5
GRF_SOC_CON7	0x0e21c	W	0x00001000	SoC control register 7
GRF_SOC_CON8	0x0e220	W	0x00000000	SoC control register 8
GRF_SOC_CON_9_PCIE	0x0e224	W	0x00000000	SoC control register 9 for PCIE
GRF_SOC_STATUS0	0x0e2a0	W	0x00000003	SOC status register 0
GRF_SOC_STATUS1	0x0e2a4	W	0x00000000	SOC status register 1
GRF_SOC_STATUS2	0x0e2a8	W	0x00000000	SOC status register 2
GRF_SOC_STATUS3	0x0e2ac	W	0x00000000	SOC status register 3
GRF_SOC_STATUS4	0x0e2b0	W	0x00000000	SOC status register 4
GRF_SOC_STATUS5	0x0e2b4	W	0x00000000	SOC status register 5
GRF_DDRC0_CON0	0x0e380	W	0x00001f81	ddrc0 control register 0
GRF_DDRC0_CON1	0x0e384	W	0x00000000	ddrc0 control register 1
GRF_DDRC1_CON0	0x0e388	W	0x00001f81	ddrc1 control register 0
GRF_DDRC1_CON1	0x0e38c	W	0x00000000	ddrc1 control register 1
GRF_SIG_DETECT_CON0	0x0e3c0	W	0x00000000	Singal detect control register0
GRF_SIG_DETECT_CON1	0x0e3c8	W	0x00000000	Singal detect control register1
GRF_SIG_DETECT_CLR	0x0e3d0	W	0x00000000	Signal detect status clear register
GRF_SIG_DETECT_STATUS	0x0e3e0	W	0x00000000	Signal detect status register
GRF_USB20_PHY0_CON0	0x0e450	W	0x00000000	USB20 PHY0 GRF Register 0
GRF_USB20_PHY0_CON1	0x0e454	W	0x00001452	USB20 PHY0 GRF Register 1

Name	Offset	Size	Reset Value	Description
GRF_USB20_PHY0_CON2	0x0e458	W	0x000003d2	USB20 PHY0 GRF Register 2
GRF_USB20_PHY0_CON3	0x0e45c	W	0x00000001	USB20 PHY0 GRF Register 3
GRF_USB20_PHY1_CON0	0x0e460	W	0x00000000	USB20 PHY1 GRF Register 0
GRF_USB20_PHY1_CON1	0x0e464	W	0x00001452	USB20 PHY1GRF Register 1
GRF_USB20_PHY1_CON2	0x0e468	W	0x000003d2	USB20 PHY1 GRF Register 2
GRF_USB20_PHY1_CON3	0x0e46c	W	0x00000001	USB20 PHY1 GRF Register 3
GRF_USB3PHY0_CON0	0x0e580	W	0x000099c8	TypeC PHY/TCPD PHY/TCPC Control register0
GRF_USB3PHY0_CON1	0x0e584	W	0x00001000	TypeC PHY/TCPD PHY/TCPC Control register1
GRF_USB3PHY0_CON2	0x0e588	W	0x00003cc8	TypeC PHY/TCPD PHY/TCPC Control register2
GRF_USB3PHY1_CON0	0x0e58c	W	0x000019c8	TypeC PHY/TCPD PHY/TCPC Control register0
GRF_USB3PHY1_CON1	0x0e590	W	0x00001000	TypeC PHY/TCPD PHY/TCPC Control register1
GRF_USB3PHY1_CON2	0x0e594	W	0x00003cc8	TypeC PHY/TCPD PHY/TCPC Control register2
GRF_USB3PHY_STATUS0	0x0e5c0	W	0x00000000	USB3PHY_STATUS0
GRF_USB3PHY_STATUS1	0x0e5c4	W	0x00000000	USB3PHY_STATUS1
GRF_DLL_CON0	0x0e600	W	0x00000000	pvtm control register
GRF_DLL_CON1	0x0e604	W	0x016e3600	pvtm control register
GRF_DLL_CON2	0x0e608	W	0x016e3600	pvtm control register
GRF_DLL_CON3	0x0e60c	W	0x016e3600	pvtm control register
GRF_DLL_CON4	0x0e610	W	0x016e3600	pvtm control register
GRF_DLL_CON5	0x0e614	W	0x00000000	pvtm control register
GRF_DLL_STATUS0	0x0e620	W	0x00000000	pvtm status register
GRF_DLL_STATUS1	0x0e624	W	0x00000000	pvtm status register
GRF_DLL_STATUS2	0x0e628	W	0x00000000	pvtm status register
GRF_DLL_STATUS3	0x0e62c	W	0x00000000	pvtm status register

Name	Offset	Size	Reset Value	Description
GRF_DLL_STATUS4	0x0e630	W	0x00000000	pvtm status register
GRF_IO_VSEL	0x0e640	W	0x00000000	
GRF_SARADC_TESTBIT	0x0e644	W	0x00000000	saradc test bit control register
GRF_TSADC_TESTBIT_L	0x0e648	W	0x00000000	saradc test bit control register
GRF_TSADC_TESTBIT_H	0x0e64c	W	0x00000000	tsadc test bit control register
GRF_CHIP_ID_ADDR	0x0e800	W	0x00000000	chip id register
GRF_FAST_BOOT_ADDR	0x0e880	W	0x00000000	faster boot address register
GRF_EMMCCORE_CON0	0x0f000	W	0x00000000	emmc core control register
GRF_EMMCCORE_CON1	0x0f004	W	0x00000000	emmc core control register
GRF_EMMCCORE_CON2	0x0f008	W	0x00000000	emmc core control register
GRF_EMMCCORE_CON3	0x0f00c	W	0x00000000	emmc core control register
GRF_EMMCCORE_CON4	0x0f010	W	0x00000000	emmc core control register
GRF_EMMCCORE_CON5	0x0f014	W	0x00000000	emmc core control register
GRF_EMMCCORE_CON6	0x0f018	W	0x00000000	emmc core control register
GRF_EMMCCORE_CON7	0x0f01c	W	0x00000000	emmc core control register
GRF_EMMCCORE_CON8	0x0f020	W	0x00000000	emmc core control register
GRF_EMMCCORE_CON9	0x0f024	W	0x00000000	emmc core control register
GRF_EMMCCORE_CON10	0x0f028	W	0x00000000	emmc core control register
GRF_EMMCCORE_CON11	0x0f02c	W	0x00000000	emmc core control register
GRF_EMMCCORE_STATUS0	0x0f040	W	0x00000000	emmc core status register
GRF_EMMCCORE_STATUS1	0x0f044	W	0x00000000	emmc core status register
GRF_EMMCCORE_STATUS2	0x0f048	W	0x00000000	emmc core status register
GRF_EMMCCORE_STATUS3	0x0f04c	W	0x00000000	emmc core status register

Name	Offset	Size	Reset Value	Description
GRF_EMMCPHY_CON0	0x0f780	W	0x00000000	emmc phy control register
GRF_EMMCPHY_CON1	0x0f784	W	0x00000000	emmc phy control register
GRF_EMMCPHY_CON2	0x0f788	W	0x00000000	emmc phy control register
GRF_EMMCPHY_CON3	0x0f78c	W	0x00000000	emmc phy control register
GRF_EMMCPHY_CON4	0x0f790	W	0x00000000	emmc phy control register
GRF_EMMCPHY_CON5	0x0f794	W	0x00000000	emmc phy control register
GRF_EMMCPHY_CON6	0x0f798	W	0x00000000	emmc phy control register
GRF_EMMCPHY_STATUS	0x0f7a0	W	0x00000000	emmc phy status register

*Notes:Size:**B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access*

4.3.2 Detail Register Description

GRF_USB3_PERF_CON0

Address: Operational Base + offset (0x02000)

usb3 performance monitor control register

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15	RW	0x0	usb3_rksoc_axi_perf_sel 0: usb3otg0 1: usb3otg1
14:12	RO	0x0	reserved

Bit	Attr	Reset Value	Description
11:8	RW	0x0	usb3_sw_rd_latency_id Axi read channel id for latency AXI_PERFormance test
7	RO	0x0	reserved
6:5	RW	0x0	usb3_sw_ddr_align_type 0: 16-Byte align 1: 32-Byte align 2: 64-Byte align 3: 128-Byte align
4	RW	0x0	usb3_sw_aw_cnt_id_type axi_perf counter id control 0: count all read channel id 1: count sw_ar_count_id read channel only
3	RW	0x0	usb3_sw_ar_cnt_id_type axi_perf counter id control 0: count all read channel id 1: count sw_ar_count_id read channel only
2	RW	0x0	usb3_sw_axi_cnt_type axi_perf counter type 0: axi transfer test 1: ddr align transfer test
1	RW	0x0	usb3_sw_axi_perf_clr Fi axi_perf clear bit 0: disable 1: enable
0	RW	0x0	usb3_sw_axi_perf_work axi_perf enable bit 0: disable 1: enable

GRF_USB3_PERF_CON1
Address: Operational Base + offset (0x02004)
usb3 performance monitor control register

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:12	RO	0x0	reserved
11:0	RW	0x000	usb3_sw_rd_latency_thr Axi read channel id for latency AXI_PERFormance test

GRF_USB3_PERF_CON2
Address: Operational Base + offset (0x02008)
usb3 performance monitor control register

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:12	RO	0x0	reserved
11:8	RW	0x0	usb3_sw_aw_count_id When sw_aw_cnt_id_type=1, only count the id designated by sw_aw_count_id
7:4	RO	0x0	reserved

Bit	Attr	Reset Value	Description
3:0	RW	0x0	usb3_sw_ar_count_id When sw_ar_cnt_id_type=1, only count the id designated by sw_ar_count_id

GRF_USB3_PERF_RD_MAX_LATENCY_NUM
Address: Operational Base + offset (0x0200c)
usb3 performance monitor status register

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	rd_max_latency_r axi read max latency oaxi read max latency outpututput

GRF_USB3_PERF_RD_LATENCY_SAMP_NUM
Address: Operational Base + offset (0x02010)
usb3 performance monitor status register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rd_latency_samp_r AXI read latency total sample number

GRF_USB3_PERF_RD_LATENCY_ACC_NUM
Address: Operational Base + offset (0x02014)
usb3 performance monitor status register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rd_latency_acc_cnt_r AXI read latency (>sw_rd_latency_thr) total number

GRF_USB3_PERF_RD_AXI_TOTAL_BYTE
Address: Operational Base + offset (0x02018)
usb3 performance monitor status register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rd_axi_total_byte AXI active total read bytes/ddr align read bytes

GRF_USB3_PERF_WR_AXI_TOTAL_BYTE
Address: Operational Base + offset (0x0201c)
usb3 performance monitor status register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	wr_axi_total_byte AXI active total write bytes/ddr align write bytes

GRF_USB3_PERF_WORKING_CNT
Address: Operational Base + offset (0x02020)
usb3 performance monitor status register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	working_cnt_r working counter

GRF_USB3OTG0_CON0
Address: Operational Base + offset (0x02430)
USB3 OTG0 GRF Register0

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15	RW	0x0	host_u2_port_disable host_u2_port_disable USB2.0 Port Disable control. 0: Port Enabled 1: Port Disabled When 1, this signal stops reporting connect/disconnect events the port and keeps the port in disabled state.
14	RW	0x0	host_port_power_control_present host_port_power_control_present This indicates whether the host controller implementation includes port power control. 0: Indicates that the port does not have port power switches. 1: Indicates that the port has port power switches

Bit	Attr	Reset Value	Description
13:8	RW	0x20	fladj_30mhz_reg fladj_30mhz_reg HS Jitter Adjustment. Indicates the correction required to accommodate mac3 clock and utmi clock jitter to measure 125 's duration. With fladj_30mhz_reg tied to zero, the high speed 125us micro-frame is counted for 123933ns. You must program the value in terms of high speed bit times in a 30 MHz cycle. The default value that must be driven is 32 (assuming 30 MHz perfect clock).
7:6	RW	0x0	hub_port_perm_attach hub_port_perm_attach Indicates if the device attached to a downstream port is permanently attached or not. 0: Not permanently attached 1: Permanently attached Bit0 is for USB2.0 port and bit1 are for USB 3.0 SS port.
5:4	RW	0x0	hub_port_overcurrent hub_port_overcurrent This is the per port Overcurrent indication of the root-hub ports: 0: No Overcurrent 1: Overcurrent Bit0 is for USB 2.0 port and bit1 are for USB 3.0 SS port.

Bit	Attr	Reset Value	Description
3:0	RW	0x0	bus_filter_bypass bus_filter_bypass It is expected that this signal is set or reset at power-on reset and is not changed during the normal operation of the core. The function of each bit is: bus_filter_bypass[3]: Bypass the filter for utmiotg_iddig bus_filter_bypass[2]: Bypass the filters for utmisrp_bvalid and utmisrp_sessend bus_filter_bypass[1]: Bypass the filter for pipe3_PowerPresent all U3 ports bus_filter_bypass[0]: Bypass the filter for utmiotg_vbusvalid all U2 ports In non-OTG Host-only mode, internal bus filters are not needed. Values: 1'b0: Bus filter(s) enabled 1'b1: Bus filter(s) disabled (bypassed)

GRF_USB3OTG0_CON1
Address: Operational Base + offset (0x02434)
USB3 OTG0 GRF Register1

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:12	RW	0x1	host_u3_port host_u3_port xHCI usb3 port number, default as 1.

Bit	Attr	Reset Value	Description
11:8	RW	0x1	host_u2_port host_u2_port xHCI host USB2 Port number, default as 1.
7:6	RO	0x0	reserved
5	RW	0x0	host_legacy_smi_bar host_legacy_smi_bar Use this register to support SMI on BAR defined in xHCI spec. SW must set this register, then clear this register to indicate Base Address Register written
4	RW	0x0	host_legacy_smi_pci_cmd host_legacy_smi_pci_cmd Use this register to support SMI on PCI Command defined in xHCI spec. SW must set this register, then clear this register to indicate PCI command register written.
3:2	RO	0x0	reserved
1	RW	0x0	pme_en pme_en Enable signal for the pme_generation. Enable the core to assert pme_generation.
0	RW	0x0	host_u3_port_disable host_u3_port_disable USB 3.0 SS Port Disable control. 0: Port Enabled 1: Port Disabled

GRF_USB3OTG1_CON0
Address: Operational Base + offset (0x02440)
USB3 OTG1 GRF Register0

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15	RW	0x0	host_u2_port_disable host_u2_port_disable USB2.0 Port Disable control. 0: Port Enabled 1: Port Disabled When 1, this signal stops reporting connect/disconnect events the port and keeps the port in disabled state.
14	RW	0x0	host_port_power_control_present host_port_power_control_present This indicates whether the host controller implementation includes port power control. 0: Indicates that the port does not have port power switches. 1: Indicates that the port has port power switches
13:8	RW	0x20	fladj_30mhz_reg fladj_30mhz_reg HS Jitter Adjustment. Indicates the correction required to accommodate mac3 clock and utmi clock jitter to measure 125 's duration. With fladj_30mhz_reg tied to zero, the high speed 125us micro-frame is counted for 123933ns. You must program the value in terms of high speed bit times in a 30 MHz cycle. The default value that must be driven is 32 (assuming 30 MHz perfect clock).

Bit	Attr	Reset Value	Description
7:6	RW	0x0	hub_port_perm_attach hub_port_perm_attach Indicates if the device attached to a downstream port is permanently attached or not. 0: Not permanently attached 1: Permanently attached Bit0 is for USB2.0 port and bit1 are for USB 3.0 SS port.
5:4	RW	0x0	hub_port_overcurrent hub_port_overcurrent This is the per port Overcurrent indication of the root-hub ports: 0: No Overcurrent 1: Overcurrent Bit0 is for USB 2.0 port and bit1 are for USB 3.0 SS port.
3:0	RW	0x0	bus_filter_bypass bus_filter_bypass It is expected that this signal is set or reset at power-on reset and is not changed during the normal operation of the core. The function of each bit is: bus_filter_bypass[3]: Bypass the filter for utmiotg_iddig bus_filter_bypass[2]: Bypass the filters for utmisrp_bvalid and utmisrp_sessend bus_filter_bypass[1]: Bypass the filter for pipe3_PowerPresent all U3 ports bus_filter_bypass[0]: Bypass the filter for utmiotg_vbusvalid all U2 ports In non-OTG Host-only mode, internal bus filters are not needed. Values: 1'b0: Bus filter(s) enabled 1'b1: Bus filter(s) disabled (bypassed)

GRF_USB3OTG1_CON1
Address: Operational Base + offset (0x02444)
USB3 OTG1 GRF Register1

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:12	RW	0x1	host_u3_port host_u3_port xHCI usb3 port number, default as 1.
11:8	RW	0x1	host_u2_port host_u2_port xHCI host USB2 port number, default as 1.
7:6	RO	0x0	reserved
5	RW	0x0	host_legacy_smi_bar host_legacy_smi_bar Use this register to support SMI on BAR defined in xHCI spec. SW must set this register, then clear this register to indicate Base Address Register written
4	RW	0x0	host_legacy_smi_pci_cmd host_legacy_smi_pci_cmd Use this register to support SMI on PCI Command defined in xHCI spec. SW must set this register, then clear this register to indicate PCI command register written.
3:2	RO	0x0	reserved
1	RW	0x0	pme_en pme_en Enable signal for the pme_generation. Enable the core to assert pme_generation.

Bit	Attr	Reset Value	Description
0	RW	0x0	host_u3_port_disable host_u3_port_disable USB 3.0 SS Port Disable control. 0: Port Enabled 1: Port Disabled

GRF_USB3OTG0_STATUS_LAT0
Address: Operational Base + offset (0x02450)
USB3 OTG0 status register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	usbcphy0_otg_utmi_iddig status of usbcphy0_otg_utmi_iddig[31:0]

GRF_USB3OTG0_STATUS_LAT1
Address: Operational Base + offset (0x02454)
USB3 OTG1 status register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	usbcphy0_otg_utmi_iddig status of usbcphy0_otg_utmi_iddig[63:32]

GRF_USB3OTG0_STATUS_CB
Address: Operational Base + offset (0x02458)
USB3 OTG0 status register

Bit	Attr	Reset Value	Description
31:12	RO	0x0	reserved
11:0	RW	0x000	usb3otg0_host_current_belt status of usb3otg0_host_current_belt

GRF_USB3OTG1_STATUS_LAT0
Address: Operational Base + offset (0x02460)
USB3 OTG1 status register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	usbcphy1_otg_utmi_iddig status of usbcphy1_otg_utmi_iddig[31:0]

GRF_USB3OTG1_STATUS_LAT1
Address: Operational Base + offset (0x02464)
USB3 OTG1 status register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	usbcphy1_otg_utmi_iddig status of usbcphy1_otg_utmi_iddig[63:32]

GRF_USB3OTG1_STATUS_CB
Address: Operational Base + offset (0x02468)

USB3 OTG1 status register

Bit	Attr	Reset Value	Description
31:12	RO	0x0	reserved
11:0	RW	0x000	usb3otg1_host_current_belt status of usb3otg1_host_current_belt

GRF_PCIE_PERF_CON0

Address: Operational Base + offset (0x04000)

pcie performance monitor control register

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:13	RO	0x0	reserved
12:8	RW	0x00	pcie_sw_rd_latency_id Axi read channel id for latency AXI_PERFormance test
7	RO	0x0	reserved
6:5	RW	0x0	pcie_sw_ddr_align_type 0: 16-Byte align 1: 32-Byte align 2: 64-Byte align 3: 128-Byte align
4	RW	0x0	pcie_sw_aw_cnt_id_type 0: count all write channels 1: count sw_aw_count_id write channel only
3	RW	0x0	pcie_sw_ar_cnt_id_type axi_perf counter id control 0: count all read channel id 1: count sw_ar_count_id read channel only
2	RW	0x0	pcie_sw_axi_cnt_type axi_perf counter type 0: axi transfer test 1: ddr align transfer test

Bit	Attr	Reset Value	Description
1	RW	0x0	pcie_sw_axi_perf_clr axi_perf clear bit 0: disable 1: enable
0	RW	0x0	pcie_sw_axi_perf_work axi_perf enable bit 0: disable 1: enable

GRF_PCIE_PERF_CON1
Address: Operational Base + offset (0x04004)
pcie performance monitor control register

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:12	RO	0x0	reserved
11:0	RW	0x000	pcie_sw_rd_latency_thr Axi read channel id for latency AXI_PERFormance test

GRF_PCIE_PERF_CON2
Address: Operational Base + offset (0x04008)
pcie performance monitor control register

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:13	RO	0x0	reserved
12:8	RW	0x00	pcie_sw_aw_count_id When sw_aw_cnt_id_type=1, only count the id designated by sw_aw_count_id
7:5	RO	0x0	reserved
4:0	RW	0x00	pcie_sw_ar_count_id When sw_ar_cnt_id_type=1, only count the id designated by sw_ar_count_id

GRF_PCIE_PERF_RD_MAX_LATENCY_NUM
Address: Operational Base + offset (0x0400c)
pcieperformance monitor status register

Bit	Attr	Reset Value	Description
31:13	RO	0x0	reserved
12:0	RW	0x0000	rd_max_latency_r axi read max latency output

GRF_PCIE_PERF_RD_LATENCY_SAMP_NUM
Address: Operational Base + offset (0x04010)
pcie performance monitor status register

Bit	Attr	Reset Value	Description
31:27	RO	0x0	reserved
26:0	RW	0x0000000	rd_latency_samp_r AXI read latency total sample number

GRF_PCIE_PERF_RD_LATENCY_ACC_NUM
Address: Operational Base + offset (0x04014)
pcie performance monitor status register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rd_latency_acc_cnt_r AXI read latency (>sw_rd_latency_thr) total number

GRF_PCIE_PERF_RD_AXI_TOTAL_BYTE
Address: Operational Base + offset (0x04018)
pcie performance monitor status register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rd_axi_total_byte AXI active total read bytes/ddr align read bytes

GRF_PCIE_PERF_WR_AXI_TOTAL_BYTE
Address: Operational Base + offset (0x0401c)
pcie performance monitor status register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	wr_axi_total_byte AXI active total write bytes/ddr align write bytes

GRF_PCIE_PERF_WORKING_CNT
Address: Operational Base + offset (0x04020)
pcie performance monitor status register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	working_cnt_r working counter

GRF_USB20_HOST0_CON0
Address: Operational Base + offset (0x04100)
USB20 Host0 GRF register0

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:14	RO	0x0	reserved
13	RW	0x1	word_if word_if 1: select 16bit utmi interface 0: select 8bit utmi interface Note: usb2phy only support 16bit interface.
12	RW	0x0	sim_mode sim_mode Simulation only.
11	RW	0x0	ohci_susp_lgcy ohci_susp_lgcy
10	RW	0x0	ohci_cntsel ohci_cntsel
9	RW	0x1	ohci_clkcktrst ohci_clkcktrst

Bit	Attr	Reset Value	Description
8	RW	0x1	incrx_en incrx_en Forces AHB master to start INCR4/8/16 busts only on burst boundaries. AHB requires that double word width burst be addressed-aligned only to the double-word boundary. 1'b1: Start INCRX burst only on burst x-aligned addresses 1'b0: Normal AHB operation; start bursts on any double word boundary Note: When this function is enabled, the burst are started only when the lowest bits of haddr are: INCR4: haddr[3:0] == 4'b0000 INCR8: haddr[4:0] == 5'b00000 INCR16: haddr[5:0] == 6'b000000
7	RW	0x1	incr8_en incr8_en 1: enable AHB INCR8 burst 0: disable AHB INCR8 burst
6	RW	0x1	incr4_en incr4_en 1: enable AHB INCR4 burst 0: disable AHB INCR4 burst
5	RW	0x1	incr16_en incr16_en 1: enable AHB INCR16 burst 0: disable AHB INCR16 burst
4	RW	0x0	hubsetup_min hubsetup_min
3	RW	0x0	autoppd_on_overcur_en autoppd_on_overcur_en
2	RW	0x0	arb_pause arb_pause
1	RW	0x0	app_start_clk app_start_clk
0	RW	0x0	app_prt_ovrcur app_prt_ovrcur

GRF_USB20_HOST0_CON1
Address: Operational Base + offset (0x04104)
USB20 Host0 GRF register1

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:12	RO	0x0	reserved
11:6	RW	0x20	fladj_val_common fladj_val_common Must set this register to 0x20.
5:0	RW	0x20	fladj_val fladj_val Must set this register to 0x20.

GRF_USB20_HOST1_CON0
Address: Operational Base + offset (0x04110)
USB20 Host1 GRF register0

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:14	RO	0x0	reserved

Bit	Attr	Reset Value	Description
13	RW	0x1	word_if word_if 1: select 16bit utmi interface 0: select 8bit utmi interface Note: usb2phy only support 16bit interface.
12	RW	0x0	sim_mode sim_mode Simulation only.
11	RW	0x0	ohci_susp_lgcy ohci_susp_lgcy
10	RW	0x0	ohci_cntsel ohci_cntsel
9	RW	0x1	ohci_clkcktrst ohci_clkcktrst
8	RW	0x1	incrx_en incrx_en Forces AHB master to start INCR4/8/16 busts only on burst boundaries. AHB requires that double word width burst be addressed-aligned only to the double-word boundary. 1'b1: Start INCRX burst only on burst x-aligned addresses 1'b0: Normal AHB operation; start bursts on any double word boundary Note: When this function is enabled, the burst are started only when the lowest bits of haddr are: INCR4: haddr[3:0] == 4'b0000 INCR8: haddr[4:0] == 5'b00000 INCR16: haddr[5:0] == 6'b000000
7	RW	0x1	incr8_en incr8_en 1: enable AHB INCR8 burst 0: disable AHB INCR8 burst
6	RW	0x1	incr4_en incr4_en 1: enable AHB INCR4 burst 0: disable AHB INCR4 burst
5	RW	0x1	incr16_en incr16_en 1: enable AHB INCR16 burst 0: disable AHB INCR16 burst
4	RW	0x0	hubsetup_min hubsetup_min

Bit	Attr	Reset Value	Description
3	RW	0x0	autoppd_on_overcur_en autoppd_on_overcur_en
2	RW	0x0	arb_pause arb_pause
1	RW	0x0	app_start_clk app_start_clk
0	RW	0x0	app_prt_ovrcur app_prt_ovrcur

GRF_USB20_HOST1_CON1
Address: Operational Base + offset (0x04114)
USB20 Host1 GRF register1

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:12	RO	0x0	reserved
11:6	RW	0x20	fladj_val_common fladj_val_common Must set this register to 0x20.
5:0	RW	0x20	fladj_val fladj_val Must set this register to 0x20.

GRF_HSIC_CON0
Address: Operational Base + offset (0x04120)
HSIC controller GRF register 0

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:10	RO	0x0	reserved
9	RW	0x1	hsic_word_if word_if 1: select 16bit utmi interface 0: select 8bit utmi interface Note: HSICPHY only support 16bit utmi interface.
8	RW	0x0	hsic_sim_mode sim_mode Simulation only.
7	RW	0x1	hsic_incrx_en Burst Alignment Enable Forces AHB master to start INCR4/8/16 busts only on burst boundaries. AHB requires that double word width burst be addressed-aligned only to the double-word boundary. 1'b1: Start INCRX burst only on burst x-aligned addresses 1'b0: Normal AHB operation; start bursts on any double word boundary Note: When this function is enabled, the burst are started only when the lowest bits of haddr are: INCR4: haddr[3:0] == 4'b0000 INCR8: haddr[4:0] == 5'b00000 INCR16: haddr[5:0] == 6'b000000
6	RW	0x1	hsic_incr8_en incr8_en 1: enable AHB INCR8 burst 0: disable AHB INCR8 burst

Bit	Attr	Reset Value	Description
5	RW	0x1	hsic_incr4_en incr4_en 1: enable AHB INCR4 burst 0: disable AHB INCR4 burst
4	RW	0x1	hsic_incr16_en incr16_en 1: enable AHB INCR16 burst 0: disable AHB INCR16 burst
3	RW	0x0	hsic_hubsetup_min hubsetup_min
2	RW	0x0	hsic_autoppd_on_overcur autoppd_on_overcur
1	RW	0x0	hsic_app_start_clk app_start_clk
0	RW	0x0	hsic_app_prt_ovrcur app_prt_ovrcur

GRF_HSIC_CON1
Address: Operational Base + offset (0x04124)
HSIC controller GRF register1

Bit	Attr	Reset Value	Description
31:13	RO	0x0	reserved
12	RW	0x0	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
11:6	RW	0x20	hsic_fladj_val_common fladj_val_common Must set this register to 0x20
5:0	RW	0x20	hsic_fladj fladj Must set this register to 0x20.

GRF_GRF_USBHOST0_STATUS

Address: Operational Base + offset (0x04140)
usb host0 controller status register

Bit	Attr	Reset Value	Description
31:29	RO	0x0	reserved
28	RW	0x0	host0_ehci_bufacc
27	RW	0x0	host0_ehci_xfer_prdc
26	RW	0x0	host0_ohci_bufacc
25	RW	0x0	host0_ohci_ccs
24	RW	0x0	host0_ohci_drwe
23	RW	0x0	host0_ohci_globalsuspend
22	RW	0x0	host0_ohci_rmtwkp
21	RW	0x0	host0_ohci_rwe
20:17	RW	0x0	host0_ehci_lpsmc_state
16:11	RW	0x00	host0_ehci_usbsts
10:0	RW	0x000	host0_ehci_xfer_cnt

GRF_GRF_USBHOST1_STATUS
Address: Operational Base + offset (0x04144)
usb host1 controller status register

Bit	Attr	Reset Value	Description
31:29	RO	0x0	reserved
28	RW	0x0	host1_ehci_bufacc
27	RW	0x0	host1_ehci_xfer_prdc
26	RW	0x0	host1_ohci_bufacc
25	RW	0x0	host1_ohci_ccs
24	RW	0x0	host1_ohci_drwe
23	RW	0x0	host1_ohci_globalsuspend
22	RW	0x0	host1_ohci_rmtwkp
21	RW	0x0	host1_ohci_rwe
20:17	RW	0x0	host1_ehci_lpsmc_state
16:11	RW	0x00	host1_ehci_usbsts
10:0	RW	0x000	host1_ehci_xfer_cnt

GRF_GRF_HSIC_STATUS
Address: Operational Base + offset (0x04148)
hsic controller status register

Bit	Attr	Reset Value	Description
31:22	RO	0x0	reserved
21	RW	0x0	hsic_ehci_xfer_prdc
20:17	RW	0x0	hsic_ehci_lpsms_state
16:11	RW	0x00	hsic_ehci_usbsts
10:0	RW	0x000	hsic_ehci_xfer_cnt

GRF_HSICPHY_CON0
Address: Operational Base + offset (0x04470)

HSICPHY GRF control register

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:9	RO	0x0	reserved
8	RW	0x0	hsicphy_soft_con_sel soft_con_sel 1: soft control select utmi signals from GRF to HSIC PHY 0: soft control select utmi signals from HSIC controller to HSIC PHY
7:6	RW	0x1	i_hsic_utmi_xcvrselect utmi_xcvrselect select the value of this register to xcvrselect port of HSIC PHY when soft_con_sel=1.
5:4	RW	0x0	i_hsic_utmi_opmode utmi_opmode select the value of this register to opmode port of HSIC PHY when soft_con_sel=1
3	RW	0x1	i_hsic_utmi_termselect utmi_termselect select the value of this register to termselect port of HSIC PHY when soft_con_sel=1
2	RW	0x1	i_hsic_utmi_suspend_n utmi_suspend_n select the value of this register to ususpend_n port of HSIC PHY when soft_con_sel=1
1	RW	0x1	hsicphy_utmi_dmpulldown utmi_dmpulldown 1:DM pull down resistor enable 0:DM pull down resistor disable

Bit	Attr	Reset Value	Description
0	RW	0x1	hsicphy_utmi_dppulldown utmi_dppulldown 1:DP pull down resistor enable 0:DP pull down resistor disable

GRF_usbphy0_ctrl0
Address: Operational Base + offset (0x04480)
usbphy0_ctrl0

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x850f	usbphy_ctrl0 usbphy_ctrl0 Bit0~15 of usbphy_ctrl register

GRF_usbphy0_ctrl1
Address: Operational Base + offset (0x04484)
usbphy0_ctrl1

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0xe007	usbphy_ctrl1 usbphy_ctrl1 Bit16~31 of usbphy_ctrl register

GRF_usbphy0_ctrl2
Address: Operational Base + offset (0x04488)
usbphy0_ctrl2

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x82e7	usbphy_ctrl2 usbphy_ctrl2 Bit32~47 of usbphy_ctrl register

GRF_usbphy0_ctrl3
Address: Operational Base + offset (0x0448c)
usbphy0_ctrl3

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x02a2	usbphy_ctrl3 usbphy_ctrl3 Bit48~63 of usbphy_ctrl register

GRF_usbphy0_ctrl4
Address: Operational Base + offset (0x04490)
usbphy0_ctrl4

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x5554	usbphy_ctrl4 usbphy_ctrl4 Bit64~79 of usbphy_ctrl register

GRF_usbphy0_ctrl5
Address: Operational Base + offset (0x04494)
usbphy0_ctrl5

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x4555	usbphy_ctrl5 usbphy_ctrl5 Bit80~95 of usbphy_ctrl register

GRF_usbphy0_ctrl6
Address: Operational Base + offset (0x04498)
usbphy0_ctrl6

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x0005	usbphy_ctrl6 usbphy_ctrl6 Bit96~111 of usbphy_ctrl register

GRF_usbphy0_ctrl7
Address: Operational Base + offset (0x0449c)
usbphy0_ctrl7

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x68c8	usbphy_ctrl7 usbphy_ctrl7 Bit112~127 of usbphy_ctrl register

GRF_usbphy0_ctrl8
Address: Operational Base + offset (0x044a0)
usbphy0_ctrl8

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x0000	usbphy_ctrl8 usbphy_ctrl8 Bit128~143 of usbphy_ctrl register

GRF_usbphy0_ctrl9
Address: Operational Base + offset (0x044a4)
usbphy0_ctrl9

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x0000	usbphy_ctrl9 usbphy_ctrl9 Bit144~159 of usbphy_ctrl register

GRF_usbphy0_ctrl10
Address: Operational Base + offset (0x044a8)
usbphy0_ctrl10

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x0000	usbphy_ctrl10 usbphy_ctrl10 Bit160~175 of usbphy_ctrl register

GRF_usbphy0_ctrl11
Address: Operational Base + offset (0x044ac)
usbphy0_ctrl11

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x0000	usbphy_ctrl11 usbphy_ctrl11 Bit176~191 of usbphy_ctrl register

GRF_usbphy0_ctrl12
Address: Operational Base + offset (0x044b0)
usbphy0_ctrl12

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x00a1	usbphy_ctrl12 usbphy_ctrl12 Bit192~207 of usbphy_ctrl register

GRF_usbphy0_ctrl13
Address: Operational Base + offset (0x044b4)
usbphy0_ctrl13

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x850f	usbphy_ctrl13 usbphy_ctrl13 Bit208~223 of usbphy_ctrl register

GRF_usbphy0_ctrl14
Address: Operational Base + offset (0x044b8)
usbphy0_ctrl14

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0xe007	usbphy_ctrl14 usbphy_ctrl14 Bit224~239 of usbphy_ctrl register

GRF_usbphy0_ctrl15
Address: Operational Base + offset (0x044bc)
usbphy0_ctrl15

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x02e7	usbphy_ctrl15 usbphy_ctrl15 Bit240~255 of usbphy_ctrl register

GRF_usbphy0_ctrl16
Address: Operational Base + offset (0x044c0)
usbphy0_ctrl16

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x0200	usbphy_ctrl16 usbphy_ctrl16 Bit256~271 of usbphy_ctrl register

GRF_usbphy0_ctrl17
Address: Operational Base + offset (0x044c4)
usbphy0_ctrl17

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x5554	usbphy_ctrl17 usbphy_ctrl17 Bit272~287 of usbphy_ctrl register

GRF_usbphy0_ctrl18
Address: Operational Base + offset (0x044c8)
usbphy0_ctrl18

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x4555	usbphy_ctrl18 usbphy_ctrl18 Bit288~303 of usbphy_ctrl register

GRF_usbphy0_ctrl19
Address: Operational Base + offset (0x044cc)
usbphy0_ctrl19

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x0005	usbphy_ctrl19 usbphy_ctrl19 Bit304~319 of usbphy_ctrl register

GRF_usbphy0_ctrl20
Address: Operational Base + offset (0x044d0)
usbphy0_ctrl20

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x68c8	usbphy_ctrl20 usbphy_ctrl20 Bit320~335 of usbphy_ctrl register

GRF_usbphy0_ctrl21
Address: Operational Base + offset (0x044d4)
usbphy0_ctrl21

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x0000	usbphy_ctrl21 usbphy_ctrl21 Bit336~351 of usbphy_ctrl register

GRF_usbphy0_ctrl22
Address: Operational Base + offset (0x044d8)
usbphy0_ctrl22

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x0000	usbphy_ctrl22 usbphy_ctrl22 Bit352~367 of usbphy_ctrl register

GRF_usbphy0_ctrl23
Address: Operational Base + offset (0x044dc)
usbphy0_ctrl23

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x0000	usbphy_ctrl23 usbphy_ctrl23 Bit368~383 of usbphy_ctrl register

GRF_usbphy0_ctrl24
Address: Operational Base + offset (0x044e0)
usbphy0_ctrl24

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x0000	usbphy_ctrl24 usbphy_ctrl24 Bit384~399 of usbphy_ctrl register

GRF_usbphy0_ctrl25
Address: Operational Base + offset (0x044e4)
usbphy0_ctrl25

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x0021	usbphy_ctrl25 usbphy_ctrl25 Bit400~415 of usbphy_ctrl register

GRF_usbphy1_ctrl0
Address: Operational Base + offset (0x04500)
usbphy1_ctrl0

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x850f	usbphy_ctrl0 usbphy_ctrl0 Bit0~15 of usbphy_ctrl register

GRF_usbphy1_ctrl1
Address: Operational Base + offset (0x04504)
usbphy1_ctrl1

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0xe007	usbphy_ctrl1 usbphy_ctrl1 Bit16~31 of usbphy_ctrl register

GRF_usbphy1_ctrl2
Address: Operational Base + offset (0x04508)
usbphy1_ctrl2

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x82e7	usbphy_ctrl2 usbphy_ctrl2 Bit32~47 of usbphy_ctrl register

GRF_usbphy1_ctrl3
Address: Operational Base + offset (0x0450c)
usbphy1_ctrl3

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x02a2	usbphy_ctrl3 usbphy_ctrl3 Bit48~63 of usbphy_ctrl register

GRF_usbphy1_ctrl4
Address: Operational Base + offset (0x04510)
usbphy1_ctrl4

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x5554	usbphy_ctrl4 usbphy_ctrl4 Bit64~79 of usbphy_ctrl register

GRF_usbphy1_ctrl5
Address: Operational Base + offset (0x04514)
usbphy1_ctrl5

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x4555	usbphy_ctrl5 usbphy_ctrl5 Bit80~95 of usbphy_ctrl register

GRF_usbphy1_ctrl6
Address: Operational Base + offset (0x04518)
usbphy1_ctrl6

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x0005	usbphy_ctrl6 usbphy_ctrl6 Bit96~111 of usbphy_ctrl register

GRF_usbphy1_ctrl7
Address: Operational Base + offset (0x0451c)
usbphy1_ctrl7

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x68c8	usbphy_ctrl7 usbphy_ctrl7 Bit112~127 of usbphy_ctrl register

GRF_usbphy1_ctrl8
Address: Operational Base + offset (0x04520)
usbphy1_ctrl8

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x0000	usbphy_ctrl8 usbphy_ctrl8 Bit128~143 of usbphy_ctrl register

GRF_usbphy1_ctrl9
Address: Operational Base + offset (0x04524)
usbphy1_ctrl9

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x0000	usbphy_ctrl9 usbphy_ctrl9 Bit144~159 of usbphy_ctrl register

GRF_usbphy1_ctrl10
Address: Operational Base + offset (0x04528)
usbphy1_ctrl10

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x0000	usbphy_ctrl10 usbphy_ctrl10 Bit160~175 of usbphy_ctrl register

GRF_usbphy1_ctrl11
Address: Operational Base + offset (0x0452c)
usbphy1_ctrl11

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x0000	usbphy_ctrl11 usbphy_ctrl11 Bit176~191 of usbphy_ctrl register

GRF_usbphy1_ctrl12
Address: Operational Base + offset (0x04530)
usbphy1_ctrl12

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x00a1	usbphy_ctrl12 usbphy_ctrl12 Bit192~207 of usbphy_ctrl register

GRF_usbphy1_ctrl13
Address: Operational Base + offset (0x04534)
usbphy1_ctrl13

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x850f	usbphy_ctrl13 usbphy_ctrl13 Bit208~223 of usbphy_ctrl register

GRF_usbphy1_ctrl14
Address: Operational Base + offset (0x04538)
usbphy1_ctrl14

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0xe007	usbphy_ctrl14 usbphy_ctrl14 Bit224~239 of usbphy_ctrl register

GRF_usbphy1_ctrl15
Address: Operational Base + offset (0x0453c)
usbphy1_ctrl15

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x02e7	usbphy_ctrl15 usbphy_ctrl15 Bit240~255 of usbphy_ctrl register

GRF_usbphy1_ctrl16
Address: Operational Base + offset (0x04540)
usbphy1_ctrl16

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x0200	usbphy_ctrl16 usbphy_ctrl16 Bit256~271 of usbphy_ctrl register

GRF_usbphy1_ctrl17
Address: Operational Base + offset (0x04544)
usbphy1_ctrl17

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x5554	usbphy_ctrl17 usbphy_ctrl17 Bit272~287 of usbphy_ctrl register

GRF_usbphy1_ctrl18
Address: Operational Base + offset (0x04548)
usbphy1_ctrl18

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x4555	usbphy_ctrl18 usbphy_ctrl18 Bit288~303 of usbphy_ctrl register

GRF_usbphy1_ctrl19
Address: Operational Base + offset (0x0454c)
usbphy1_ctrl19

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x0005	usbphy_ctrl19 usbphy_ctrl19 Bit304~319 of usbphy_ctrl register

GRF_usbphy1_ctrl20
Address: Operational Base + offset (0x04550)
usbphy1_ctrl20

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x68c8	usbphy_ctrl20 usbphy_ctrl20 Bit320~335 of usbphy_ctrl register

GRF_usbphy1_ctrl21
Address: Operational Base + offset (0x04554)
usbphy1_ctrl21

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x0000	usbphy_ctrl21 usbphy_ctrl21 Bit336~351 of usbphy_ctrl register

GRF_usbphy1_ctrl22
Address: Operational Base + offset (0x04558)
usbphy1_ctrl22

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x0000	usbphy_ctrl22 usbphy_ctrl22 Bit352~367 of usbphy_ctrl register

GRF_usbphy1_ctrl23
Address: Operational Base + offset (0x0455c)
usbphy1_ctrl23

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x0000	usbphy_ctrl23 usbphy_ctrl23 Bit368~383 of usbphy_ctrl register

GRF_usbphy1_ctrl24
Address: Operational Base + offset (0x04560)
usbphy1_ctrl24

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x0000	usbphy_ctrl24 usbphy_ctrl24 Bit384~399 of usbphy_ctrl register

GRF_usbphy1_ctrl25
Address: Operational Base + offset (0x04564)
usbphy1_ctrl25

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x0021	usbphy_ctrl25 usbphy_ctrl25 Bit400~415 of usbphy_ctrl register

GRF_HDCP22_PERF_CON0

Address: Operational Base + offset (0x06000)

hdcp performance monitor control register

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:12	RO	0x0	reserved
11:8	RW	0x0	hdcp22_sw_rd_latency_id Axi read channel id for latency AXI_PERFormance test
7	RO	0x0	reserved

Bit	Attr	Reset Value	Description
6:5	RW	0x0	hdcp22_sw_ddr_align_type 0: 16-Byte align 1: 32-Byte align 2: 64-Byte align 3: 128-Byte align
4	RW	0x0	hdcp22_sw_aw_cnt_id_type 0: count all write channels 1: count sw_aw_count_id write channel only
3	RW	0x0	hdcp22_sw_ar_cnt_id_type axi_perf counter id control 0: count all read channel id 1: count sw_ar_count_id read channel only
2	RW	0x0	hdcp22_sw_axi_cnt_type axi_perf counter type 0: axi transfer test 1: ddr align transfer test
1	RW	0x0	hdcp22_sw_axi_perf_clr axi_perf clear bit 0: disable 1: enable
0	RW	0x0	hdcp22_sw_axi_perf_work axi_perf enable bit 0: disable 1: enable

GRF_HDCP22_PERF_CON1
Address: Operational Base + offset (0x06004)
hdcp performance monitor control register

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:12	RO	0x0	reserved

Bit	Attr	Reset Value	Description
11:0	RW	0x000	hdcp22_sw_rd_latency_thr Axi Read latency threshold

GRF_HDCP22_PERF_CON2
Address: Operational Base + offset (0x06008)
hdcp performance monitor control register

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:8	RO	0x0	reserved
7:4	RW	0x0	hdcp22_sw_aw_count_id When sw_aw_cnt_id_type=1, only count the id designated by sw_aw_count_id
3:0	RW	0x0	hdcp22_sw_ar_count_id When sw_ar_cnt_id_type=1, only count the id designated by sw_ar_count_id

GRF_HDCP22_PERF_RD_MAX_LATENCY_NUM
Address: Operational Base + offset (0x0600c)
hdcp performance monitor status register

Bit	Attr	Reset Value	Description
31:13	RO	0x0	reserved
12:0	RW	0x0000	rd_max_latency_r axi read max latency output

GRF_HDCP22_PERF_RD_LATENCY_SAMP_NUM
Address: Operational Base + offset (0x06010)
hdcp performance monitor status register

Bit	Attr	Reset Value	Description
31:27	RO	0x0	reserved
26:0	RW	0x0000000	rd_latency_samp_r AXI read latency total sample number

GRF_HDCP22_PERF_RD_LATENCY_ACC_NUM
Address: Operational Base + offset (0x06014)
hdcp performance monitor status register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rd_latency_acc_cnt_r AXI read latency (>sw_rd_latency_thr) total number

GRF_HDCP22_PERF_RD_AXI_TOTAL_BYTE
Address: Operational Base + offset (0x06018)
hdcp performance monitor status register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rd_axi_total_byte AXI active total read bytes/ddr align read bytes

GRF_HDCP22_PERF_WR_AXI_TOTAL_BYTE
Address: Operational Base + offset (0x0601c)
hdcp performance monitor status register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	wr_axi_total_byte AXI active total write bytes/ddr align write bytes

GRF_HDCP22_PERF_WORKING_CNT
Address: Operational Base + offset (0x06020)
hdcp performance monitor status register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	working_cnt_r working counter

GRF_SOC_CON9
Address: Operational Base + offset (0x06224)
SoC control register 9

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15	RW	0x0	dsi0_dpiupdatecfg DSI host0 dpiupdatecfg bit control
14	RW	0x0	dsi0_dpishutdn DSI0 dpishutdn bit control
13	RW	0x0	dsi0_dpicolorm DSI0 dpicolorm bit control
12	RW	0x0	dp_lcdc_sel dp lcdc select 1'b0: vop big 1'b1: vop little
11	RW	0x0	dphy_rx1_clk_inv_sel dphy rx1 clock inveter select bit
10	RW	0x0	dphy_rx0_clk_inv_sel dphy rx0 clock inveter select bit
9	RW	0x0	disable_isp1 isp1 disable control
8	RW	0x0	disable_isp0 isp0 disable control
7:4	RO	0x0	reserved
3:0	RW	0x0	dphy_rx0_turnrequest dphy rx0 runrequest port control

GRF_SOC_CON20
Address: Operational Base + offset (0x06250)
SoC control register 20

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15	RO	0x0	reserved
14	RW	0x0	hdcp_i2c_force_sda hdcp_i2c_force_sda bit control
13	RW	0x0	hdcp_i2c_force_scl hdcp_i2c_force_scl control
12	RW	0x0	grf_vop_rgb_dclk_rev_sel dclk phase selct 0: 0 degree 1: 180 degree
11	RW	0x0	grf_con_rgb_lcdc_sel vop select 1'b0: vop big 1'b1: vop little
10	RO	0x0	reserved
9	RW	0x1	pclkin_dvp_rev_sel pclkin dvp clock select 0: not invet phase 1: invert phase
8	RW	0x0	edp_video_bist_en edp video bist enable 1: enable 0: disable
7	RW	0x0	vop_finish_sel vop finish select 1'b0: vop big 1'b1: vop little

Bit	Attr	Reset Value	Description
6	RW	0x1	hdmi_lcdc_sel hdmi lcdc select 1'b0: vop big 1'b1: vop little
5	RW	0x0	edp_lcdc_sel edp lcdc select 1'b0: vop big 1'b1: vop little
4	RW	0x0	dsi1_lcdc_sel dsi1 lcdc select 1'b0: vop big 1'b1: vop little
3	RW	0x1	dsi1_dpiupdatecfg dsi1 dpiupdatecfg bit control
2	RW	0x0	dsi1_dpishutdn dsi1 dpishutdn bit control
1	RW	0x0	dsi1_dpicolorm dsi1 dpicolorm bit control
0	RW	0x1	dsi0_lcdc_sel dsi0 vol select bit 1'b0: vop big 1'b1: vop little

GRF_SOC_CON21
Address: Operational Base + offset (0x06254)
SoC control register 21

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:12	RW	0x0	dphy_rx0_turndisable dphy_rx0_turndisable bit control

Bit	Attr	Reset Value	Description
11:8	RW	0x2	dphy_rx0_forcetxstopmode dphy_rx0_forcetxstopmode bit control
7:4	RW	0xc	dphy_rx0_forcerxmode dphy_rx0_forcerxmode bit control
3:0	RW	0xb	dphy_rx0_enable dphy_rx0_enable bit control

GRF_SOC_CON22
Address: Operational Base + offset (0x06258)
SoC control register 22

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:12	RW	0x1	dphy_tx0_turnrequest dphy_tx0_turnrequest bit control
11:8	RW	0x0	dphy_tx0_turndisable dphy_tx0_turndisable bit control
7:4	RW	0xc	dphy_tx0_forcetxstopmode dphy_tx0_forcetxstopmode bit control
3:0	RW	0xb	dphy_tx0_forcerxmode dphy_tx0_forcerxmode bit control

GRF_SOC_CON23
Address: Operational Base + offset (0x0625c)
SoC control register 23

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:12	RW	0x0	dphy_tx1rx1_turndisable dphy_tx1rx1_turndisable bit control
11:8	RW	0x0	dphy_tx1rx1_forcetxstopmode dphy_tx1rx1_forcetxstopmode bit control
7:4	RW	0x2	dphy_tx1rx1_forcerxmode dphy_tx1rx1_forcerxmode bit control
3:0	RW	0x1	dphy_tx1rx1_enable dphy_tx1rx1_enable bit control

GRF_SOC_CON24

Address: Operational Base + offset (0x06260)

SoC control register 24

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;

Bit	Attr	Reset Value	Description
15:14	RW	0x0	vopl_dsi_ite_sel vopl_dsi_ite_sel bit control 0: mipi_dsi0_edpite 1: mipi_dsi1_edpite 2: 0 3: 1
13:12	RW	0x3	vopb_dsi_ite_sel vopb_dsi_ite_sel bit control 0: mipi_dsi0_edpite 1: mipi_dsi1_edpite 2: 0 3: 1
11:10	RW	0x2	vopl_dsi_halt_sel vopl_dsi_halt_sel bit control 0: mipi_dsi0_edpihalt 1: mipi_dsi0_edpihalt 2: low 3: high
9:8	RW	0x1	vopb_dsi_halt_sel vopb_dsi_halt_sel bit control 0: mipi_dsi0_edpihalt 1: mipi_dsi0_edpihalt 2: low 3: high
7	RW	0x1	dphy_tx1rx1_masterslavez dphy_tx1rx1_masterslavez bit control
6	RW	0x1	dphy_tx1rx1_enableclk dphy_tx1rx1_enableclk bit control
5	RW	0x1	dphy_tx1rx1_basedir dphy_tx1rx1_basedir bit control
4	RW	0x1	dphy_rx1_src_sel dphy_rx1_src_sel bit control
3:0	RW	0x0	dphy_tx1tx1_turnrequest dphy_tx1tx1_turnrequest bit control

GRF_SOC_CON25
Address: Operational Base + offset (0x06264)
SoC control register 25

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:12	RW	0xd	edp_tx_bscan_data edp_tx_bscan_data bit control
11	RW	0x0	edp_ref_clk_sel edp_ref_clk_sel bit control
10	RW	0x1	dphy_rx0_tsetclr dphy_rx0_tsetclr bit control
9	RW	0x0	dphy_rx0_tsetclk dphy_rx0_tsetclk bit control
8	RW	0x0	dphy_rx0_tseten dphy_rx0_tseten bit control
7:0	RW	0x5b	dphy_rx0_tsetdin dphy_rx0_tsetdin bit control

GRF_SOC_CON26
Address: Operational Base + offset (0x06268)
SoC control register 26

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:14	RO	0x0	reserved
13:12	RW	0x0	dptx_hpd_sel dptx_hpd_sel bit control
11:9	RO	0x0	reserved
8	RW	0x1	force_dp_xt_ocdhaltonreset force_dp_xt_ocdhaltonreset bit control
7:4	RW	0x1	dptx_lane_sel dptx_lane_sel bit control
3	RW	0x0	uphy_dp_sel uphy_dp_sel bit control
2	RW	0x0	hdcp22_src_sel hdcp22_src_sel bit control
1	RO	0x0	reserved
0	RW	0x0	edp_tx_bscan_en edp_tx_bscan_en bit control

GRF_GPU_PERF_CON0
Address: Operational Base + offset (0x08000)
gpu performance monitor control register

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:14	RO	0x0	reserved
13:8	RW	0x00	gpu_sw_rd_latency_id Axi read channel id for latency AXI_PERFormance test
7	RO	0x0	reserved
6:5	RW	0x0	gpu_sw_ddr_align_type 0: 16-Byte align 1: 32-Byte align 2: 64-Byte align 3: 128-Byte align
4	RW	0x0	gpu_sw_aw_cnt_id_type axi_perf counter id control 0: count all write channel id 1: count sw_ar_count_id write channel only
3	RW	0x0	gpu_sw_ar_cnt_id_type axi_perf counter id control 0: count all read channel id 1: count sw_ar_count_id read channel only
2	RW	0x0	gpu_sw_axi_cnt_type axi_perf counter type 0: axi transfer test 1: ddr align transfer test
1	RW	0x0	gpu_sw_axi_perf_clr axi_perf clear bit 0: disable 1: enable
0	RW	0x0	gpu_sw_axi_perf_work axi_perf enable bit 0: disable 1: enable

GRF_GPU_PERF_CON1
Address: Operational Base + offset (0x08004)
gpu performance monitor control register

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:12	RO	0x0	reserved
11:0	RW	0x000	gpu_sw_rd_latency_thr Axi read channel id for latency AXI_PERFormance test

GRF_GPU_PERF_CON2
Address: Operational Base + offset (0x08008)
gpu performance monitor control register

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:13	RO	0x0	reserved

Bit	Attr	Reset Value	Description
12:8	RW	0x00	gpu_sw_aw_count_id When sw_aw_cnt_id_type=1, only count the id designated by sw_aw_count_id
7:5	RO	0x0	reserved
4:0	RW	0x00	gpu_sw_ar_count_id When sw_ar_cnt_id_type=1, only count the id designated by sw_ar_count_id

GRF_GPU_PERF_RD_MAX_LATENCY_NUM
Address: Operational Base + offset (0x0800c)
gpu performance monitor status register

Bit	Attr	Reset Value	Description
31:13	RO	0x0	reserved
12:0	RW	0x0000	rd_max_latency_r axi read max latency output

GRF_GPU_PERF_RD_LATENCY_SAMP_NUM
Address: Operational Base + offset (0x08010)
gpu performance monitor status register

Bit	Attr	Reset Value	Description
31:27	RO	0x0	reserved
26:0	RW	0x0000000	rd_latency_samp_r AXI read latency total sample number

GRF_GPU_PERF_RD_LATENCY_ACC_NUM
Address: Operational Base + offset (0x08014)
gpu performance monitor status register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rd_latency_acc_cnt_r AXI read latency (>sw_rd_latency_thr) total number

GRF_GPU_PERF_RD_AXI_TOTAL_BYTE
Address: Operational Base + offset (0x08018)
gpu performance monitor status register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rd_axi_total_byte AXI active total read bytes/ddr align read bytes

GRF_GPU_PERF_WR_AXI_TOTAL_BYTE
Address: Operational Base + offset (0x0801c)
gpu performance monitor status register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	wr_axi_total_byte AXI active total write bytes/ddr align write bytes

GRF_GPU_PERF_WORKING_CNT
Address: Operational Base + offset (0x08020)
gpu performance monitor status register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	working_cnt_r working counter

GRF_CPU_CON0
Address: Operational Base + offset (0x0a000)
cpu control register 0

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:12	RW	0x0	cfgte_pd_core_l pd_core_l cpu cfgte bit control 0: disable 1: enable
11:8	RW	0x0	cfgend_pd_core_l pd_core_l cpu cfgend bit control 0: disable 1: enable
7	RW	0x0	l2rstdisable_pd_core_l pd_core_l cpu l2rstdisable bit control 0: disable 1: enable

Bit	Attr	Reset Value	Description
6	RW	0x0	dbgl1rstdisable_pd_core_l pd_core_l cpu dbgl1rstdisable bit control 0: disable 1: enable
5	RO	0x0	reserved
4	RW	0x0	clrexmonreq_pd_core_l pd_core_l cpu clrexmonreq bit control 0: disable 1: enable
3	RW	0x1	sysbardisable_pd_core_l pd_core_l cpu sysbardisable bit control 0: disable 1: enable
2	RW	0x0	broadcastcachemaint_pd_core_l pd_core_l cpu broadcastcachemaint bit control 0: disable 1: enable
1	RW	0x1	broadcastouter_pd_core_l pd_core_l cpu broadcastouter bit control 0: disable 1: enable
0	RW	0x1	broadcastinner_pd_core_l pd_core_l cpu broadcastinner bit control 1: enable 0: disable

GRF_CPU_CON1
Address: Operational Base + offset (0x0a004)
cpu control register 1

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:12	RW	0xf	gic_active_core_l pd_core_l gic_active bit control
11:8	RW	0x0	clusteridaff1_pd_core_l pd_core_l clusteridaff1 bit control
7:4	RW	0x0	arqos_pd_core_l pd_core_l arqos bit control
3:0	RW	0x0	awqos_pd_core_l pd_core_l awqos bit control

GRF_CPU_CON2

Address: Operational Base + offset (0x0a008)

cpu control register 2

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15	RO	0x0	reserved

Bit	Attr	Reset Value	Description
14	RW	0x0	gic_axim_err_ack gic axi master error acknowledges 0: disable 1: enable
13:12	RW	0x0	cfgte_pd_core_b pd_core_b cpu cfgte bit control 0: disable 1: enable
11:10	RO	0x0	reserved
9:8	RW	0x0	cfgend_pd_core_b pd_core_b cpu cfgend bit control 0: disable 1: enable
7	RW	0x0	l2rstdisable_pd_core_b pd_core_b cpu l2rstdisable bit control 0: disable 1: enable
6	RW	0x0	dbgl1rstdisable_pd_core_b pd_core_b cpu dbgl1rstdisable bit control 0: disable 1: enable
5	RO	0x0	reserved
4	RW	0x0	clrexmonreq_pd_core_b pd_core_b cpu clrexmonreq bit control 0: disable 1: enable
3	RW	0x1	sysbardisable_pd_core_b pd_core_b cpu sysbardisable bit control 0: disable 1: enable
2	RW	0x0	broadcastcachemaint_pd_core_b pd_core_b cpu broadcastcachemaint bit control 0: disable 1: enable
1	RW	0x1	broadcastouter_pd_core_b pd_core_b cpu broadcastouter bit control 0: disable 1: enable
0	RW	0x1	broadcastinner_pd_core_b pd_core_b cpu broadcastinner bit control 1: enable 0: disable

GRF_CPU_CON3

Address: Operational Base + offset (0x0a00c)
cpu control register 3

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:14	RO	0x0	reserved
13:12	RW	0x3	gic_active_core_b pd_core_b gic_active bit control
11:10	RO	0x0	reserved
9:8	RW	0x1	clusteridaff1_pd_core_b pd_core_b clusteridaff1 bit control
7:6	RO	0x0	reserved
5:4	RW	0x1	arqos_pd_core_b pd_core_b arqos bit control
3:2	RO	0x0	reserved
1:0	RW	0x0	awqos_pd_core_b pd_core_b awqos bit control

GRF_CPU_STATUS0
Address: Operational Base + offset (0x0a080)
cpu status register 0

Bit	Attr	Reset Value	Description
31:24	RW	0x00	wrmemattr_pd_core_b wrmemattr of pd_core_b status
23:16	RW	0x00	rdmemattr_pd_core_b rdmemattr of pd_core_b status
15:8	RW	0x00	rdmemattr_pd_core_l rdmemattr of pd_core_l status
7:0	RW	0x00	wrmemattr_pd_core_l wrmemattr of pd_core_l status

GRF_CPU_STATUS1
Address: Operational Base + offset (0x0a084)

cpu status register 1

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27	RW	0x0	clremonack_pd_core_b the status of clremonack_pd_core_b
26	RW	0x0	clremonack_pd_core_l the status of clremonack_pd_core_l
25	RW	0x0	standbywfil2_pd_core_b standbywfil2 of pd_core_b status bit
24	RW	0x0	standbywfil2_pd_core_l standbywfil2 of pd_core_l status bit
23:22	RO	0x0	reserved
21:20	RW	0x0	smpen_pd_core_b status of smpen_pd_core_b
19:16	RW	0x0	smpen_pd_core_l status of smpen_pd_core_l
15:14	RO	0x0	reserved
13:12	RW	0x0	standbywfe_pd_core_b standbywfe of pd_core_b status bit
11:10	RO	0x0	reserved
9:8	RW	0x0	standbywfi_pd_core_b standbywfi of pd_core_b status bit
7:4	RW	0x0	standbywfe_pd_core_l standbywfe of pd_core_l status bit
3:0	RW	0x0	standbywfi_pd_core_l standbywfi of pd_core_l status bit

GRF_CPU_STATUS2

Address: Operational Base + offset (0x0a088)

cpu status register 2

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	cci_event_bus the status of cci_event_bus[31:0]

GRF_CPU_STATUS3

Address: Operational Base + offset (0x0a08c)

cpu status register 3

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	cci_event_bus the status of cci_event_bus[63:32]

GRF_CPU_STATUS4

Address: Operational Base + offset (0x0a090)

cpu status register 4

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved

Bit	Attr	Reset Value	Description
29:0	RW	0x00000000	cci_event_bus the status of cci_event_bus[93:64]

GRF_CPU_STATUS5
Address: Operational Base + offset (0x0a094)
cpu status register 5

Bit	Attr	Reset Value	Description
31:10	RO	0x0	reserved
9	RW	0x0	gic_axim_err gic_axim_err status bit
8	RW	0x0	gic_ecc_fatal gic_ecc_fatal status bit
7:0	RW	0x00	cci_nevntcntoverflow cci_nevntcntoverflow status bit

GRF_A53_PERF_CON0
Address: Operational Base + offset (0x0a100)
a53 performance monitor control register

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:14	RO	0x0	reserved
13:8	RW	0x00	a53_sw_rd_latency_id Axi read channel id for latency AXI_PERFormance test
7	RO	0x0	reserved
6:5	RW	0x0	a53_sw_ddr_align_type 0: 16-Byte align 1: 32-Byte align 2: 64-Byte align 3: 128-Byte align

Bit	Attr	Reset Value	Description
4	RW	0x0	a53_sw_aw_cnt_id_type axi_perf counter id control 0: count all write channel id 1: count sw_ar_count_id write channel only
3	RW	0x0	a53_sw_ar_cnt_id_type axi_perf counter id control 0: count all read channel id 1: count sw_ar_count_id read channel only
2	RW	0x0	a53_sw_axi_cnt_type axi_perf counter type 0: axi transfer test 1: ddr align transfer test
1	RW	0x0	a53_sw_axi_perf_clr axi_perf clear bit 0: disable 1: enable
0	RW	0x0	a53_sw_axi_perf_work a53 performance monitor control register axi_perf enable bit 0: disable 1: enable

GRF_A53_PERF_CON1
Address: Operational Base + offset (0x0a104)
a53 performance monitor control register

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:12	RO	0x0	reserved
11:0	RW	0x000	a53_sw_rd_latency_thr Axi read channel id for latency AXI_PERFormance test

GRF_A53_PERF_CON2
Address: Operational Base + offset (0x0a108)
a53 performance monitor control register

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:13	RO	0x0	reserved
12:8	RW	0x00	a53_sw_aw_count_id When sw_aw_cnt_id_type=1, only count the id designated by sw_aw_count_id
7:6	RO	0x0	reserved
5:0	RW	0x00	a53_sw_ar_count_id When sw_ar_cnt_id_type=1, only count the id designated by sw_ar_count_id

GRF_A53_PERF_CON3
Address: Operational Base + offset (0x0a10c)
a53 performance monitor control register

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:9	RW	0x00	mon_id mon_id bit control
8:2	RW	0x00	mon_id_bmsk mon_id_bmsk bit control
1	RW	0x0	mon_id_type mon_id_type bit control
0	RW	0x0	mon_id_msk mon_id_msk bit control

GRF_A53_PERF_RD_MON_ST
Address: Operational Base + offset (0x0a110)
performance monitor read start address

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rd_start_addr monitor read start address

GRF_A53_PERF_RD_MON_END
Address: Operational Base + offset (0x0a114)
performance monitor end address

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rd_end_addr monitor read end address

GRF_A53_PERF_WR_MON_ST
Address: Operational Base + offset (0x0a118)
performance write monitor start address

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	wr_start_addr monitor write start address

GRF_A53_PERF_WR_MON_END
Address: Operational Base + offset (0x0a11c)
performance monitor write end address

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	wr_end_addr monitor write end address

GRF_A53_PERF_RD_MAX_LATENCY_NUM
Address: Operational Base + offset (0x0a120)
a53 performance monitor status register

Bit	Attr	Reset Value	Description
31:13	RO	0x0	reserved
12:0	RW	0x0000	rd_max_latency_r axi read max latency output

GRF_A53_PERF_RD_LATENCY_SAMP_NUM
Address: Operational Base + offset (0x0a124)
a53 performance monitor status register

Bit	Attr	Reset Value	Description
31:27	RO	0x0	reserved
26:0	RW	0x0000000	rd_latency_samp_r AXI read latency total sample number

GRF_A53_PERF_RD_LATENCY_ACC_NUM
Address: Operational Base + offset (0x0a128)
a53 performance monitor status register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rd_latency_acc_cnt_r AXI read latency (>sw_rd_latency_thr) total number

GRF_A53_PERF_RD_AXI_TOTAL_BYTE
Address: Operational Base + offset (0x0a12c)
a53 performance monitor status register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rd_axi_total_byte AXI active total read bytes/ddr align read bytes

GRF_A53_PERF_WR_AXI_TOTAL_BYTE
Address: Operational Base + offset (0x0a130)
a53 performance monitor status register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	wr_axi_total_byte AXI active total write bytes/ddr align write bytes

GRF_A53_PERF_WORKING_CNT
Address: Operational Base + offset (0x0a134)
a53 performance monitor status register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	working_cnt_r working counter

GRF_A53_PERF_INT_STATUS
Address: Operational Base + offset (0x0a138)
a53 performance monitor status register

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	int_status interrupt status bit

GRF_A72_PERF_CON0
Address: Operational Base + offset (0x0a200)
a72 performance monitor control register

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15	RO	0x0	reserved
14:8	RW	0x00	a72_sw_rd_latency_id Axi read channel id for latency AXI_PERFormance test
7	RO	0x0	reserved

Bit	Attr	Reset Value	Description
6:5	RW	0x0	a72_sw_ddr_align_type 0: 16-Byte align 1: 32-Byte align 2: 64-Byte align 3: 128-Byte align
4	RW	0x0	a72_sw_aw_cnt_id_type axi_perf counter id control 0: count all write channel id 1: count sw_ar_count_id write channel only
3	RW	0x0	a72_sw_ar_cnt_id_type axi_perf counter id control 0: count all read channel id 1: count sw_ar_count_id read channel only
2	RW	0x0	a72_sw_axi_cnt_type axi_perf counter type 0: axi transfer test 1: ddr align transfer test
1	RW	0x0	a72_sw_axi_perf_clr axi_perf clear bit 0: disable 1: enable
0	RW	0x0	a72_sw_axi_perf_work axi_perf enable bit 0: disable 1: enable

GRF_A72_PERF_CON1

Address: Operational Base + offset (0x0a204)

a72 performance monitor control register

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;

Bit	Attr	Reset Value	Description
15:12	RO	0x0	reserved
11:0	RW	0x000	a72_sw_rd_latency_thr Axi read channel id for latency AXI_PERFormance test

GRF_A72_PERF_CON2
Address: Operational Base + offset (0x0a208)
a72 performance monitor control register

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15	RO	0x0	reserved
14:8	RW	0x00	a72_sw_aw_count_id When sw_aw_cnt_id_type=1, only count the id designated by sw_aw_count_id
7:6	RO	0x0	reserved
5:0	RW	0x00	a72_sw_ar_count_id When sw_ar_cnt_id_type=1, only count the id designated by sw_ar_count_id

GRF_A72_PERF_CON3
Address: Operational Base + offset (0x0a20c)
a72 performance monitor control register

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:9	RW	0x00	mon_id mon_id bit control
8:2	RW	0x00	mon_id_bmsk mon_id_bmsk bit control
1	RW	0x0	mon_id_type mon_id_type bit control
0	RW	0x0	mon_id_msk mon_id_msk bit control

GRF_A72_PERF_RD_MON_ST
Address: Operational Base + offset (0x0a210)
performance monitor read start address

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rd_start_addr monitor read start address

GRF_A72_PERF_RD_MON_END
Address: Operational Base + offset (0x0a214)
performance monitor end address

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rd_end_addr monitor read end address

GRF_A72_PERF_WR_MON_ST
Address: Operational Base + offset (0x0a218)
performance write monitor start address

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	wr_start_addr monitor write start address

GRF_A72_PERF_WR_MON_END
Address: Operational Base + offset (0x0a21c)
performance monitor write end address

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	wr_end_addr monitor write end address

GRF_A72_PERF_RD_MAX_LATENCY_NUM
Address: Operational Base + offset (0x0a220)
a72 performance monitor status register

Bit	Attr	Reset Value	Description
31:13	RO	0x0	reserved
12:0	RW	0x0000	rd_max_latency_r axi read max latency output

GRF_A72_PERF_RD_LATENCY_SAMP_NUM
Address: Operational Base + offset (0x0a224)
a72 performance monitor status register

Bit	Attr	Reset Value	Description
31:27	RO	0x0	reserved
26:0	RW	0x0000000	rd_latency_samp_r AXI read latency total sample number

GRF_A72_PERF_RD_LATENCY_ACC_NUM
Address: Operational Base + offset (0x0a228)
a72 performance monitor status register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rd_latency_acc_cnt_r AXI read latency (>sw_rd_latency_thr) total number

GRF_A72_PERF_RD_AXI_TOTAL_BYTE
Address: Operational Base + offset (0x0a22c)
a72 performance monitor status register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rd_axi_total_byte AXI active total read bytes/ddr align read bytes

GRF_A72_PERF_WR_AXI_TOTAL_BYTE
Address: Operational Base + offset (0x0a230)
a72 performance monitor status register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	wr_axi_total_byte AXI active total write bytes/ddr align write bytes

GRF_A72_PERF_WORKING_CNT
Address: Operational Base + offset (0x0a234)
a72 performance monitor status register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	working_cnt_r working counter

GRF_A72_PERF_INT_STATUS
Address: Operational Base + offset (0x0a238)
a72 performance monitor status register

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	int_status interrupt status bit

GRF_GMAC_PERF_CON0
Address: Operational Base + offset (0x0c000)
gmac performance monitor control register

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:13	RO	0x0	reserved
12:8	RW	0x00	gmac_sw_rd_latency_id Axi read channel id for latency AXI_PERFormance test
7	RO	0x0	reserved

Bit	Attr	Reset Value	Description
6:5	RW	0x0	gmac_sw_ddr_align_type 0: 16-Byte align 1: 32-Byte align 2: 64-Byte align 3: 128-Byte align
4	RW	0x0	gmac_sw_aw_cnt_id_type 0: count all write channels 1: count sw_aw_count_id write channel only
3	RW	0x0	gmac_sw_ar_cnt_id_type axi_perf counter id control 0: count all read channel id 1: count sw_ar_count_id read channel only
2	RW	0x0	gmac_sw_axi_cnt_type axi_perf counter type 0: axi transfer test 1: ddr align transfer test
1	RW	0x0	gmac_sw_axi_perf_clr axi_perf clear bit 0: disable 1: enable
0	RW	0x0	gmac_sw_axi_perf_work axi_perf enable bit 0: disable 1: enable

GRF_GMAC_PERF_CON1
Address: Operational Base + offset (0x0c004)
gmac performance monitor control register

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:12	RO	0x0	reserved

Bit	Attr	Reset Value	Description
11:0	RW	0x000	gmac_sw_rd_latency_thr Axi read channel id for latency AXI_PERFormance test

GRF_GMAC_PERF_CON2
Address: Operational Base + offset (0x0c008)
gmac performance monitor control register

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:12	RO	0x0	reserved
11:8	RW	0x0	gmac_sw_aw_count_id When sw_aw_cnt_id_type=1, only count the id designated by sw_aw_count_id
7:4	RO	0x0	reserved
3:0	RW	0x0	gmac_sw_ar_count_id When sw_ar_cnt_id_type=1, only count the id designated by sw_ar_count_id

GRF_GMAC_PERF_RD_MAX_LATENCY_NUM
Address: Operational Base + offset (0x0c00c)
gmac performance monitor status register

Bit	Attr	Reset Value	Description
31:13	RO	0x0	reserved
12:0	RW	0x0000	rd_max_latency_r axi read max latency oaxi read max latency outpututput

GRF_GMAC_PERF_RD_LATENCY_SAMP_NUM
Address: Operational Base + offset (0x0c010)
gmac performance monitor status register

Bit	Attr	Reset Value	Description
31:27	RO	0x0	reserved
26:0	RW	0x0000000	rd_latency_samp_r AXI read latency total sample number

GRF_GMAC_PERF_RD_LATENCY_ACC_NUM
Address: Operational Base + offset (0x0c014)
gmac performance monitor status register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rd_latency_acc_cnt_r AXI read latency (>sw_rd_latency_thr) total number

GRF_GMAC_PERF_RD_AXI_TOTAL_BYTE
Address: Operational Base + offset (0x0c018)
gmac performance monitor status register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rd_axi_total_byte AXI active total read bytes/ddr align read bytes

GRF_GMAC_PERF_WR_AXI_TOTAL_BYTE
Address: Operational Base + offset (0x0c01c)
gmac performance monitor status register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	wr_axi_total_byte AXI active total write bytes/ddr align write bytes

GRF_GMAC_PERF_WORKING_CNT
Address: Operational Base + offset (0x0c020)
gmac performance monitor status register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	working_cnt_r working counter

GRF_SOC_CON5
Address: Operational Base + offset (0x0c214)
SoC control register 5

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:12	RO	0x0	reserved
11:9	RW	0x0	gmac_phy_intf_sel PHY interface select 3'b001: RGMII 3'b100: RMII All others: Reserved
8	RW	0x0	gmac_flowctrl GMAC transmit flow control When set high, instructs the GMAC to transmit PAUSE Control frames in Full-duplex mode. In Half-duplex mode, the GMAC enables the Back-pressure function until this signal is made low again
7	RW	0x0	gmac_speed MAC speed 1'b1: 100-Mbps 1'b0: 10-Mbps
6	RW	0x0	rmii_mode RMII mode selection 1'b1: RMII mode 1'b0: MII mode
5:4	RW	0x0	gmac_clk_sel RGMII clock selection 2'b00: 125MHz 2'b11: 25MHz 2'b10: 2.5MHz
3	RW	0x1	rmii_clk_sel RMII clock selection 1'b1: 25MHz 1'b0: 2.5MHz

Bit	Attr	Reset Value	Description
2:0	RO	0x0	reserved

GRF_SOC_CON6
Address: Operational Base + offset (0x0c218)
SoC control register 6

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15	RW	0x0	gmac_rxclk_dly_ena RGMII TX clock delayline enable 1'b1: enable 1'b0: disable
14:8	RW	0x00	gmac_clk_rx_dl_cfg RGMII RX clock delayline value
7	RW	0x0	gmac_txclk_dly_ena RGMII TX clock delayline enable 1'b1: enable 1'b0: disable
6:0	RW	0x00	gmac_clk_tx_dl_cfg RGMII TX clock delayline value

GRF_GPIO2A_IOMUX
Address: Operational Base + offset (0x0e000)
GPIO2A iomux control

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:14	RW	0x0	gpio2a7_sel GPIO2A[7] iomux select 2'b00: gpio 2'b01: vop_data7 2'b10: i2c7nfc_sda 2'b11: cif_data7
13:12	RW	0x0	gpio2a6_sel GPIO2A[6] iomux select 2'b00: gpio 2'b01: vop_data6 2'b10: uphyjtag_tms 2'b11:cif_data6
11:10	RW	0x0	gpio2a5_sel GPIO2A[5] iomux select 2'b00: gpio 2'b01: vop_data5 2'b10: uphyjtag_tck 2'b11: cif_data5
9:8	RW	0x0	gpio2a4_sel GPIO2A[4] iomux select 2'b00: gpio 2'b01: vop_data4 2'b10: uphyjtag_tdo 2'b11: cif_data4
7:6	RW	0x0	gpio2a3_sel GPIO2A[3] iomux select 2'b00: gpio 2'b01: vop_data3 2'b10: uphyjtag_tdi 2'b11: cif_data3

Bit	Attr	Reset Value	Description
5:4	RW	0x0	gpio2a2_sel GPIO2A[2] iomux select 2'b00: gpio 2'b01: vop_data2 2'b10: uphyjtag_trstn 2'b11: cif_data2
3:2	RW	0x0	gpio2a1_sel GPIO2A[1] iomux select 2'b00: gpio 2'b01: vop_data1 2'b10: i2c2tp_scl 2'b11: cif_data1
1:0	RW	0x0	gpio2a0_sel GPIO2A[0] iomux select 2'b00: gpio 2'b01: vop_data0 2'b10: i2c2tp_sda 2'b11: cif_data0

GRF_GPIO2B_IOMUX
Address: Operational Base + offset (0x0e004)
GPIO2B iomux control

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:10	RO	0x0	reserved
9:8	RW	0x0	gpio2b4_sel GPIO2B[4] iomux select 2'b00: gpio 2'b01: spi2tpm_csn0 2'b10: reserved 2'b11: reserved

Bit	Attr	Reset Value	Description
7:6	RW	0x0	gpio2b3_sel GPIO2B[3] iomux select 2'b00: gpio 2'b01: spi2tpm_clk 2'b10: vop_den 2'b11: cif_clkouta
5:4	RW	0x0	gpio2b2_sel GPIO2B[2] iomux select 2'b00: gpio 2'b01: spi2tpm_txd 2'b10: i2c6tpm_scl 2'b11: cif_clkin
3:2	RW	0x0	gpio2b1_sel GPIO2B[1] iomux select 2'b00: gpio 2'b01: spi2tpm_rxd 2'b10: i2c6tpm_sda 2'b11: cif_href
1:0	RW	0x0	gpio2b0_sel GPIO2B[0] iomux select 2'b00: gpio 2'b01: vop_dclk 2'b10: i2c7nfc_scl 2'b11: cif_vsync

GRF_GPIO2C_IOMUX
Address: Operational Base + offset (0x0e008)
GPIO2C iomux control

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;

Bit	Attr	Reset Value	Description
15:14	RW	0x0	gpio2c7_sel GPIO2C[7] iomux select 2'b00: gpio 2'b01: sdio_data3 2'b10: spi5expplus_csn0 2'b11: reserved
13:12	RW	0x0	gpio2c6_sel GPIO2C[6] iomux select 2'b00: gpio 2'b01: sdio_data2 2'b10: spi5expplus_clk 2'b11: reserved
11:10	RW	0x0	gpio2c5_sel GPIO2C[5] iomux select 2'b00: gpio 2'b01: sdio_data1 2'b10: spi5expplus_txd 2'b11: reserved
9:8	RW	0x0	gpio2c4_sel GPIO2C[4] iomux select 2'b00: gpio 2'b01: sdio_data0 2'b10: spi5expplus_rxd 2'b11: reserved
7:6	RW	0x0	gpio2c3_sel GPIO2C[3] iomux select 2'b00: gpio 2'b01: uart0bt_rtsn 2'b10: reserved 2'b11: reserved
5:4	RW	0x0	gpio2c2_sel GPIO2C[2] iomux select 2'b00: gpio 2'b01: uart0bt_ctsn 2'b10: reserved 2'b11: reserved
3:2	RW	0x0	gpio2c1_sel GPIO2C[1] iomux select 2'b00: gpio 2'b01: uart0bt_sout 2'b10: reserved 2'b11: reserved

Bit	Attr	Reset Value	Description
1:0	RW	0x0	gpio2c0_sel GPIO2C[0] iomux select 2'b00: gpio 2'b01: uart0bt_sin 2'b10: reserved 2'b11: reserved

GRF_GPIO2D_IOMUX
Address: Operational Base + offset (0x0e00c)
GPIO2D iomux control

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:10	RO	0x0	reserved
9:8	RW	0x0	gpio2d4_sel GPIO2D[4] iomux select 2'b00: gpio 2'b01: sdio_bkpwr 2'b10: reserved 2'b11: reserved
7:6	RW	0x0	gpio2d3_sel GPIO2D[3] iomux select 2'b00: gpio 2'b01: sdio_pwren 2'b10: reserved 2'b11: reserved
5:4	RW	0x0	gpio2d2_sel GPIO2D[2] iomux select 2'b00: gpio 2'b01: sdio_detectn 2'b10: pcie_clkreqn 2'b11: reserved

Bit	Attr	Reset Value	Description
3:2	RW	0x0	gpio2d1_sel GPIO2D[1] iomux select 2'b00: gpio 2'b01: sdio_clkout 2'b10: test_clkout1 2'b11: reserved
1:0	RW	0x0	gpio2d0_sel GPIO2D[0] iomux select 2'b00: gpio 2'b01: sdio_cmd 2'b10: reserved 2'b11: reserved

GRF_GPIO3A_IOMUX
Address: Operational Base + offset (0x0e010)
GPIO3A iomux control

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:14	RW	0x0	gpio3a7_sel GPIO3A[7] iomux select 2'b00: gpio 2'b01: mac_rxd1 2'b10: spi0norcodec_csn0 2'b11: reserved
13:12	RW	0x0	gpio3a6_sel GPIO3A[6] iomux select 2'b00: gpio 2'b01: mac_rxd0 2'b10: spi0norcodec_clk 2'b11: reserved

Bit	Attr	Reset Value	Description
11:10	RW	0x0	gpio3a5_sel GPIO3A[5] iomux select 2'b00: gpio 2'b01: mac_txd1 2'b10: spi0norcodec_txd 2'b11: reserved
9:8	RW	0x0	gpio3a4_sel GPIO3A[4] iomux select 2'b00: gpio 2'b01: mac_txd0 2'b10: spi0norcodec_rxd 2'b11: reserved
7:6	RW	0x0	gpio3a3_sel GPIO3A[3] iomux select 2'b00: gpio 2'b01: mac_rxd3 2'b10: spi4exp_csn0 2'b11: trace_data15
5:4	RW	0x0	gpio3a2_sel GPIO3A[2] iomux select 2'b00: gpio 2'b01: mac_rxd2 2'b10: spi4exp_clk 2'b11: trace_data14
3:2	RW	0x0	gpio3a1_sel GPIO3A[1] iomux select 2'b00: gpio 2'b01: mac_txd3 2'b10: spi4exp_txd 2'b11: trace_data13
1:0	RW	0x0	gpio3a0_sel GPIO3A[0] iomux select 2'b00: gpio 2'b01: mac_txd2 2'b10: spi4exp_rxd 2'b11: trace_data12

GRF_GPIO3B_IOMUX
Address: Operational Base + offset (0x0e014)
GPIO3B iomux control

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:14	RW	0x0	gpio3b7_sel GPIO3B[7] iomux select 2'b00: gpio 2'b01: mac_crs 2'b10: uart3gps_sout 2'b11: cif_clkoutb
13:12	RW	0x0	gpio3b6_sel GPIO3B[6] iomux select 2'b00: gpio 2'b01: mac_rxclk 2'b10: uart3gps_sin 2'b11: reserved
11:10	RW	0x0	gpio3b5_sel GPIO3B[5] iomux select 2'b00: gpio 2'b01: mac_mdio 2'b10: uart1bb_sout 2'b11: reserved
9:8	RW	0x0	gpio3b4_sel GPIO3B[4] iomux select 2'b00: gpio 2'b01: mac_txen 2'b10: uart1bb_sin 2'b11: reserved
7:6	RW	0x0	gpio3b3_sel GPIO3B[3] iomux select 2'b00: gpio 2'b01: mac_clk 2'b10: i2c5trackpad_scl 2'b11: reserved

Bit	Attr	Reset Value	Description
5:4	RW	0x0	gpio3b2_sel GPIO3B[2] iomux select 2'b00: gpio 2'b01: mac_rxer 2'b10: i2c5trackpad_sda 2'b11: reserved
3:2	RW	0x0	gpio3b1_sel GPIO3B[1] iomux select 2'b00: gpio 2'b01: mac_rxdv 2'b10: reserved 2'b11: reserved
1:0	RW	0x0	gpio3b0_sel GPIO3B[0] iomux select 2'b00: gpio 2'b01: mac_mdc 2'b10: spi0norcodec_csn1 2'b11: reserved

GRF_GPIO3C_IOMUX
Address: Operational Base + offset (0x0e018)
GPIO3C iomux control

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:4	RO	0x0	reserved
3:2	RW	0x0	gpio3c1_sel GPIO3C[1] iomux select 2'b00: gpio 2'b01: mac_txclk 2'b10: uart3gps_rtsn 2'b11: reserved

Bit	Attr	Reset Value	Description
1:0	RW	0x0	gpio3c0_sel GPIO3C[0] iomux select 2'b00: gpio 2'b01: mac_col 2'b10: uart3gps_ctsn 2'b11: spdif_txb

GRF_GPIO3D_IOMUX
Address: Operational Base + offset (0x0e01c)
GPIO3D iomux control

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:14	RW	0x0	gpio3d7_sel GPIO3D[7] iomux select 2'b00: gpio 2'b01: i2s0_sdo0 2'b10: trace_data7 2'b11: a53l2_wfi
13:12	RW	0x0	gpio3d6_sel GPIO3D[6] iomux select 2'b00: gpio 2'b01: i2s0_sdi3sdo1 2'b10: trace_data6 2'b11: a72l2_wfi
11:10	RW	0x0	gpio3d5_sel GPIO3D[5] iomux select 2'b00: gpio 2'b01: i2s0_sdi2sdo2 2'b10: trace_data5 2'b11: a53core3_wfi

Bit	Attr	Reset Value	Description
9:8	RW	0x0	gpio3d4_sel GPIO3D[4] iomux select 2'b00: gpio 2'b01: i2s0_sdi1sdo3 2'b10: trace_data4 2'b11: a53core2_wfi
7:6	RW	0x0	gpio3d3_sel GPIO3D[3] iomux select 2'b00: gpio 2'b01: i2s0_sdi0 2'b10: trace_data3 2'b11: a53core1_wfi
5:4	RW	0x0	gpio3d2_sel GPIO3D[2] iomux select 2'b00: gpio 2'b01: i2s0_lrcktx 2'b10: trace_data2 2'b11: a53core0_wfi
3:2	RW	0x0	gpio3d1_sel GPIO3D[1] iomux select 2'b00: gpio 2'b01: i2s0_lrckrx 2'b10: trace_data1 2'b11: a72core1_wfi
1:0	RW	0x0	gpio3d0_sel GPIO3D[0] iomux select 2'b00: gpio 2'b01: i2s0_sclk 2'b10: trace_data0 2'b11: a72core0_wfi

GRF_GPIO4A_IOMUX
Address: Operational Base + offset (0x0e020)
GPIO4A iomux control

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:14	RW	0x0	gpio4a7_sel GPIO4A[7] iomux select 2'b00: gpio 2'b01: i2s1_sdo0 2'b10: reserved 2'b11: reserved
13:12	RW	0x0	gpio4a6_sel GPIO4A[6] iomux select 2'b00: gpio 2'b01: i2s1_sdi0 2'b10: reserved 2'b11: reserved
11:10	RW	0x0	gpio4a5_sel GPIO4A[5] iomux select 2'b00: gpio 2'b01: i2s1_lrcktx 2'b10: trace_data11 2'b11: reserved
9:8	RW	0x0	gpio4a4_sel GPIO4A[4] iomux select 2'b00: gpio 2'b01: i2s1_lrckrx 2'b10: trace_data10 2'b11: reserved
7:6	RW	0x0	gpio4a3_sel GPIO4A[3] iomux select 2'b00: gpio 2'b01: i2s1_sclk 2'b10: trace_data9 2'b11: reserved

Bit	Attr	Reset Value	Description
5:4	RW	0x0	gpio4a2_sel GPIO4A[2] iomux select 2'b00: gpio 2'b01: i2c1audiocam_scl 2'b10: trace_data8 2'b11: reserved
3:2	RW	0x0	gpio4a1_sel GPIO4A[1] iomux select 2'b00: gpio 2'b01: i2c1audiocam_sda 2'b10: trace_clk 2'b11: reserved
1:0	RW	0x0	gpio4a0_sel GPIO4A[0] iomux select 2'b00: gpio 2'b01: i2s_clk 2'b10: trace_ctl 2'b11: lpm0_wfi

GRF_GPIO4B_IOMUX
Address: Operational Base + offset (0x0e024)
GPIO4B iomux control

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:12	RO	0x0	reserved
11:10	RW	0x0	gpio4b5_sel GPIO4B[5] iomux select 2'b00: gpio 2'b01: sdmmc_cmd 2'b10: mcujtag_tms 2'b11: hdcpjtag_tms

Bit	Attr	Reset Value	Description
9:8	RW	0x0	gpio4b4_sel GPIO4B[4] iomux select 2'b00: gpio 2'b01: sdmmc_clkout 2'b10: mcujtag_tck 2'b11: hdcpjtag_tck
7:6	RW	0x0	gpio4b3_sel GPIO4B[3] iomux select 2'b00: gpio 2'b01: sdmmc_data3 2'b10: cxcsjtag_tms 2'b11: hdcpjtag_tdo
5:4	RW	0x0	gpio4b2_sel GPIO4B[2] iomux select 2'b00: gpio 2'b01: sdmmc_data2 2'b10: cxcsjtag_tck 2'b11: hdcpjtag_tdi
3:2	RW	0x0	gpio4b1_sel GPIO4B[1] iomux select 2'b00: gpio 2'b01: sdmmc_data1 2'b10: uart2dbga_sout 2'b11: hdcpjtag_trstn
1:0	RW	0x0	gpio4b0_sel GPIO4B[0] iomux select 2'b00: gpio 2'b01: sdmmc_data0 2'b10: uart2dbga_sin 2'b11: reserved

GRF_GPIO4C_IOMUX
Address: Operational Base + offset (0x0e028)
GPIO4C iomux control

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:14	RW	0x0	gpio4c7_sel GPIO4C[7] iomux select 2'b00: gpio 2'b01: hdmi_cecinout 2'b10: edp_hotplug 2'b11: reserved
13:12	RW	0x0	gpio4c6_sel GPIO4C[6] iomux select 2'b00: gpio 2'b01: pwm_1 2'b10: reserved 2'b11: reserved
11:10	RW	0x0	gpio4c5_sel GPIO4C[5] iomux select 2'b00: gpio 2'b01: spdif_tx 2'b10: reserved 2'b11: reserved
9:8	RW	0x0	gpio4c4_sel GPIO4C[4] iomux select 2'b00: gpio 2'b01: uart2dbgc_sout 2'b10: uarthdcp_sout 2'b11: reserved
7:6	RW	0x0	gpio4c3_sel GPIO4C[3] iomux select 2'b00: gpio 2'b01: uart2dbgc_sin 2'b10: uarthdcp_sin 2'b11: reserved

Bit	Attr	Reset Value	Description
5:4	RW	0x0	gpio4c2_sel GPIO4C[2] iomux select 2'b00: gpio 2'b01: pwm_0 2'b10: vop0_pwm 2'b11: vop1_pwm
3:2	RW	0x0	gpio4c1_sel GPIO4C[1] iomux select 2'b00: gpio 2'b01: i2c3hdmi_scl 2'b10: uart2dbgb_sout 2'b11: hdmii2c_scl
1:0	RW	0x0	gpio4c0_sel GPIO4C[0] iomux select 2'b00: gpio 2'b01: i2c3hdmi_sda 2'b10: uart2dbgb_sin 2'b11: hdmii2c_sda

GRF_GPIO4D_IOMUX
Address: Operational Base + offset (0x0e02c)
GPIO4D iomux control

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:4	RO	0x0	reserved
3:2	RW	0x0	gpio4d1_sel GPIO4D[1] iomux select 2'b00: gpio 2'b01: dp_hotplug 2'b10: reserved 2'b11: reserved

Bit	Attr	Reset Value	Description
1:0	RW	0x0	gpio4d0_sel GPIO4D[0] iomux select 2'b00: gpio 2'b01: pcie_clkreqnb 2'b10: reserved 2'b11: reserved

GRF_GPIO2A_P

Address: Operational Base + offset (0x0e040)

GPIO2A PU/PD control

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:14	RW	0x1	gpio2a7_p GPIO2A PU/PD programmation section, every GPIO bit corresponding to 2bits 2'b00: Z(Normal operaton); 2'b01: weak 1(pull-up); 2'b10: weak 0(pull-down); 2'b11: Reserved;
13:12	RW	0x2	gpio2a6_p GPIO2A PU/PD programmation section, every GPIO bit corresponding to 2bits 2'b00: Z(Normal operaton); 2'b01: weak 1(pull-up); 2'b10: weak 0(pull-down); 2'b11: Reserved;

Bit	Attr	Reset Value	Description
11:10	RW	0x2	gpio2a5_p GPIO2A PU/PD programmation section, every GPIO bit corresponding to 2bits 2'b00: Z(Normal operaton); 2'b01: weak 1(pull-up); 2'b10: weak 0(pull-down); 2'b11: Reserved;
9:8	RW	0x2	gpio2a4_p GPIO2A PU/PD programmation section, every GPIO bit corresponding to 2bits 2'b00: Z(Normal operaton); 2'b01: weak 1(pull-up); 2'b10: weak 0(pull-down); 2'b11: Reserved;
7:6	RW	0x2	gpio2a3_p GPIO2A PU/PD programmation section, every GPIO bit corresponding to 2bits 2'b00: Z(Normal operaton); 2'b01: weak 1(pull-up); 2'b10: weak 0(pull-down); 2'b11: Reserved;
5:4	RW	0x2	gpio2a2_p GPIO2A PU/PD programmation section, every GPIO bit corresponding to 2bits 2'b00: Z(Normal operaton); 2'b01: weak 1(pull-up); 2'b10: weak 0(pull-down); 2'b11: Reserved;
3:2	RW	0x1	gpio2a1_p GPIO2A PU/PD programmation section, every GPIO bit corresponding to 2bits 2'b00: Z(Normal operaton); 2'b01: weak 1(pull-up); 2'b10: weak 0(pull-down); 2'b11: Reserved;
1:0	RW	0x1	gpio2a0_p GPIO2A PU/PD programmation section, every GPIO bit corresponding to 2bits 2'b00: Z(Normal operaton); 2'b01: weak 1(pull-up); 2'b10: weak 0(pull-down); 2'b11: Reserved;

GRF_GPIO2B_P
Address: Operational Base + offset (0x0e044)

GPIO2B PU/PD control

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:10	RO	0x0	reserved
9:8	RW	0x1	gpio2b4_p GPIO2B PU/PD programmation section, every GPIO bit corresponding to 2bits 2'b00: Z(Normal operaton); 2'b01: weak 1(pull-up); 2'b10: weak 0(pull-down); 2'b11: Reserved;
7:6	RW	0x1	gpio2b3_p GPIO2B PU/PD programmation section, every GPIO bit corresponding to 2bits 2'b00: Z(Normal operaton); 2'b01: weak 1(pull-up); 2'b10: weak 0(pull-down); 2'b11: Reserved;
5:4	RW	0x1	gpio2b2_p GPIO2B PU/PD programmation section, every GPIO bit corresponding to 2bits 2'b00: Z(Normal operaton); 2'b01: weak 1(pull-up); 2'b10: weak 0(pull-down); 2'b11: Reserved;
3:2	RW	0x1	gpio2b1_p GPIO2B PU/PD programmation section, every GPIO bit corresponding to 2bits 2'b00: Z(Normal operaton); 2'b01: weak 1(pull-up); 2'b10: weak 0(pull-down); 2'b11: Reserved;

Bit	Attr	Reset Value	Description
1:0	RW	0x1	gpio2b0_p GPIO2B PU/PD programmation section, every GPIO bit corresponding to 2bits 2'b00: Z(Normal operaton); 2'b01: weak 1(pull-up); 2'b10: weak 0(pull-down); 2'b11: Reserved;

GRF_GPIO2C_P

Address: Operational Base + offset (0x0e048)

GPIO2C PU/PD control

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:14	RW	0x3	gpio2c7_p GPIO2C PE/PS programmation section, every GPIO bit corresponding to 2bits 2'b00: Z(Normal operation); 2'b01: weak 0(pull-down); 2'b10: Z(Normal operation); 2'b11: weak 1(pull-up);
13:12	RW	0x3	gpio2c6_p GPIO2C PE/PS programmation section, every GPIO bit corresponding to 2bits 2'b00: Z(Normal operation); 2'b01: weak 0(pull-down); 2'b10: Z(Normal operation); 2'b11: weak 1(pull-up);

Bit	Attr	Reset Value	Description
11:10	RW	0x3	gpio2c5_p GPIO2C PE/PS programmation section, every GPIO bit corresponding to 2bits 2'b00: Z(Normal operation); 2'b01: weak 0(pull-down); 2'b10: Z(Normal operation); 2'b11: weak 1(pull-up);
9:8	RW	0x3	gpio2c4_p GPIO2C PE/PS programmation section, every GPIO bit corresponding to 2bits 2'b00: Z(Normal operation); 2'b01: weak 0(pull-down); 2'b10: Z(Normal operation); 2'b11: weak 1(pull-up);
7:6	RW	0x3	gpio2c3_p GPIO2C PE/PS programmation section, every GPIO bit corresponding to 2bits 2'b00: Z(Normal operation); 2'b01: weak 0(pull-down); 2'b10: Z(Normal operation); 2'b11: weak 1(pull-up);
5:4	RW	0x3	gpio2c2_p GPIO2C PE/PS programmation section, every GPIO bit corresponding to 2bits 2'b00: Z(Normal operation); 2'b01: weak 0(pull-down); 2'b10: Z(Normal operation); 2'b11: weak 1(pull-up);
3:2	RW	0x3	gpio2c1_p GPIO2C PE/PS programmation section, every GPIO bit corresponding to 2bits 2'b00: Z(Normal operation); 2'b01: weak 0(pull-down); 2'b10: Z(Normal operation); 2'b11: weak 1(pull-up);
1:0	RW	0x3	gpio2c0_p GPIO2C PE/PS programmation section, every GPIO bit corresponding to 2bits 2'b00: Z(Normal operation); 2'b01: weak 0(pull-down); 2'b10: Z(Normal operation); 2'b11: weak 1(pull-up);

GRF_GPIO2D_P
Address: Operational Base + offset (0x0e04c)

GPIO2D PU/PD control

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:10	RO	0x0	reserved
9:8	RW	0x0	gpio2d4_p GPIO2D PE/PS programmation section, every GPIO bit corresponding to 2bits 2'b00: Z(Normal operation); 2'b01: weak 0(pull-down); 2'b10: Z(Normal operation); 2'b11: weak 1(pull-up);
7:6	RW	0x1	gpio2d3_p GPIO2D PE/PS programmation section, every GPIO bit corresponding to 2bits 2'b00: Z(Normal operation); 2'b01: weak 0(pull-down); 2'b10: Z(Normal operation); 2'b11: weak 1(pull-up);
5:4	RW	0x3	gpio2d2_p GPIO2D PE/PS programmation section, every GPIO bit corresponding to 2bits 2'b00: Z(Normal operation); 2'b01: weak 0(pull-down); 2'b10: Z(Normal operation); 2'b11: weak 1(pull-up);
3:2	RW	0x3	gpio2d1_p GPIO2D PE/PS programmation section, every GPIO bit corresponding to 2bits 2'b00: Z(Normal operation); 2'b01: weak 0(pull-down); 2'b10: Z(Normal operation); 2'b11: weak 1(pull-up);

Bit	Attr	Reset Value	Description
1:0	RW	0x3	gpio2d0_p GPIO2D PE/PS programmation section, every GPIO bit corresponding to 2bits 2'b00: Z(Normal operation); 2'b01: weak 0(pull-down); 2'b10: Z(Normal operation); 2'b11: weak 1(pull-up);

GRF_GPIO3A_P
Address: Operational Base + offset (0x0e050)
GPIO3A PU/PD control

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:14	RW	0x1	gpio3a7_p GPIO3A PU/PD programmation section, every GPIO bit corresponding to 2bits 2'b00: Z(Normal operaton); 2'b01: weak 1(pull-up); 2'b10: weak 0(pull-down); 2'b11: Reserved;
13:12	RW	0x1	gpio3a6_p GPIO3A PU/PD programmation section, every GPIO bit corresponding to 2bits 2'b00: Z(Normal operaton); 2'b01: weak 1(pull-up); 2'b10: weak 0(pull-down); 2'b11: Reserved;

Bit	Attr	Reset Value	Description
11:10	RW	0x2	gpio3a5_p GPIO3A PU/PD programmation section, every GPIO bit corresponding to 2bits 2'b00: Z(Normal operaton); 2'b01: weak 1(pull-up); 2'b10: weak 0(pull-down); 2'b11: Reserved;
9:8	RW	0x2	gpio3a4_p GPIO3A PU/PD programmation section, every GPIO bit corresponding to 2bits 2'b00: Z(Normal operaton); 2'b01: weak 1(pull-up); 2'b10: weak 0(pull-down); 2'b11: Reserved;
7:6	RW	0x1	gpio3a3_p GPIO3A PU/PD programmation section, every GPIO bit corresponding to 2bits 2'b00: Z(Normal operaton); 2'b01: weak 1(pull-up); 2'b10: weak 0(pull-down); 2'b11: Reserved;
5:4	RW	0x1	gpio3a2_p GPIO3A PU/PD programmation section, every GPIO bit corresponding to 2bits 2'b00: Z(Normal operaton); 2'b01: weak 1(pull-up); 2'b10: weak 0(pull-down); 2'b11: Reserved;
3:2	RW	0x2	gpio3a1_p GPIO3A PU/PD programmation section, every GPIO bit corresponding to 2bits 2'b00: Z(Normal operaton); 2'b01: weak 1(pull-up); 2'b10: weak 0(pull-down); 2'b11: Reserved;
1:0	RW	0x2	gpio3a0_p GPIO3A PU/PD programmation section, every GPIO bit corresponding to 2bits 2'b00: Z(Normal operaton); 2'b01: weak 1(pull-up); 2'b10: weak 0(pull-down); 2'b11: Reserved;

GRF_GPIO3B_P
Address: Operational Base + offset (0x0e054)

GPIO3B PU/PD control

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:14	RW	0x1	gpio3b7_p GPIO3B PU/PD programmation section, every GPIO bit corresponding to 2bits 2'b00: Z(Normal operaton); 2'b01: weak 1(pull-up); 2'b10: weak 0(pull-down); 2'b11: Reserved;
13:12	RW	0x1	gpio3b6_p GPIO3B PU/PD programmation section, every GPIO bit corresponding to 2bits 2'b00: Z(Normal operaton); 2'b01: weak 1(pull-up); 2'b10: weak 0(pull-down); 2'b11: Reserved;
11:10	RW	0x1	gpio3b5_p GPIO3B PU/PD programmation section, every GPIO bit corresponding to 2bits 2'b00: Z(Normal operaton); 2'b01: weak 1(pull-up); 2'b10: weak 0(pull-down); 2'b11: Reserved;
9:8	RW	0x1	gpio3b4_p GPIO3B PU/PD programmation section, every GPIO bit corresponding to 2bits 2'b00: Z(Normal operaton); 2'b01: weak 1(pull-up); 2'b10: weak 0(pull-down); 2'b11: Reserved;

Bit	Attr	Reset Value	Description
7:6	RW	0x1	gpio3b3_p GPIO3B PU/PD programmation section, every GPIO bit corresponding to 2bits 2'b00: Z(Normal operaton); 2'b01: weak 1(pull-up); 2'b10: weak 0(pull-down); 2'b11: Reserved;
5:4	RW	0x1	gpio3b2_p GPIO3B PU/PD programmation section, every GPIO bit corresponding to 2bits 2'b00: Z(Normal operaton); 2'b01: weak 1(pull-up); 2'b10: weak 0(pull-down); 2'b11: Reserved;
3:2	RW	0x2	gpio3b1_p GPIO3B PU/PD programmation section, every GPIO bit corresponding to 2bits 2'b00: Z(Normal operaton); 2'b01: weak 1(pull-up); 2'b10: weak 0(pull-down); 2'b11: Reserved;
1:0	RW	0x1	gpio3b0_p GPIO3B PU/PD programmation section, every GPIO bit corresponding to 2bits 2'b00: Z(Normal operaton); 2'b01: weak 1(pull-up); 2'b10: weak 0(pull-down); 2'b11: Reserved;

GRF_GPIO3C_P
Address: Operational Base + offset (0x0e058)
GPIO3C PU/PD control

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:4	RO	0x0	reserved
3:2	RW	0x1	gpio3c1_p GPIO3A PU/PD programmation section, every GPIO bit corresponding to 2bits 2'b00: Z(Normal operaton); 2'b01: weak 1(pull-up); 2'b10: weak 0(pull-down); 2'b11: Reserved;
1:0	RW	0x1	gpio3c0_p GPIO3C PU/PD programmation section, every GPIO bit corresponding to 2bits 2'b00: Z(Normal operaton); 2'b01: weak 1(pull-up); 2'b10: weak 0(pull-down); 2'b11: Reserved;

GRF_GPIO3D_P
Address: Operational Base + offset (0x0e05c)
GPIO3D PU/PD control

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:14	RW	0x2	gpio3d7_p GPIO3D PU/PD programmation section, every GPIO bit corresponding to 2bits 2'b00: Z(Normal operaton); 2'b01: weak 1(pull-up); 2'b10: weak 0(pull-down); 2'b11: Reserved;
13:12	RW	0x2	gpio3d6_p GPIO3D PU/PD programmation section, every GPIO bit corresponding to 2bits 2'b00: Z(Normal operaton); 2'b01: weak 1(pull-up); 2'b10: weak 0(pull-down); 2'b11: Reserved;
11:10	RW	0x2	gpio3d5_p GPIO3D PU/PD programmation section, every GPIO bit corresponding to 2bits 2'b00: Z(Normal operaton); 2'b01: weak 1(pull-up); 2'b10: weak 0(pull-down); 2'b11: Reserved;
9:8	RW	0x2	gpio3d4_p GPIO3D PU/PD programmation section, every GPIO bit corresponding to 2bits 2'b00: Z(Normal operaton); 2'b01: weak 1(pull-up); 2'b10: weak 0(pull-down); 2'b11: Reserved;

Bit	Attr	Reset Value	Description
7:6	RW	0x2	gpio3d3_p GPIO3D PU/PD programmation section, every GPIO bit corresponding to 2bits 2'b00: Z(Normal operaton); 2'b01: weak 1(pull-up); 2'b10: weak 0(pull-down); 2'b11: Reserved;
5:4	RW	0x2	gpio3d2_p GPIO3D PU/PD programmation section, every GPIO bit corresponding to 2bits 2'b00: Z(Normal operaton); 2'b01: weak 1(pull-up); 2'b10: weak 0(pull-down); 2'b11: Reserved;
3:2	RW	0x2	gpio3d1_p GPIO3D PU/PD programmation section, every GPIO bit corresponding to 2bits 2'b00: Z(Normal operaton); 2'b01: weak 1(pull-up); 2'b10: weak 0(pull-down); 2'b11: Reserved;
1:0	RW	0x2	gpio3d0_p GPIO3D PU/PD programmation section, every GPIO bit corresponding to 2bits 2'b00: Z(Normal operaton); 2'b01: weak 1(pull-up); 2'b10: weak 0(pull-down); 2'b11: Reserved;

GRF_GPIO4A_P
Address: Operational Base + offset (0x0e060)
GPIO4A PU/PD control

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:14	RW	0x2	gpio4a7_p GPIO4A PU/PD programmation section, every GPIO bit corresponding to 2bits 2'b00: Z(Normal operaton); 2'b01: weak 1(pull-up); 2'b10: weak 0(pull-down); 2'b11: Reserved;
13:12	RW	0x2	gpio4a6_p GPIO4A PU/PD programmation section, every GPIO bit corresponding to 2bits 2'b00: Z(Normal operaton); 2'b01: weak 1(pull-up); 2'b10: weak 0(pull-down); 2'b11: Reserved;
11:10	RW	0x2	gpio4a5_p GPIO4A PU/PD programmation section, every GPIO bit corresponding to 2bits 2'b00: Z(Normal operaton); 2'b01: weak 1(pull-up); 2'b10: weak 0(pull-down); 2'b11: Reserved;
9:8	RW	0x2	gpio4a4_p GPIO4A PU/PD programmation section, every GPIO bit corresponding to 2bits 2'b00: Z(Normal operaton); 2'b01: weak 1(pull-up); 2'b10: weak 0(pull-down); 2'b11: Reserved;

Bit	Attr	Reset Value	Description
7:6	RW	0x2	gpio4a3_p GPIO4A PU/PD programmation section, every GPIO bit corresponding to 2bits 2'b00: Z(Normal operaton); 2'b01: weak 1(pull-up); 2'b10: weak 0(pull-down); 2'b11: Reserved;
5:4	RW	0x1	gpio4a2_p GPIO4A PU/PD programmation section, every GPIO bit corresponding to 2bits 2'b00: Z(Normal operaton); 2'b01: weak 1(pull-up); 2'b10: weak 0(pull-down); 2'b11: Reserved;
3:2	RW	0x1	gpio4a1_p GPIO4A PU/PD programmation section, every GPIO bit corresponding to 2bits 2'b00: Z(Normal operaton); 2'b01: weak 1(pull-up); 2'b10: weak 0(pull-down); 2'b11: Reserved;
1:0	RW	0x2	gpio4a0_p GPIO4A PU/PD programmation section, every GPIO bit corresponding to 2bits 2'b00: Z(Normal operaton); 2'b01: weak 1(pull-up); 2'b10: weak 0(pull-down); 2'b11: Reserved;

GRF_GPIO4B_P
Address: Operational Base + offset (0x0e064)
GPIO4B PU/PD control

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:12	RO	0x0	reserved
11:10	RW	0x1	gpio4b5_p GPIO4B PU/PD programmation section, every GPIO bit corresponding to 2bits 2'b00: Z(Normal operaton); 2'b01: weak 1(pull-up); 2'b10: weak 0(pull-down); 2'b11: Reserved;
9:8	RW	0x2	gpio4b4_p GPIO4B PU/PD programmation section, every GPIO bit corresponding to 2bits 2'b00: Z(Normal operaton); 2'b01: weak 1(pull-up); 2'b10: weak 0(pull-down); 2'b11: Reserved;
7:6	RW	0x1	gpio4b3_p GPIO4B PU/PD programmation section, every GPIO bit corresponding to 2bits 2'b00: Z(Normal operaton); 2'b01: weak 1(pull-up); 2'b10: weak 0(pull-down); 2'b11: Reserved;
5:4	RW	0x1	gpio4b2_p GPIO4B PU/PD programmation section, every GPIO bit corresponding to 2bits 2'b00: Z(Normal operaton); 2'b01: weak 1(pull-up); 2'b10: weak 0(pull-down); 2'b11: Reserved;

Bit	Attr	Reset Value	Description
3:2	RW	0x1	gpio4b1_p GPIO4B PU/PD programmation section, every GPIO bit corresponding to 2bits 2'b00: Z(Normal operaton); 2'b01: weak 1(pull-up); 2'b10: weak 0(pull-down); 2'b11: Reserved;
1:0	RW	0x1	gpio4b0_p GPIO4B PU/PD programmation section, every GPIO bit corresponding to 2bits 2'b00: Z(Normal operaton); 2'b01: weak 1(pull-up); 2'b10: weak 0(pull-down); 2'b11: Reserved;

GRF_GPIO4C_P
Address: Operational Base + offset (0x0e068)
GPIO4C PU/PD control

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:14	RW	0x1	gpio4c7_p GPIO4C PU/PD programmation section, every GPIO bit corresponding to 2bits 2'b00: Z(Normal operaton); 2'b01: weak 1(pull-up); 2'b10: weak 0(pull-down); 2'b11: Reserved;

Bit	Attr	Reset Value	Description
13:12	RW	0x2	gpio4c6_p GPIO4C PU/PD programmation section, every GPIO bit corresponding to 2bits 2'b00: Z(Normal operaton); 2'b01: weak 1(pull-up); 2'b10: weak 0(pull-down); 2'b11: Reserved;
11:10	RW	0x2	gpio4c5_p GPIO4C PU/PD programmation section, every GPIO bit corresponding to 2bits 2'b00: Z(Normal operaton); 2'b01: weak 1(pull-up); 2'b10: weak 0(pull-down); 2'b11: Reserved;
9:8	RW	0x1	gpio4c4_p GPIO4C PU/PD programmation section, every GPIO bit corresponding to 2bits 2'b00: Z(Normal operaton); 2'b01: weak 1(pull-up); 2'b10: weak 0(pull-down); 2'b11: Reserved;
7:6	RW	0x1	gpio4c3_p GPIO4C PU/PD programmation section, every GPIO bit corresponding to 2bits 2'b00: Z(Normal operaton); 2'b01: weak 1(pull-up); 2'b10: weak 0(pull-down); 2'b11: Reserved;
5:4	RW	0x2	gpio4c2_p GPIO4C PU/PD programmation section, every GPIO bit corresponding to 2bits 2'b00: Z(Normal operaton); 2'b01: weak 1(pull-up); 2'b10: weak 0(pull-down); 2'b11: Reserved;
3:2	RW	0x1	gpio4c1_p GPIO4C PU/PD programmation section, every GPIO bit corresponding to 2bits 2'b00: Z(Normal operaton); 2'b01: weak 1(pull-up); 2'b10: weak 0(pull-down); 2'b11: Reserved;

Bit	Attr	Reset Value	Description
1:0	RW	0x1	gpio4c0_p GPIO4C PU/PD programmation section, every GPIO bit corresponding to 2bits 2'b00: Z(Normal operaton); 2'b01: weak 1(pull-up); 2'b10: weak 0(pull-down); 2'b11: Reserved;

GRF_GPIO4D_P

Address: Operational Base + offset (0x0e06c)

GPIO4D PU/PD control

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:14	RO	0x0	reserved
13:12	RW	0x2	gpio4d6_p GPIO4D PU/PD programmation section, every GPIO bit corresponding to 2bits 2'b00: Z(Normal operaton); 2'b01: weak 1(pull-up); 2'b10: weak 0(pull-down); 2'b11: Reserved;
11:10	RW	0x2	gpio4d5_p GPIO4D PU/PD programmation section, every GPIO bit corresponding to 2bits 2'b00: Z(Normal operaton); 2'b01: weak 1(pull-up); 2'b10: weak 0(pull-down); 2'b11: Reserved;

Bit	Attr	Reset Value	Description
9:8	RW	0x2	gpio4d4_p GPIO4D PU/PD programmation section, every GPIO bit corresponding to 2bits 2'b00: Z(Normal operaton); 2'b01: weak 1(pull-up); 2'b10: weak 0(pull-down); 2'b11: Reserved;
7:6	RW	0x2	gpio4d3_p GPIO4D PU/PD programmation section, every GPIO bit corresponding to 2bits 2'b00: Z(Normal operaton); 2'b01: weak 1(pull-up); 2'b10: weak 0(pull-down); 2'b11: Reserved;
5:4	RW	0x2	gpio4d2_p GPIO4D PU/PD programmation section, every GPIO bit corresponding to 2bits 2'b00: Z(Normal operaton); 2'b01: weak 1(pull-up); 2'b10: weak 0(pull-down); 2'b11: Reserved;
3:2	RW	0x2	gpio4d1_p GPIO4D PU/PD programmation section, every GPIO bit corresponding to 2bits 2'b00: Z(Normal operaton); 2'b01: weak 1(pull-up); 2'b10: weak 0(pull-down); 2'b11: Reserved;
1:0	RW	0x1	gpio4d0_p GPIO4D PU/PD programmation section, every GPIO bit corresponding to 2bits 2'b00: Z(Normal operaton); 2'b01: weak 1(pull-up); 2'b10: weak 0(pull-down); 2'b11: Reserved;

GRF_GPIO2A_SR
Address: Operational Base + offset (0x0e080)
GPIO2A slew rate control

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:8	RO	0x0	reserved
7:0	RW	0x00	gpio2a_sr GPIO slew rate programmation section, every GPIO bit corresponding to 1bits 1'b0: slow 1'b1: fast

GRF_GPIO2B_SR
Address: Operational Base + offset (0x0e084)
GPIO2B slew rate control

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:5	RO	0x0	reserved

Bit	Attr	Reset Value	Description
4:0	RW	0x00	gpio2b_sr GPIO slew rate programmation section, every GPIO bit corresponding to 1bits 1'b0: slow 1'b1: fast

GRF_GPIO2C_SR
Address: Operational Base + offset (0x0e088)
GPIO2C slew rate control

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:8	RO	0x0	reserved
7:0	RW	0x00	gpio2c_sr GPIO slew rate programmation section, every GPIO bit corresponding to 1bits 1'b0: slow 1'b1: fast

GRF_GPIO2D_SR
Address: Operational Base + offset (0x0e08c)
GPIO2D slew rate control

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:5	RO	0x0	reserved
4:0	RW	0x00	gpio2d_sr GPIO slew rate programmation section, every GPIO bit corresponding to 1bits 1'b0: slow 1'b1: fast

GRF_GPIO3D_SR
Address: Operational Base + offset (0x0e09c)
GPIO3D slew rate control

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:8	RO	0x0	reserved

Bit	Attr	Reset Value	Description
7:0	RW	0x00	gpio3d_sr GPIO slew rate programmation section, every GPIO bit corresponding to 1bits 1'b0: slow 1'b1: fast

GRF_GPIO4A_SR
Address: Operational Base + offset (0x0e0a0)
GPIO4A slew rate control

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:8	RO	0x0	reserved
7:0	RW	0x00	gpio4a_sr GPIO slew rate programmation section, every GPIO bit corresponding to 1bits 1'b0: slow 1'b1: fast

GRF_GPIO4B_SR
Address: Operational Base + offset (0x0e0a4)
GPIO4B slew rate control

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:6	RO	0x0	reserved
5:0	RW	0x3f	gpio4b_sr GPIO slew rate programmation section, every GPIO bit corresponding to 1bits 1'b0: slow 1'b1: fast

GRF_GPIO4C_SR
Address: Operational Base + offset (0x0e0a8)
GPIO4C slew rate control

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:8	RO	0x0	reserved

Bit	Attr	Reset Value	Description
7:0	RW	0x00	gpio4c_sr GPIO slew rate programmation section, every GPIO bit corresponding to 1bits 1'b0: slow 1'b1: fast

GRF_GPIO4D_SR
Address: Operational Base + offset (0x0e0ac)
GPIO4D slew rate control

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:7	RO	0x0	reserved
6:0	RW	0x00	gpio4d_sr GPIO slew rate programmation section, every GPIO bit corresponding to 1bits 1'b0: slow 1'b1: fast

GRF_GPIO2A_SMT
Address: Operational Base + offset (0x0e0c0)
GPIO2A smitter control register

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:8	RO	0x0	reserved
7:0	RW	0x00	gpio2a_smt GPIO schmitt trigger control, every GPIO bit corresponding to 1bits . 0: No hysteresis 1: Schmitt trigger enabled.

GRF_GPIO2B_SMT
Address: Operational Base + offset (0x0e0c4)
GPIO2B smitter control register

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:8	RO	0x0	reserved

Bit	Attr	Reset Value	Description
7:0	RW	0x00	gpio2b_smt GPIO schmitt trigger control, every GPIO bit corresponding to 1bits . 0: No hysteresis 1: Schmitt trigger enabled.

GRF_GPIO2C_SMT
Address: Operational Base + offset (0x0e0c8)
GPIO2C smitter control register

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x0000	gpio2c_smt GPIO schmitt trigger control, every GPIO bit corresponding to 2 bits . 2'b00: level 0 2'b01: level 1 2'b10: level 2 2'b11: level 3

GRF_GPIO2D_SMT
Address: Operational Base + offset (0x0e0cc)
GPIO2D smitter control register

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x0000	gpio2d_smt GPIO schmitt trigger control, every GPIO bit corresponding to 2 bits . 2'b00: level 0 2'b01: level 1 2'b10: level 2 2'b11: level 3

GRF_GPIO3A_SMT
Address: Operational Base + offset (0x0e0d0)
GPIO3A smitter control register

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:8	RO	0x0	reserved

Bit	Attr	Reset Value	Description
7:0	RW	0xf0	gpio3a_smt GPIO schmitt trigger control, every GPIO bit corresponding to 1bits . 0: No hysteresis 1: Schmitt trigger enabled.

GRF_GPIO3B_SMT
Address: Operational Base + offset (0x0e0d4)
GPIO3B smitter control register

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:8	RO	0x0	reserved
7:0	RW	0x00	gpio3b_smt GPIO schmitt trigger control, every GPIO bit corresponding to 1bits . 0: No hysteresis 1: Schmitt trigger enabled.

GRF_GPIO3C_SMT
Address: Operational Base + offset (0x0e0d8)
GPIO3C smitter control register

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:2	RO	0x0	reserved
1:0	RW	0x0	gpio3c_smt GPIO schmitt trigger control, every GPIO bit corresponding to 1bits . 0: No hysteresis 1: Schmitt trigger enabled.

GRF_GPIO3D_SMT
Address: Operational Base + offset (0x0e0dc)
GPIO3D smitter control register

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:8	RO	0x0	reserved

Bit	Attr	Reset Value	Description
7:0	RW	0x00	gpio3d_smt GPIO schmitt trigger control, every GPIO bit corresponding to 1bits . 0: No hysteresis 1: Schmitt trigger enabled.

GRF_GPIO4A_SMT
Address: Operational Base + offset (0x0e0e0)
GPIO4A smitter control register

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:8	RO	0x0	reserved
7:0	RW	0x00	gpio4a_smt GPIO schmitt trigger control, every GPIO bit corresponding to 1bits . 0: No hysteresis 1: Schmitt trigger enabled.

GRF_GPIO4B_SMT
Address: Operational Base + offset (0x0e0e4)
GPIO4B smitter control register

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:6	RO	0x0	reserved
5:0	RW	0x3f	gpio4b_smt GPIO schmitt trigger control, every GPIO bit corresponding to 1bits . 0: No hysteresis 1: Schmitt trigger enabled.

GRF_GPIO4C_SMT
Address: Operational Base + offset (0x0e0e8)
GPIO4C smitter control register

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:8	RO	0x0	reserved

Bit	Attr	Reset Value	Description
7:0	RW	0x00	gpio4c_smt GPIO schmitt trigger control, every GPIO bit corresponding to 1bits . 0: No hysteresis 1: Schmitt trigger enabled.

GRF_GPIO4D_SMT
Address: Operational Base + offset (0x0e0ec)
GPIO4D smitter control register

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:7	RO	0x0	reserved
6:0	RW	0x00	gpio4d_smt GPIO schmitt trigger control, every GPIO bit corresponding to 1bits . 0: No hysteresis 1: Schmitt trigger enabled.

GRF_GPIO2A_E
Address: Operational Base + offset (0x0e100)
GPIO2A drive strength control

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:14	RW	0x0	gpio2a7_e GPIO2A7 drive strength control, every GPIO bit corresponding to 2bits 2'b00: 3mA 2'b01: 6mA 2'b10: 9mA 2'b11: 12mA
13:12	RW	0x0	gpio2a6_e GPIO2A6 drive strength control, every GPIO bit corresponding to 2bits 2'b00: 3mA 2'b01: 6mA 2'b10: 9mA 2'b11: 12mA
11:10	RW	0x0	gpio2a5_e GPIO2A5 drive strength control, every GPIO bit corresponding to 2bits 2'b00: 3mA 2'b01: 6mA 2'b10: 9mA 2'b11: 12mA
9:8	RW	0x0	gpio2a4_e GPIO2A4 drive strength control, every GPIO bit corresponding to 2bits 2'b00: 3mA 2'b01: 6mA 2'b10: 9mA 2'b11: 12mA

Bit	Attr	Reset Value	Description
7:6	RW	0x0	gpio2a3_e GPIO2A3 drive strength control, every GPIO bit corresponding to 2bits 2'b00: 3mA 2'b01: 6mA 2'b10: 9mA 2'b11: 12mA
5:4	RW	0x0	gpio2a2_e GPIO2A2 drive strength control, every GPIO bit corresponding to 2bits 2'b00: 3mA 2'b01: 6mA 2'b10: 9mA 2'b11: 12mA
3:2	RW	0x0	gpio2a1_e GPIO2A1 drive strength control, every GPIO bit corresponding to 2bits 2'b00: 3mA 2'b01: 6mA 2'b10: 9mA 2'b11: 12mA
1:0	RW	0x0	gpio2a0_e GPIO2A0 drive strength control, every GPIO bit corresponding to 2bits 2'b00: 3mA 2'b01: 6mA 2'b10: 9mA 2'b11: 12mA

GRF_GPIO2B_E
Address: Operational Base + offset (0x0e104)
GPIO2B drive strength control

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:14	RW	0x0	gpio2b7_e GPIO2B7 drive strength control, every GPIO bit corresponding to 2bits 2'b00: 3mA 2'b01: 6mA 2'b10: 9mA 2'b11: 12mA
13:12	RW	0x0	gpio2b6_e GPIO2B6 drive strength control, every GPIO bit corresponding to 2bits 2'b00: 3mA 2'b01: 6mA 2'b10: 9mA 2'b11: 12mA
11:10	RW	0x0	gpio2b5_e GPIO2B5 drive strength control, every GPIO bit corresponding to 2bits 2'b00: 3mA 2'b01: 6mA 2'b10: 9mA 2'b11: 12mA
9:8	RW	0x0	gpio2b4_e GPIO2B4 drive strength control, every GPIO bit corresponding to 2bits 2'b00: 3mA 2'b01: 6mA 2'b10: 9mA 2'b11: 12mA

Bit	Attr	Reset Value	Description
7:6	RW	0x0	gpio2b3_e GPIO2B3 drive strength control, every GPIO bit corresponding to 2bits 2'b00: 3mA 2'b01: 6mA 2'b10: 9mA 2'b11: 12mA
5:4	RW	0x0	gpio2b2_e GPIO2B2 drive strength control, every GPIO bit corresponding to 2bits 2'b00: 3mA 2'b01: 6mA 2'b10: 9mA 2'b11: 12mA
3:2	RW	0x0	gpio2b1_e GPIO2B1 drive strength control, every GPIO bit corresponding to 2bits 2'b00: 3mA 2'b01: 6mA 2'b10: 9mA 2'b11: 12mA
1:0	RW	0x0	gpio2b0_e GPIO2B0 drive strength control, every GPIO bit corresponding to 2bits 2'b00: 3mA 2'b01: 6mA 2'b10: 9mA 2'b11: 12mA

GRF_GPIO2C_E
Address: Operational Base + offset (0x0e108)
GPIO2C drive strength control

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:14	RW	0x0	gpio2c7_e GPIO2C7 drive strength control, every GPIO bit corresponding to 2bits 2'b00: 5mA 2'b01: 10mA 2'b10: 15mA 2'b11: 20mA
13:12	RW	0x0	gpio2c6_e GPIO2C6 drive strength control, every GPIO bit corresponding to 2bits 2'b00: 5mA 2'b01: 10mA 2'b10: 15mA 2'b11: 20mA
11:10	RW	0x0	gpio2c5_e GPIO2C5 drive strength control, every GPIO bit corresponding to 2bits 2'b00: 5mA 2'b01: 10mA 2'b10: 15mA 2'b11: 20mA
9:8	RW	0x0	gpio2c4_e GPIO2C4 drive strength control, every GPIO bit corresponding to 2bits 2'b00: 5mA 2'b01: 10mA 2'b10: 15mA 2'b11: 20mA

Bit	Attr	Reset Value	Description
7:6	RW	0x0	gpio2c3_e GPIO2C3 drive strength control, every GPIO bit corresponding to 2bits 2'b00: 5mA 2'b01: 10mA 2'b10: 15mA 2'b11: 20mA
5:4	RW	0x0	gpio2c2_e GPIO2C2 drive strength control, every GPIO bit corresponding to 2bits 2'b00: 5mA 2'b01: 10mA 2'b10: 15mA 2'b11: 20mA
3:2	RW	0x0	gpio2c1_e GPIO2C1 drive strength control, every GPIO bit corresponding to 2bits 2'b00: 5mA 2'b01: 10mA 2'b10: 15mA 2'b11: 20mA
1:0	RW	0x0	gpio2c0_e GPIO2C0 drive strength control, every GPIO bit corresponding to 2bits 2'b00: 5mA 2'b01: 10mA 2'b10: 15mA 2'b11: 20mA

GRF_GPIO2D_E
Address: Operational Base + offset (0x0e10c)
GPIO2D drive strength control

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:14	RW	0x0	gpio2d7_e GPIO2D7 drive strength control, every GPIO bit corresponding to 2bits 2'b00: 5mA 2'b01: 10mA 2'b10: 15mA 2'b11: 20mA
13:12	RW	0x0	gpio2d6_e GPIO2D6 drive strength control, every GPIO bit corresponding to 2bits 2'b00: 5mA 2'b01: 10mA 2'b10: 15mA 2'b11: 20mA
11:10	RW	0x0	gpio2d5_e GPIO2D5 drive strength control, every GPIO bit corresponding to 2bits 2'b00: 5mA 2'b01: 10mA 2'b10: 15mA 2'b11: 20mA
9:8	RW	0x0	gpio2d4_e GPIO2D4 drive strength control, every GPIO bit corresponding to 2bits 2'b00: 5mA 2'b01: 10mA 2'b10: 15mA 2'b11: 20mA

Bit	Attr	Reset Value	Description
7:6	RW	0x0	gpio2d3_e GPIO2D3 drive strength control, every GPIO bit corresponding to 2bits 2'b00: 5mA 2'b01: 10mA 2'b10: 15mA 2'b11: 20mA
5:4	RW	0x0	gpio2d2_e GPIO2D2 drive strength control, every GPIO bit corresponding to 2bits 2'b00: 5mA 2'b01: 10mA 2'b10: 15mA 2'b11: 20mA
3:2	RW	0x0	gpio2d1_e GPIO2D1 drive strength control, every GPIO bit corresponding to 2bits 2'b00: 5mA 2'b01: 10mA 2'b10: 15mA 2'b11: 20mA
1:0	RW	0x0	gpio2d0_e GPIO2D0 drive strength control, every GPIO bit corresponding to 2bits 2'b00: 5mA 2'b01: 10mA 2'b10: 15mA 2'b11: 20mA

GRF_GPIO3A_E01
Address: Operational Base + offset (0x0e110)
GPIO3A drive strength control

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15	RW	0x0	gpio3a5_e0 GPIO3A5 drive strength control bit0, bit1 and bit2 are in the GPIO3A_E2 register 3'b000:4mA; 3'b001:7mA; 3'b010:10mA; 3'b011:13mA; 3'b100:16mA; 3'b101:19mA; 3'b110:22mA; 3'b111:26mA;
14:12	RW	0x0	gpio3a4_e GPIO3A4 drive strength control bit0 to bit2 3'b000:4mA; 3'b001:7mA; 3'b010:10mA; 3'b011:13mA; 3'b100:16mA; 3'b101:19mA; 3'b110:22mA; 3'b111:26mA;
11:9	RW	0x0	gpio3a3_e GPIO3A3 drive strength control bit0 to bit2 3'b000:4mA; 3'b001:7mA; 3'b010:10mA; 3'b011:13mA; 3'b100:16mA; 3'b101:19mA; 3'b110:22mA; 3'b111:26mA;

Bit	Attr	Reset Value	Description
8:6	RW	0x0	gpio3a2_e GPIO3A2 drive strength control bit0 to bit2 3'b000:4mA; 3'b001:7mA; 3'b010:10mA; 3'b011:13mA; 3'b100:16mA; 3'b101:19mA; 3'b110:22mA; 3'b111:26mA;
5:3	RW	0x0	gpio3a1_e GPIO3A1 drive strength control bit0 to bit2 3'b000:4mA; 3'b001:7mA; 3'b010:10mA; 3'b011:13mA; 3'b100:16mA; 3'b101:19mA; 3'b110:22mA; 3'b111:26mA;
2:0	RW	0x0	gpio3a0_e GPIO3A0 drive strength control bit0 to bit2 3'b000:4mA; 3'b001:7mA; 3'b010:10mA; 3'b011:13mA; 3'b100:16mA; 3'b101:19mA; 3'b110:22mA; 3'b111:26mA;

GRF_GPIO3A_E2
Address: Operational Base + offset (0x0e114)
GPIO3B drive strength control

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:8	RO	0x0	reserved
7:5	RW	0x0	gpio3a7_e GPIO3A7 drive strength control bit0 to bit2 3'b000:4mA; 3'b001:7mA; 3'b010:10mA; 3'b011:13mA; 3'b100:16mA; 3'b101:19mA; 3'b110:22mA; 3'b111:26mA;
4:2	RW	0x0	gpio3a6_e GPIO3A6 drive strength control bit0 to bit2 3'b000:4mA; 3'b001:7mA; 3'b010:10mA; 3'b011:13mA; 3'b100:16mA; 3'b101:19mA; 3'b110:22mA; 3'b111:26mA;

Bit	Attr	Reset Value	Description
1:0	RW	0x0	gpio3a5_e12 GPIO3A5 drive strength control bit1 and bit2, bit0 is in the GPIO3A_E01 register 3'b000:4mA; 3'b001:7mA; 3'b010:10mA; 3'b011:13mA; 3'b100:16mA; 3'b101:19mA; 3'b110:22mA; 3'b111:26mA;

GRF_GPIO3B_E01
Address: Operational Base + offset (0x0e118)
GPIO3B drive strength control

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15	RW	0x0	gpio3b5_e0 GPIO3B5 drive strength control bit0, bit1 and bit2 are in the GPIO3B_E2 register 3'b000:4mA; 3'b001:7mA; 3'b010:10mA; 3'b011:13mA; 3'b100:16mA; 3'b101:19mA; 3'b110:22mA; 3'b111:26mA;

Bit	Attr	Reset Value	Description
14:12	RW	0x0	gpio3b4_e GPIO3B4 drive strength control bit0 to bit2 3'b000:4mA; 3'b001:7mA; 3'b010:10mA; 3'b011:13mA; 3'b100:16mA; 3'b101:19mA; 3'b110:22mA; 3'b111:26mA;
11:9	RW	0x0	gpio3b3_e GPIO3B3 drive strength control bit0 to bit2 3'b000:4mA; 3'b001:7mA; 3'b010:10mA; 3'b011:13mA; 3'b100:16mA; 3'b101:19mA; 3'b110:22mA; 3'b111:26mA;
8:6	RW	0x0	gpio3b2_e GPIO3B2 drive strength control bit0 to bit2 3'b000:4mA; 3'b001:7mA; 3'b010:10mA; 3'b011:13mA; 3'b100:16mA; 3'b101:19mA; 3'b110:22mA; 3'b111:26mA;
5:3	RW	0x0	gpio3b1_e GPIO3B1 drive strength control bit0 to bit2 3'b000:4mA; 3'b001:7mA; 3'b010:10mA; 3'b011:13mA; 3'b100:16mA; 3'b101:19mA; 3'b110:22mA; 3'b111:26mA;

Bit	Attr	Reset Value	Description
2:0	RW	0x0	gpio3b0_e GPIO3B0 drive strength control bit0 to bit2 3'b000:4mA; 3'b001:7mA; 3'b010:10mA; 3'b011:13mA; 3'b100:16mA; 3'b101:19mA; 3'b110:22mA; 3'b111:26mA;

GRF_GPIO3B_E2

Address: Operational Base + offset (0x0e11c)

GPIO3B drive strength control

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:8	RO	0x0	reserved
7:5	RW	0x0	gpio3b7_e GPIO3B7 drive strength control bit0 to bit2 3'b000:4mA; 3'b001:7mA; 3'b010:10mA; 3'b011:13mA; 3'b100:16mA; 3'b101:19mA; 3'b110:22mA; 3'b111:26mA;

Bit	Attr	Reset Value	Description
4:2	RW	0x0	gpio3b6_e GPIO3B6 drive strength control bit0 to bit2 3'b000:4mA; 3'b001:7mA; 3'b010:10mA; 3'b011:13mA; 3'b100:16mA; 3'b101:19mA; 3'b110:22mA; 3'b111:26mA;
1:0	RW	0x0	gpio3b5_e12 GPIO3B5 drive strength control bit1 to bit2, bit0 is in the GPIO3B_E01 register 3'b000:4mA; 3'b001:7mA; 3'b010:10mA; 3'b011:13mA; 3'b100:16mA; 3'b101:19mA; 3'b110:22mA; 3'b111:26mA;

GRF_GPIO3C_E01
Address: Operational Base + offset (0x0e120)
GPIO3C drive strength control

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;

Bit	Attr	Reset Value	Description
15	RW	0x0	gpio3c5_e0 GPIO3C5 drive strength control bit0, bit1 and bit2 are in the GPIO3B_E01 register 3'b000:4mA; 3'b001:7mA; 3'b010:10mA; 3'b011:13mA; 3'b100:16mA; 3'b101:19mA; 3'b110:22mA; 3'b111:26mA;
14:12	RW	0x0	gpio3c4_e GPIO3C4 drive strength control bit0 to bit2 3'b000:4mA; 3'b001:7mA; 3'b010:10mA; 3'b011:13mA; 3'b100:16mA; 3'b101:19mA; 3'b110:22mA; 3'b111:26mA;
11:9	RW	0x0	gpio3c3_e GPIO3C3 drive strength control bit0 to bit2 3'b000:4mA; 3'b001:7mA; 3'b010:10mA; 3'b011:13mA; 3'b100:16mA; 3'b101:19mA; 3'b110:22mA; 3'b111:26mA;
8:6	RW	0x0	gpio3c2_e GPIO3C2 drive strength control bit0 to bit2 3'b000:4mA; 3'b001:7mA; 3'b010:10mA; 3'b011:13mA; 3'b100:16mA; 3'b101:19mA; 3'b110:22mA; 3'b111:26mA;

Bit	Attr	Reset Value	Description
5:3	RW	0x0	gpio3c1_e GPIO3C1 drive strength control bit0 to bit2 3'b000:4mA; 3'b001:7mA; 3'b010:10mA; 3'b011:13mA; 3'b100:16mA; 3'b101:19mA; 3'b110:22mA; 3'b111:26mA;
2:0	RW	0x0	gpio3c0_e GPIO3C0 drive strength control bit0 to bit2 3'b000:4mA; 3'b001:7mA; 3'b010:10mA; 3'b011:13mA; 3'b100:16mA; 3'b101:19mA; 3'b110:22mA; 3'b111:26mA;

GRF_GPIO3C_E2
Address: Operational Base + offset (0x0e124)
GPIO3C drive strength control

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:8	RO	0x0	reserved

Bit	Attr	Reset Value	Description
7:5	RW	0x0	gpio3c7_e GPIO3C7 drive strength control bit0 to bit2 3'b000:4mA; 3'b001:7mA; 3'b010:10mA; 3'b011:13mA; 3'b100:16mA; 3'b101:19mA; 3'b110:22mA; 3'b111:26mA;
4:2	RW	0x0	gpio3c6_e GPIO3C6 drive strength control bit0 to bit2 3'b000:4mA; 3'b001:7mA; 3'b010:10mA; 3'b011:13mA; 3'b100:16mA; 3'b101:19mA; 3'b110:22mA; 3'b111:26mA;
1:0	RW	0x0	gpio3c5_e12 GPIO3C5 drive strength control bit1 and bit2, bit0 is in the GPIO3C_E01 register 3'b000:4mA; 3'b001:7mA; 3'b010:10mA; 3'b011:13mA; 3'b100:16mA; 3'b101:19mA; 3'b110:22mA; 3'b111:26mA;

GRF_GPIO3D_E
Address: Operational Base + offset (0x0e128)
GPIO3D drive strength control

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:14	RW	0x0	gpio3d7_e GPIO3D7 drive strength control, every GPIO bit corresponding to 2bits 2'b00: 3mA 2'b01: 6mA 2'b10: 9mA 2'b11: 12mA
13:12	RW	0x0	gpio3d6_e GPIO3D6 drive strength control, every GPIO bit corresponding to 2bits 2'b00: 3mA 2'b01: 6mA 2'b10: 9mA 2'b11: 12mA
11:10	RW	0x0	gpio3d5_e GPIO3D5 drive strength control, every GPIO bit corresponding to 2bits 2'b00: 3mA 2'b01: 6mA 2'b10: 9mA 2'b11: 12mA
9:8	RW	0x0	gpio3d4_e GPIO3D4 drive strength control, every GPIO bit corresponding to 2bits 2'b00: 3mA 2'b01: 6mA 2'b10: 9mA 2'b11: 12mA

Bit	Attr	Reset Value	Description
7:6	RW	0x0	gpio3d3_e GPIO3D3 drive strength control, every GPIO bit corresponding to 2bits 2'b00: 3mA 2'b01: 6mA 2'b10: 9mA 2'b11: 12mA
5:4	RW	0x0	gpio3d2_e GPIO3D2 drive strength control, every GPIO bit corresponding to 2bits 2'b00: 3mA 2'b01: 6mA 2'b10: 9mA 2'b11: 12mA
3:2	RW	0x0	gpio3d1_e GPIO3D1 drive strength control, every GPIO bit corresponding to 2bits 2'b00: 3mA 2'b01: 6mA 2'b10: 9mA 2'b11: 12mA
1:0	RW	0x0	gpio3d0_e GPIO3D0 drive strength control, every GPIO bit corresponding to 2bits 2'b00: 3mA 2'b01: 6mA 2'b10: 9mA 2'b11: 12mA

GRF_GPIO4A_E
Address: Operational Base + offset (0x0e12c)
GPIO4A drive strength control

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:14	RW	0x0	gpio4a7_e GPIO4A7 drive strength control, every GPIO bit corresponding to 2bits 2'b00: 3mA 2'b01: 6mA 2'b10: 9mA 2'b11: 12mA
13:12	RW	0x0	gpio4a6_e GPIO4A6 drive strength control, every GPIO bit corresponding to 2bits 2'b00: 3mA 2'b01: 6mA 2'b10: 9mA 2'b11: 12mA
11:10	RW	0x0	gpio4a5_e GPIO4A5 drive strength control, every GPIO bit corresponding to 2bits 2'b00: 3mA 2'b01: 6mA 2'b10: 9mA 2'b11: 12mA
9:8	RW	0x0	gpio4a4_e GPIO4A4 drive strength control, every GPIO bit corresponding to 2bits 2'b00: 3mA 2'b01: 6mA 2'b10: 9mA 2'b11: 12mA

Bit	Attr	Reset Value	Description
7:6	RW	0x0	gpio4a3_e GPIO4A3 drive strength control, every GPIO bit corresponding to 2bits 2'b00: 3mA 2'b01: 6mA 2'b10: 9mA 2'b11: 12mA
5:4	RW	0x0	gpio4a2_e GPIO4A2 drive strength control, every GPIO bit corresponding to 2bits 2'b00: 3mA 2'b01: 6mA 2'b10: 9mA 2'b11: 12mA
3:2	RW	0x0	gpio4a1_e GPIO4A1 drive strength control, every GPIO bit corresponding to 2bits 2'b00: 3mA 2'b01: 6mA 2'b10: 9mA 2'b11: 12mA
1:0	RW	0x0	gpio4a0_e GPIO4A0 drive strength control, every GPIO bit corresponding to 2bits 2'b00: 3mA 2'b01: 6mA 2'b10: 9mA 2'b11: 12mA

GRF_GPIO4B_E01
Address: Operational Base + offset (0x0e130)
GPIO4B drive strength control

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15	RW	0x1	gpio4b5_e0 GPIO4B5 drive strength control bit0 3'b000:4mA; 3'b001:6mA; 3'b010:8mA; 3'b011:10mA; 3'b100:12mA; 3'b101:14mA; 3'b110:16mA; 3'b111:18mA;
14:12	RW	0x1	gpio4b4_e GPIO4B4 drive strength control bit0 to bit2 3'b000:4mA; 3'b001:6mA; 3'b010:8mA; 3'b011:10mA; 3'b100:12mA; 3'b101:14mA; 3'b110:16mA; 3'b111:18mA;
11:9	RW	0x1	gpio4b3_e GPIO4B3 drive strength control bit0 to bit2 3'b000:4mA; 3'b001:6mA; 3'b010:8mA; 3'b011:10mA; 3'b100:12mA; 3'b101:14mA; 3'b110:16mA; 3'b111:18mA;

Bit	Attr	Reset Value	Description
8:6	RW	0x1	gpio4b2_e GPIO4B2 drive strength control bit0 to bit2 3'b000:4mA; 3'b001:6mA; 3'b010:8mA; 3'b011:10mA; 3'b100:12mA; 3'b101:14mA; 3'b110:16mA; 3'b111:18mA;
5:3	RW	0x1	gpio4b1_e GPIO4B1 drive strength control bit0 to bit2 3'b000:4mA; 3'b001:6mA; 3'b010:8mA; 3'b011:10mA; 3'b100:12mA; 3'b101:14mA; 3'b110:16mA; 3'b111:18mA;
2:0	RW	0x1	gpio4b0_e GPIO4B0 drive strength control bit0 to bit2 3'b000:4mA; 3'b001:6mA; 3'b010:8mA; 3'b011:10mA; 3'b100:12mA; 3'b101:14mA; 3'b110:16mA; 3'b111:18mA;

GRF_GPIO4B_E2
Address: Operational Base + offset (0x0e134)
GPIO4B drive strength control

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:8	RO	0x0	reserved
7:5	RW	0x0	gpio4b7_e GPIO4B7 drive strength control bit0 to bit2 3'b000:4mA; 3'b001:6mA; 3'b010:8mA; 3'b011:10mA; 3'b100:12mA; 3'b101:14mA; 3'b110:16mA; 3'b111:18mA;
4:2	RW	0x0	gpio4b6_e GPIO4B6 drive strength control bit0 to bit2 3'b000:4mA; 3'b001:6mA; 3'b010:8mA; 3'b011:10mA; 3'b100:12mA; 3'b101:14mA; 3'b110:16mA; 3'b111:18mA;

Bit	Attr	Reset Value	Description
1:0	RW	0x0	gpio4b5_e12 GPIO4B5 drive strength control bit2, bit0 is in the GPIO4B_E01 register 3'b000:4mA; 3'b001:6mA; 3'b010:8mA; 3'b011:10mA; 3'b100:12mA; 3'b101:14mA; 3'b110:16mA; 3'b111:18mA;

GRF_GPIO4C_E
Address: Operational Base + offset (0x0e138)
GPIO4C drive strength control

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:14	RW	0x0	gpio4c7_e GPIO4C7 drive strength control, every GPIO bit corresponding to 2bits 2'b00: 3mA 2'b01: 6mA 2'b10: 9mA 2'b11: 12mA
13:12	RW	0x0	gpio4c6_e GPIO4C6 drive strength control, every GPIO bit corresponding to 2bits 2'b00: 3mA 2'b01: 6mA 2'b10: 9mA 2'b11: 12mA

Bit	Attr	Reset Value	Description
11:10	RW	0x0	gpio4c5_e GPIO4C5 drive strength control, every GPIO bit corresponding to 2bits 2'b00: 3mA 2'b01: 6mA 2'b10: 9mA 2'b11: 12mA
9:8	RW	0x0	gpio4c4_e GPIO4C4 drive strength control, every GPIO bit corresponding to 2bits 2'b00: 3mA 2'b01: 6mA 2'b10: 9mA 2'b11: 12mA
7:6	RW	0x0	gpio4c3_e GPIO4C3 drive strength control, every GPIO bit corresponding to 2bits 2'b00: 3mA 2'b01: 6mA 2'b10: 9mA 2'b11: 12mA
5:4	RW	0x0	gpio4c2_e GPIO4C2 drive strength control, every GPIO bit corresponding to 2bits 2'b00: 3mA 2'b01: 6mA 2'b10: 9mA 2'b11: 12mA
3:2	RW	0x0	gpio4c1_e GPIO4C1 drive strength control, every GPIO bit corresponding to 2bits 2'b00: 3mA 2'b01: 6mA 2'b10: 9mA 2'b11: 12mA
1:0	RW	0x0	gpio4c0_e GPIO4C0 drive strength control, every GPIO bit corresponding to 2bits 2'b00: 3mA 2'b01: 6mA 2'b10: 9mA 2'b11: 12mA

GRF_GPIO4D_E
Address: Operational Base + offset (0x0e13c)

GPIO4D drive strength control

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:14	RW	0x0	gpio4d7_e GPIO4D7 drive strength control, every GPIO bit corresponding to 2bits 2'b00: 3mA 2'b01: 6mA 2'b10: 9mA 2'b11: 12mA
13:12	RW	0x0	gpio4d6_e GPIO4D6 drive strength control, every GPIO bit corresponding to 2bits 2'b00: 3mA 2'b01: 6mA 2'b10: 9mA 2'b11: 12mA
11:10	RW	0x0	gpio4d5_e GPIO4D5 drive strength control, every GPIO bit corresponding to 2bits 2'b00: 3mA 2'b01: 6mA 2'b10: 9mA 2'b11: 12mA
9:8	RW	0x0	gpio4d4_e GPIO4D4 drive strength control, every GPIO bit corresponding to 2bits 2'b00: 3mA 2'b01: 6mA 2'b10: 9mA 2'b11: 12mA

Bit	Attr	Reset Value	Description
7:6	RW	0x0	gpio4d3_e GPIO4D3 drive strength control, every GPIO bit corresponding to 2bits 2'b00: 3mA 2'b01: 6mA 2'b10: 9mA 2'b11: 12mA
5:4	RW	0x0	gpio4d2_e GPIO4D2 drive strength control, every GPIO bit corresponding to 2bits 2'b00: 3mA 2'b01: 6mA 2'b10: 9mA 2'b11: 12mA
3:2	RW	0x0	gpio4d1_e GPIO4D1 drive strength control, every GPIO bit corresponding to 2bits 2'b00: 3mA 2'b01: 6mA 2'b10: 9mA 2'b11: 12mA
1:0	RW	0x0	gpio4d0_e GPIO4D0 drive strength control, every GPIO bit corresponding to 2bits 2'b00: 3mA 2'b01: 6mA 2'b10: 9mA 2'b11: 12mA

GRF_GPIO2C_HE
Address: Operational Base + offset (0x0e188)
GPIO2C HE control

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:8	RO	0x0	reserved
7:0	RW	0x00	gpio2c_he GPIO2C gpio keep privous state control, every GPIO bit corresponding to 1bit 1'b0: disable 1'b1: enable

GRF_GPIO2D_HE
Address: Operational Base + offset (0x0e18c)
GPIO2D HE control

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:5	RO	0x0	reserved

Bit	Attr	Reset Value	Description
4:0	RW	0x00	gpio2d_he GPIO2D gpio keep privous state control, every GPIO bit corresponding to 1bit 1'b0: disable 1'b1: enable

GRF_SOC_CON0
Address: Operational Base + offset (0x0e200)
SoC control register 0

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15	RW	0x0	emmc_fwd_perihp_pwrDiscTargPwrStall noc_emmc_fwd_perihp_rsp_err_stall bit control 0: error response 1: stall response
14	RW	0x0	centersrv_fwd_ccim1_pwrDiscTargPwrStall noc_centersrv_fwd_ccim1_rsp_err_stall bit control 0: error response 1: stall response
13	RW	0x0	center_fwd_vio_pwrDiscTargPwrStall noc_center_fwd_vio_rsp_err_stall bit control 0: error response 1: stall response
12	RW	0x0	center_fwd_vdu_pwrDiscTargPwrStall noc_center_fwd_vdu_rsp_err_stall bit control 0: error response 1: stall response

Bit	Attr	Reset Value	Description
11	RW	0x0	center_fwd_vcodec_pwrDiscTargPwrStall noc_center_fwd_vcodec_rsp_err_stall bit control 0: error response 1: stall response
10	RW	0x0	center_fwd_usb3_pwrDiscTargPwrStall noc_center_fwd_usb3_rsp_err_stall bit control 0: error response 1: stall response
9	RW	0x0	center_fwd_rga_pwrDiscTargPwrStall noc_center_fwd_rga_rsp_err_stall bit control 0: error response 1: stall response
8	RW	0x0	center_fwd_perihp_pwrDiscTargPwrStall noc_center_fwd_perihp_rsp_err_stall bit control 0: error response 1: stall response
7	RW	0x0	center_fwd_iep_pwrDiscTargPwrStall noc_center_fwd_iep_rsp_err_stall bit control 0: error response 1: stall response
6	RW	0x0	center_fwd_gpu_pwrDiscTargPwrStall noc_center_fwd_gpu_rsp_err_stall bit control 0: error response 1: stall response
5	RW	0x0	perilp_fwd_gmac_pwrDiscTargPwrStall perilp_fwd_gmac_rsp_err_stall bit control 0: error response 1: stall response
4	RW	0x0	perilp_fwd_emmc_pwrDiscTargPwrStall perilp_fwd_emmc_rsp_err_stall bit control 0: error response 1: stall response
3	RW	0x0	center_fwd_edp_pwrDiscTargPwrStall noc_center_fwd_edp_rsp_err_stall bit control 0: error response 1: stall response
2	RW	0x0	cci_req_msch1_pwrDiscTargPwrStall noc_cci_req_msch1_rsp_err_stall bit control 0: error response 1: stall response

Bit	Attr	Reset Value	Description
1	RW	0x0	cci_req_msch0_pwrDiscTargPwrStall noc__rsp_err_stall bit control 0: error response 1: stall response
0	RW	0x0	cci_fwd_perilp_pwrDiscTargPwrStall noc_cci_fwd_perilp_rsp_err_stall bit control 0: error response 1: stall response

GRF_SOC_CON1
Address: Operational Base + offset (0x0e204)
SoC control register 2

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15	RW	0x0	perihp_fwd_cci_pwrDiscTargPwrStall noc_perihp_fwd_cci_rsp_err_stall bit control 0: error response 1: stall response
14	RW	0x0	perihp_fwd_alive_pwrDiscTargPwrStall noc_perihp_fwd_alive_rsp_err_stall bit control 0: error response 1: stall response
13	RW	0x0	perihp_cm0_fwd_perihp_pwrDiscTargPwrStall noc_perihp_cm0_fwd_perihp_rsp_err_stall bit control 0: error response 1: stall response

Bit	Attr	Reset Value	Description
12	RW	0x0	perihp_req_msch1_pwrDiscTargPwrStall noc_perihp_req_msch1_rsp_err_stall bit control 0: error response 1: stall response
11	RW	0x0	perihp_req_msch0_pwrDiscTargPwrStall noc_perihp_req_msch0_rsp_err_stall bit control 0: error response 1: stall response
10	RW	0x0	perihp_fwd_center_pwrDiscTargPwrStall noc_perihp_fwd_center_rsp_err_stall bit control 0: error response 1: stall response
9	RW	0x0	pcie_fwd_perihp_pwrDiscTargPwrStall noc_pcie_fwd_perihp_rsp_err_stall bit control 0: error response 1: stall response
8	RW	0x0	msch1regsrv_fwd_msch1_pwrDiscTargPwrStall noc_msch1regsrv_fwd_msch1_rsp_err_stall bit control 0: error response 1: stall response
7	RW	0x0	msch0regsrv_fwd_msch0_pwrDiscTargPwrStall noc_msch0regsrv_fwd_msch0_rsp_err_stall bit control 0: error response 1: stall response
6	RW	0x0	isp1_req_msch01_pwrDiscTargPwrStall noc_isp1_req_msch01_rsp_err_stall bit control 0: error response 1: stall response
5	RW	0x0	isp0_req_msch01_pwrDiscTargPwrStall noc_isp0_req_msch01_rsp_err_stall bit control 0: error response 1: stall response

Bit	Attr	Reset Value	Description
4	RW	0x0	iep_req_msch1_pwrDiscTargPwrStall noc_iep_req_msch1_rsp_err_stall bit control 0: error response 1: stall response
3	RW	0x0	iep_req_msch0_pwrDiscTargPwrStall noc_iep_req_msch0_rsp_err_stall bit control 0: error response 1: stall response
2	RW	0x0	hdcp_req_msch01_pwrDiscTargPwrStall noc_hdcp_req_msch01_rsp_err_stall bit control 0: error response 1: stall response
1	RW	0x0	gpu_req_msch1_pwrDiscTargPwrStall noc_gpu_req_msch1_rsp_err_stall bit control 0: error response 1: stall response
0	RW	0x0	gpu_req_msch0_pwrDiscTargPwrStall noc_gpu_req_msch0_rsp_err_stall bit control 0: error response 1: stall response

GRF_SOC_CON2

Address: Operational Base + offset (0x0e208)

SoC control register 1

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Fbit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;

Bit	Attr	Reset Value	Description
15	RW	0x0	vio0_req_msch0_pwrDiscTargPwrStall noc_vio0_req_msch0_rsp_err_stall bit control 0: error response 1: stall response
14	RW	0x0	vdu_req_msch1_pwrDiscTargPwrStall noc_vdu_req_msch1_rsp_err_stall bit control 0: error response 1: stall response
13	RW	0x0	vdu_req_msch0_pwrDiscTargPwrStall noc_vdu_req_msch0_rsp_err_stall bit control 0: error response 1: stall response
12	RW	0x0	vcodec_req_msch1_pwrDiscTargPwrStall noc_vcodec_req_msch1_rsp_err_stall bit control 0: error response 1: stall response
11	RW	0x0	vcodec_req_msch0_pwrDiscTargPwrStall noc_vcodec_req_msch0_rsp_err_stall bit control 0: error response 1: stall response
10	RW	0x0	usb3_req_msch1_pwrDiscTargPwrStall noc_usb3_req_msch1_rsp_err_stall bit control 0: error response 1: stall response
9	RW	0x0	usb3_req_msch0_pwrDiscTargPwrStall noc_usb3_req_msch0_rsp_err_stall bit control 0: error response 1: stall response
8	RW	0x0	rga_req_msch1_pwrDiscTargPwrStall noc_rga_req_msch1_rsp_err_stall bit control 0: error response 1: stall response
7	RW	0x0	rga_req_msch0_pwrDiscTargPwrStall noc_rga_req_msch0_rsp_err_stall bit control 0: error response 1: stall response

Bit	Attr	Reset Value	Description
6	RW	0x0	sdioaudio_fwd_perilp_pwrDiscTargPwrStall noc_pmu_fwd_perilp_rsp_err_stall bit control 0: error response 1: stall response
5	RW	0x0	gmac_fwd_perihp_pwrDiscTargPwrStall noc_gmac_fwd_perihp_rsp_err_stall bit control 0: error response 1: stall response
4	RW	0x0	perilpsrv_fwd_cm0_pwrDiscTargPwrStall noc_perilpsrv_fwd_cm0_rsp_err_stall bit control 0: error response 1: stall response
3	RW	0x0	perilp_req_msch1_pwrDiscTargPwrStall noc_perilp_req_msch1_rsp_err_stall bit control 0: error response 1: stall response
2	RW	0x0	perilp_req_msch0_pwrDiscTargPwrStall noc_perilp_req_msch0_rsp_err_stall bit control 0: error response 1: stall response
1	RW	0x0	perilp_fwd_pmu_pwrDiscTargPwrStall noc_perilp_fwd_pmu_rsp_err_stall bit control 0: error response 1: stall response
0	RW	0x0	perilp_fwd_center_pwrDiscTargPwrStall noc_perilp_fwd_center_rsp_err_stall bit control 0: error response 1: stall response

GRF_SOC_CON3
Address: Operational Base + offset (0x0e20c)
SoC control register 3

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15	RW	0x0	perihp_fwd_sd_pwrDiscTargPwrStall noc_perihp_fwd_sd_rsp_err_stall bit control 0: error response 1: stall response
14	RW	0x0	gic_fwd_perilp_pwrDiscTargPwrStall noc_gic_fwd_perilp_rsp_err_stall bit control 0: error response 1: stall response
13	RW	0x0	sd_fwd_perihp_pwrDiscTargPwrStall noc_sd_fwd_perihp_rsp_err_stall bit control 0: error response 1: stall response
12	RW	0x0	vopl_req_msch11_pwrDiscTargPwrStall noc_vopl_req_msch11_rsp_err_stall bit control 0: error response 1: stall response
11	RW	0x0	vopb_req_msch11_pwrDiscTargPwrStall noc_vopb_req_msch11_rsp_err_stall bit control 0: error response 1: stall response
10	RW	0x0	vio_fwd_hdcp_pwrDiscTargPwrStall noc_vio_fwd_hdcp_rsp_err_stall bit control 0: error response 1: stall response
9	RW	0x0	vio_fwd_vopl_pwrDiscTargPwrStall noc_vio_fwd_vopl_rsp_err_stall bit control 0: error response 1: stall response

Bit	Attr	Reset Value	Description
8	RW	0x0	vio_fwd_vopb_pwrDiscTargPwrStall noc_vio_fwd_vopb_rsp_err_stall bit control 0: error response 1: stall response
7	RW	0x0	vio_fwd_isp1_pwrDiscTargPwrStall noc_vio_fwd_isp1_rsp_err_stall bit control 0: error response 1: stall response
6	RW	0x0	vio_fwd_isp0_pwrDiscTargPwrStall noc_vio_fwd_isp0_rsp_err_stall bit control 0: error response 1: stall response
5	RW	0x0	usb3_fwd_perilp_pwrDiscTargPwrStall noc_usb3_fwd_perilp_rsp_err_stall bit control 0: error response 1: stall response
4	RW	0x0	viol_req_msch01_pwrDiscTargPwrStall noc_viol_req_msch01_rsp_err_stall bit control 0: error response 1: stall response
3	RW	0x0	viob_req_msch01_pwrDiscTargPwrStall noc_viob_req_msch01_rsp_err_stall bit control 0: error response 1: stall response
2	RW	0x0	vio1_req_msch1_pwrDiscTargPwrStall noc_vio1_req_msch1_rsp_err_stall bit control 0: error response 1: stall response
1	RW	0x0	vio1_req_msch0_pwrDiscTargPwrStall noc_vio1_req_msch0_rsp_err_stall bit control 0: error response 1: stall response
0	RW	0x0	vio0_req_msch1_pwrDiscTargPwrStall noc_vio0_req_msch1_rsp_err_stall bit control 0: error response 1: stall response

GRF_SOC_CON4
Address: Operational Base + offset (0x0e210)

SoC control register 4

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15	RO	0x0	reserved
14	RW	0x0	perilp_fwd_centerslv_pwrDiscTargPwrStall noc_perilp_fwd_centerslv_rsp_err_stall bit control 0: error response 1: stall response
13	RW	0x0	perilp_fwd_sdioaudio_pwrDiscTargPwrStall noc_perilp_fwd_sdioaudio_rsp_err_stall bit control 0: error response 1: stall response
12	RW	0x0	perilp_fwd_gic_pwrDiscTargPwrStall noc_perilp_fwd_gic_rsp_err_stall bit control 0: error response 1: stall response
11:9	RW	0x0	ddr_debug_sel select ddr debug port 0: ddr_dbug_port[7:0] 1: ddr_dbug_port[15:8] 2: ddr_dbug_port[23:16] 3: ddr_dbug_port[31:24] 4: ddr_dbug_port[39:32] 5: ddr_dbug_port[47:40] 6: ddr_dbug_port[55:48] 7: ddr_dbug_port[63:56]
8	RW	0x1	cci_force_wakeup cci force wakeup control 1'b0: disable 1'b1: enable

Bit	Attr	Reset Value	Description
7:6	RW	0x0	cci_qosoverride cci port QOSOVERRIDE bit control
5:4	RW	0x0	cci_ordered_wr_obsv cci port ORDERED_WRITE_OBSERVATION control
3:2	RW	0x3	acchannelens1_cci500 CCI ACCHANNELEN input control. Slave interface supports DVM messages. This is overridden to 0x0 if you set the Control Override Register[1].
1:0	RW	0x3	acchannelens0_cci500 CCI ACCHANNELEN input control. Slave interface supports DVM messages. This is overridden to 0x0 if you set the Control Override Register[1].

GRF_SOC_CON_5_PCIE
Address: Operational Base + offset (0x0e214)
SoC control register 5

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:7	RO	0x0	reserved
6:3	RW	0x0	pcie_tx_elec_idle_off pcie_tx_elec_idle_off[3:0] port control
2	RW	0x0	pcie_rx_elec_idle_irq_en pcie_rx_elec_idle_irq_en port control
1	RW	0x1	pcie_tx_elec_idle_set pcie_tx_elec_idle_set port control
0	RW	0x0	pcie_tx_elec_idle_sel pcie_tx_elec_idle_sel port control

GRF_SOC_CON7
Address: Operational Base + offset (0x0e21c)
SoC control register 7

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15	RW	0x0	gic_awuser_mode gic_awuser mode select 1: address mode 0: user mode
14	RW	0x0	pcie_clkreq_sel pcie_clkreq_sel port control
13	RO	0x0	reserved
12	RW	0x1	grf_con_force_jtag
11:10	RW	0x0	grf_uart_dbg_sel
9:5	RW	0x00	grf_uart_rts_sel uart_rts_sel bit control UART polarity selection for rts port Every bit for one UART. 1'b1: invert uart_rts_n 1'b0: keep the rts_n value from UART module output
4:0	RW	0x00	grf_uart_cts_sel UART polarity selection for cts port Every bit for one UART. 1'b1: invert uart_cts_n 1'b0: keep the cts_n value from IO

GRF_SOC_CON8
Address: Operational Base + offset (0x0e220)

SoC control register 8

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:14	RO	0x0	reserved
13:11	RW	0x0	i2s0_sdio_oe_n i2s0_sdio_oe_n bit control
10:7	RW	0x0	pcie_test_i pci test input
6:1	RW	0x00	pcie_test_addr pci test address control
0	RW	0x0	pcie_test_write pcie test write control 1'b0: disable 1'b1: enable

GRF_SOC_CON_9_PCIE

Address: Operational Base + offset (0x0e224)

SoC control register 9 for PCIE

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:1	RO	0x0	reserved
0	RW	0x0	pcie_rc_mode_idle_irq_clr irq clear bit for pcie_rc_mode_idle_irq

GRF_SOC_STATUS0

Address: Operational Base + offset (0x0e2a0)

SOC status register 0

Bit	Attr	Reset Value	Description
31:25	RO	0x0	reserved
24	RW	0x0	ddr1_mem_rst_valid status bit of ddr1_mem_rst_valid
23	RW	0x0	ddr1_q_almost_full status bit of ddr1_q_almost_full
22	RW	0x0	ddr1_refresh_in_process status bit of ddr1_refresh_in_process
21	RW	0x0	ddr1_controller_busy status bit of ddr1_controller_busy
20	RW	0x0	ddr1_port_busy status bit of ddr1_port_busy
19:18	RW	0x0	ddr1_zq_status_out status bit of ddr1_zq_status_out
17:16	RW	0x0	ddr1_cke_status status bit of ddr1_cke_status
15:9	RO	0x0	reserved
8	RW	0x0	ddr0_mem_rst_valid status bit of ddr0_mem_rst_valid
7	RW	0x0	ddr0_q_almost_full status bit of ddr0_q_almost_full

Bit	Attr	Reset Value	Description
6	RW	0x0	ddr0_refresh_in_process status bit of ddr0_refresh_in_process
5	RW	0x0	ddr0_controller_busy status bit of ddr0_controller_busy
4	RW	0x0	ddr0_port_busy status bit of ddr0_port_busy
3:2	RW	0x0	ddr0_zq_status_out status bit of ddr0_zq_status_out
1:0	RW	0x3	ddr0_cke_status status bit of ddr0_cke_status

GRF_SOC_STATUS1
Address: Operational Base + offset (0x0e2a4)
SOC status register 1

Bit	Attr	Reset Value	Description
31:12	RO	0x0	reserved
11:8	RW	0x0	grf_pcie_test_o status bit of grf_pcie_test_o
7:0	RW	0x00	dphy_rx0_testdout status bit of dphy_rx0_testdout

GRF_SOC_STATUS2
Address: Operational Base + offset (0x0e2a8)
SOC status register 2

Bit	Attr	Reset Value	Description
31:11	RO	0x0	reserved
10	RW	0x0	m0_perilp_cdbgpwrupreq m0_perilp_cdbgpwrupreq status bit
9	RW	0x0	m0_perilp_sysresetreq m0_perilp_sysresetreq
8	RW	0x0	jtagnsw_st status bit of jtagnsw_st
7	RW	0x0	jtagtop_st status bit of jtagtop_st
6	RW	0x0	txev_m0_perilp status bit of xev_m0_perilp
5	RW	0x0	m0_perilp_dbgrestarted status bit of m0_perilp_dbgrestarted
4	RW	0x0	m0_perilp_halted status bit of m0_perilp_halted
3	RW	0x0	m0_perilp_core_lockup status bit of m0_perilp_core_lockup
2	RW	0x0	m0_perilp_sleepdeep status bit of m0_perilp_sleeping

Bit	Attr	Reset Value	Description
1	RW	0x0	m0_perilp_sleeping status bit of m0_perilp_sleeping
0	RW	0x0	m0_perilp_wakeup status bit of m0_perilp_wakeup

GRF_SOC_STATUS3
Address: Operational Base + offset (0x0e2ac)
SOC status register 3

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27	RO	0x0	usbcphy1_host_utmi_hostdisconnect usbcphy1_host_utmi_hostdisconnect status
26:25	RO	0x0	usbcphy1_host_utmi_linestate usbcphy1_host_utmi_linestate status
24	RO	0x0	usbcphy1_host_utmi_fs_xver_own 1: ohci owns usb2phy 0: ehci owns usb2phy
23	RO	0x0	usbcphy0_host_utmi_hostdisconnect usbcphy0_host_utmi_hostdisconnect status
22:21	RO	0x0	usbcphy0_host_utmi_linestate usbcphy0_host_utmi_linestate status
20	RO	0x0	usbcphy0_host_utmi_fs_xver_own 1: ohci owns usb2phy 0: ehci owns usb2phy
19	RO	0x0	usbcphy1_otg_utmi_hostdisconnect usbcphy1_otg_utmi_hostdisconnect status
18:17	RO	0x0	usbcphy1_otg_utmi_linestate usbcphy1_otg_utmi_linestate status bit
16	RO	0x0	usbcphy1_otg_utmi_bvalid usbcphy1_otg_utmi_bvalid status bit
15	RO	0x0	usbcphy0_otg_utmi_hostdisconnect usbcphy0_otg_utmi_hostdisconnect status
14:13	RO	0x0	usbcphy0_otg_utmi_linestate usbcphy0_otg_utmi_linestate status bit
12	RO	0x0	usbcphy0_otg_utmi_bvalid usbcphy0_otg_utmi_bvalid status bit
11	RW	0x0	usbcphy1_otg_utmi_iddig usbcphy1_otg_utmi_iddig status bit
10	RW	0x0	usbcphy1_otg_utmi_avalid usbcphy1_otg_utmi_avalid status bit
9	RW	0x0	usbcphy1_otg_utmi_sessend usbcphy1_otg_utmi_sessend status bit
8	RW	0x0	usbcphy0_otg_utmi_iddig usbcphy0_otg_utmi_iddig status bit

Bit	Attr	Reset Value	Description
7	RW	0x0	usbcphy0_otg_utmi_avalid usbcphy0_otg_utmi_avalid status bit
6	RW	0x0	usbcphy0_otg_utmi_sessend usbcphy0_otg_utmi_sessend status bit
5	RW	0x0	usb20_phy1_stat_cp_detected usb20_phy1_stat_cp_detected status bit
4	RW	0x0	usb20_phy1_stat_dcp_detected usb20_phy1_stat_dcp_detected status bit
3	RW	0x0	usb20_phy1_stat_dp_attached usb20_phy1_stat_dp_attached status bit
2	RW	0x0	usb20_phy0_stat_cp_detected usb20_phy0_stat_cp_detected status bit
1	RW	0x0	usb20_phy0_stat_dcp_detected usb20_phy0_stat_dcp_detected status bit
0	RW	0x0	usb20_phy0_stat_dp_attached usb20_phy0_stat_dp_attached status bit

GRF_SOC_STATUS4
Address: Operational Base + offset (0x0e2b0)
SOC status register 4

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	ddr_monitor ddr_monitor[31:0] status bit

GRF_SOC_STATUS5
Address: Operational Base + offset (0x0e2b4)
SOC status register 5

Bit	Attr	Reset Value	Description
31	RO	0x0	reserved
30:0	RW	0x00000000	ddr_monitor ddr_monitor[62:32] status bit

GRF_DDRC0_CON0
Address: Operational Base + offset (0x0e380)
ddrc0 control register 0

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:13	RO	0x0	reserved
12	RW	0x1	ddr0_oe_polarity bit control of ddr0_oe_polarity
11	RW	0x1	ddr0_dram_clk_enable_polarity bit control of ddr0_dram_clk_enable_polarity
10	RW	0x1	ddr0_io_ctrl_oe_polarity bit control of ddr0_io_ctrl_ie_polarity
9	RW	0x1	ddr0_io_ctrl_ie_polarity bit control of ddr0_io_ctrl_ie_polarity
8	RW	0x1	ddr0_ie_polarity bit control of ddr0_ie_polarity
7	RW	0x1	ddr0_tsel_en_polarity bit control of ddr0_tsel_en_polarity
6:3	RO	0x0	reserved
2	RW	0x0	ddr0_lp4_addr_dup bit conrol of ddr0_lp4_addr_dup
1:0	RW	0x1	ddr0_zq_status_in bit control of ddr0_zq_status_in

GRF_DDRC0_CON1
Address: Operational Base + offset (0x0e384)
ddrc0 control register 1

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:10	RO	0x0	reserved
9:8	RW	0x0	denali0_command_priority bit control of denali0_command_priority
7	RW	0x0	clk_ddr0_msch_en_stdby bit control of clk_ddr0_msch_en_stdby
6	RW	0x0	clk_ddrphy0_en_stdby bit control of clk_ddrphy0_en_stdby
5	RW	0x0	clk_ddrphy_ctrl0_en_stdby bit control of clk_ddrc0_en_stdby
4	RW	0x0	clk_ddrc0_en_stdby bit control of clk_ddrc0_en_stdby
3:0	RO	0x0	reserved

GRF_DDRC1_CON0
Address: Operational Base + offset (0x0e388)
ddrc1 control register 0

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:13	RO	0x0	reserved
12	RW	0x1	ddr1_oe_polarity bit control of ddr1_oe_polarity
11	RW	0x1	ddr1_dram_clk_enable_polarity bit control of ddr1_dram_clk_enable_polarity
10	RW	0x1	ddr1_io_ctrl_oe_polarity bit control of ddr1_io_ctrl_oe_polarity
9	RW	0x1	ddr1_io_ctrl_ie_polarity bit control of ddr1_io_ctrl_ie_polarity
8	RW	0x1	ddr1_ie_polarity bit control of ddr1_ie_polarity
7	RW	0x1	ddr1_tsel_en_polarity bit control of ddr1_tsel_en_polarity
6:3	RO	0x0	reserved
2	RW	0x0	ddr1_lp4_addr_dup bit control of ddr1_lp4_addr_dup
1:0	RW	0x1	ddr1_zq_status_in bit control of ddr1_zq_status_in

GRF_DDRC1_CON1
Address: Operational Base + offset (0x0e38c)
ddrc1 control register 1

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:10	RO	0x0	reserved
9:8	RW	0x0	denali1_command_priority bit control of denali1_command_priority
7	RW	0x0	clk_ddr1_msch_en_stdby bit control of clk_ddr1_msch_en_stdby
6	RW	0x0	clk_ddrphy1_en_stdby bit control of clk_ddrphy1_en_stdby
5	RW	0x0	clk_ddrphy_ctrl1_en_stdby bit control of clk_ddrphy_ctrl1_en_stdby
4	RW	0x0	clk_ddrc1_en_stdby bit control of clk_ddrc1_en_stdby
3:0	RO	0x0	reserved

GRF_SIG_DETECT_CON0
Address: Operational Base + offset (0x0e3c0)
Singal detect control register0

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15	RW	0x0	uphy1_rxdet_en uphy1_rxdet phy conrol bit
14	RW	0x0	uphy0_rxdet_en uphy0_rxdet phy conrol bit
13	RW	0x0	uphy1_rxdet_change uphy1_rxdet_change detect control 1'b0: disbale 1'b1: enable
12	RW	0x0	uphy0_rxdet_change uphy0_rxdet_change detect control 1'b0: disbale 1'b1: enable
11	RW	0x0	cphy1_host_linestate_change cphy1_host_linestate_change detect control 1'b0: disbale 1'b1: enable
10	RW	0x0	cphy1_otg_id_fall cphy1_otg_id_fall detect control 1'b0: disbale 1'b1: enable
9	RW	0x0	cphy1_otg_id_rise cphy1_otg_id_rise detect control 1'b0: disbale 1'b1: enable
8	RW	0x0	cphy1_otg_bvalid_rise cphy1_otg_bvalid_rise detect control 1'b0: disbale 1'b1: enable

Bit	Attr	Reset Value	Description
7	RW	0x0	cphy1_otg_linestate_change cphy1_otg_linestate_change detect control 1'b0: disbale 1'b1: enable
6	RW	0x0	cphy0_host_linestate_change cphy0_host_linestate_change detect control 1'b0: disbale 1'b1: enable
5	RW	0x0	cphy0_otg_id_fall cphy0_otg_id_fall detect control 1'b0: disbale 1'b1: enable
4	RW	0x0	cphy0_otg_id_rise cphy0_otg_id_rise detect control 1'b0: disbale 1'b1: enable
3	RW	0x0	cphy0_otg_bvalid_rise cphy0_otg_bvalid_rise detect control 1'b0: disbale 1'b1: enable
2	RW	0x0	cphy0_otg_linestate_change cphy0_otg_linestate_change detect control 1'b0: disbale 1'b1: enable
1	RW	0x0	sdmmc_card_fall_edge sdmmc card fall edge detect control 1'b0: disbale 1'b1: enable
0	RW	0x0	sdmmc_card_rise_edge sdmmc card rise edge detect control 1'b0: disbale 1'b1: enable

GRF_SIG_DETECT_CON1
Address: Operational Base + offset (0x0e3c8)
Singal detect control register1

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:14	RO	0x0	reserved
13:12	RW	0x0	host0_llinestate_filter_time_sel filter time select 00: 100us 01: 500us 10: 1ms 11: 10ms
11:10	RW	0x0	otg0_llinestate_filter_time_sel filter time select 00: 100us 01: 500us 10: 1ms 11: 10ms
9:8	RW	0x0	otg0_id_filter_time_sel filter time select 00: 5ms 01: 15ms 10: 35ms 11: 50ms
7:0	RO	0x0	reserved

GRF_SIG_DETECT_CLR
Address: Operational Base + offset (0x0e3d0)
Signal detect status clear register

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:14	RO	0x0	reserved
13	RW	0x0	uphy1_rxdet_change uphy1_rxdet_change detect control 1'b0: disbale 1'b1: enable
12	RW	0x0	uphy0_rxdet_change uphy0_rxdet_change detect control 1'b0: disbale 1'b1: enable
11	RW	0x0	cphy1_host_linestate_change cphy1_host_linestate_change detect control 1'b0: disbale 1'b1: enable
10	RW	0x0	cphy1_otg_id_fall cphy1_otg_id_fall detect control 1'b0: disbale 1'b1: enable
9	RW	0x0	cphy1_otg_id_rise cphy1_otg_id_rise detect control 1'b0: disbale 1'b1: enable
8	RW	0x0	cphy1_otg_bvalid_rise cphy1_otg_bvalid_rise detect control 1'b0: disbale 1'b1: enable
7	RW	0x0	cphy1_otg_linestate_change cphy1_otg_linestate_change detect control 1'b0: disbale 1'b1: enable

Bit	Attr	Reset Value	Description
6	RW	0x0	cphy0_host_linestate_change cphy0_host_linestate_change detect control 1'b0: disbale 1'b1: enable
5	RW	0x0	cphy0_otg_id_fall cphy0_otg_id_fall detect control 1'b0: disbale 1'b1: enable
4	RW	0x0	cphy0_otg_id_rise cphy0_otg_id_rise detect control 1'b0: disbale 1'b1: enable
3	RW	0x0	cphy0_otg_bvalid_rise cphy0_otg_bvalid_rise detect control 1'b0: disbale 1'b1: enable
2	RW	0x0	cphy0_otg_linestate_change cphy0_otg_linestate_change detect control 1'b0: disbale 1'b1: enable
1	RW	0x0	sdmmc_card_fall_edge sdmmc card fall edge detect control 1'b0: disbale 1'b1: enable
0	RW	0x0	sdmmc_card_rise_edge sdmmc card rise edge detect control 1'b0: disbale 1'b1: enable

GRF_SIG_DETECT_STATUS
Address: Operational Base + offset (0x0e3e0)
Signal detect status register

Bit	Attr	Reset Value	Description
31:14	RO	0x0	reserved
13	RW	0x0	uphy1_rxdet_change uphy1_rxdet_change detect status
12	RW	0x0	uphy0_rxdet_change uphy0_rxdet_change detect status
11	RW	0x0	cphy1_host_linestate_change cphy1_host_linestate_change detect status
10	RW	0x0	cphy1_otg_id_fall cphy1_otg_id_fall detect status
9	RW	0x0	cphy1_otg_id_rise cphy1_otg_id_rise detect status

Bit	Attr	Reset Value	Description
8	RW	0x0	cphy1_otg_bvalid_rise cphy1_otg_bvalid_rise detect status
7	RW	0x0	cphy1_otg_linestate_change cphy1_otg_linestate_change detect status
6	RW	0x0	cphy0_host_linestate_change cphy0_host_linestate_change detect status
5	RW	0x0	cphy0_otg_id_fall cphy0_otg_id_fall detect status
4	RW	0x0	cphy0_otg_id_rise cphy0_otg_id_rise detect status
3	RW	0x0	cphy0_otg_bvalid_rise cphy0_otg_bvalid_rise detect status
2	RW	0x0	cphy0_otg_linestate_change cphy0_otg_linestate_change detect status
1	RW	0x0	sdmmc_card_fall_edge sdmmc card fall edge detect status
0	RW	0x0	sdmmc_card_rise_edge sdmmc card rise edge detect status

GRF_USB20_PHY0_CON0
Address: Operational Base + offset (0x0e450)
USB20 PHY0 GRF Register 0

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:13	RO	0x0	reserved
12	RW	0x0	vdm_src_en vdm_src_en 1: enable vdm_src for battery charge for usb3otg0

Bit	Attr	Reset Value	Description
11	RW	0x0	vdp_src_en vdp_src_en 1:enable vdp_src for battery charge for usb3otg0
10	RW	0x0	rdm_pdwn_en rdm_pdwn_en 1: enable rdm_pdwn for battery charge for usb3otg0
9	RW	0x0	idp_src_en idp_src_en 1: enable idp_src for battery charge for usb3otg0
8	RW	0x0	idm_sink_en idm_sink_en 1: enable idm_sink for battery charge for usb3otg0
7	RW	0x0	idp_sink_en idp_sink_en 1: enable idp_sink for battery charge for usb3otg0
6:5	RO	0x0	reserved
4	RW	0x0	otg_commononn otg_commononn configure pll clock output in suspend mode
3	RW	0x0	bypasssel bypasssel 1: bypass DP/DM as uart sin/sout for usb3otg0 0: Normal USB function for usb3otg0
2	RW	0x0	bypassdmen bypassdmen 1: enable bypass uart_sout to DM for usb3otg0 0: disable bypass uart_sout to DM for usb3otg0
1	RW	0x0	otg_disable_1 otg_disable_1 1:disable otg function of usb20 host0 0:enable otg function of usb20 host0
0	RW	0x0	otg_disable_0 otg_disable_0 1:disable otg function of usb3otg0 0:enable otg function of usb3otg0

GRF_USB20_PHY0_CON1

Address: Operational Base + offset (0x0e454)
USB20 PHY0 GRF Register 1

Bit	Attr	Reset Value	Description
31:29	RO	0x0	reserved
28:13	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
12	RW	0x1	suspend_n_sel1 suspend_n_sel1 Pls see suspend_n.
11	RW	0x0	suspend_n_sel suspend_n_sel Pls see suspend_n.
10	RW	0x1	iddig iddig Select the value of this register to usb3otg0 register
9	RW	0x0	iddig_sel iddig_sel 1: select the value of bit10 of USB20_PHY0_CON1 to usb3otg0 controller 0: select the iddig from usb2phy to usb3otg0 controller
8	RW	0x0	dmpulldown dmpulldown Select the value of this register to usb2phy when utmi_sel=1
7	RW	0x0	dppulldown dppulldown Select the value of this register to usb2phy when utmi_sel=1
6	RW	0x1	termselect termselect Select the value of this register to usb2phy when utmi_sel=1

Bit	Attr	Reset Value	Description
5:4	RW	0x1	xcvrselect xcvrselect Select the value of this register to usb2phy when utmi_sel=1
3:2	RW	0x0	opmode opmode Select the value of this register to usb2phy when utmi_sel=1
1	RW	0x1	suspend_n suspend_n utmi_sel=1, select the value of this register to usb2phy utmi_sel=0 and bit11 of USB20_PHY0_CON1=0, and bit 12 of USB20_PHY0_CON1=0 select the value of the value of this bit to usb2phy utmi_sel=0 and bit11 of USB20_PHY0_CON1=0, and bit 12 of USB20_PHY0_CON1=1 select suspend_n signals from usb3otg0 controller to usb2phy for free running utmi clock utmi_sel=0 and bit11 of USB20_PHY0_CON1=1, select suspend_com_n signals from usb3otg0 controller to usb2phy for not free running utmi clock
0	RW	0x0	utmi_sel utmi_sel 1: select utmi interface signals from GRF reister to usb2phy 0: select utmi interface signals from utmi interface of usb3otg0 controller to usb2phy

GRF_USB20_PHY0_CON2
Address: Operational Base + offset (0x0e458)
USB20 PHY0 GRF Register 2

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:10	RO	0x0	reserved
9	RW	0x1	idpullup idpullup Use the value of this register input to idpullup of usb2phy
8	RW	0x1	dmpulldown dmpulldown Use the value of this register input to dmpulldown of usb2phy
7	RW	0x1	dppulldown dppulldown Use the value of this register input to dppulldown of usb2phy
6	RW	0x1	termselect termselect Select the value of this register to usb2phy when utmi_sel=1
5:4	RW	0x1	xcvrselect xcvrselect Select the value of this register to usb2phy when utmi_sel=1
3:2	RW	0x0	opmode opmode Select the value of this register to usb2phy when utmi_sel=1
1	RW	0x1	suspend_n suspend_n Select the value of this register to usb2phy when utmi_sel=1

Bit	Attr	Reset Value	Description
0	RW	0x0	utmi_sel utmi_sel 1: select utmi interface signals from GRF reister to usb2phy 0: select utmi interface signals from utmi interface of usb20 host0 controller to usb2phy

GRF_USB20_PHY0_CON3
Address: Operational Base + offset (0x0e45c)
USB20 PHY0 GRF Register 3

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:5	RO	0x0	reserved
4	RW	0x0	dischrgvbus dischrgvbus Use the value of this register input to chrgvbus of usb2phy
3	RW	0x0	chrgvbus chrgvbus Use the value of this register input to chrgvbus of usb2phy
2	RW	0x0	drvvbus drvvbus Pls see drvvbus_sel.

Bit	Attr	Reset Value	Description
1	RW	0x0	drvvbus_sel drvvbus_sel 1: select the value of bit2 of USB20_PHY0_CON3 to drvvbus of usb2phy and GPIO to external PMIC 0: select drvvbus from usb3otg0 controller to drvvbus of usb2phy and GPIO to external PMIC
0	RW	0x1	idpullup idpullup Use the value of this register input to idpullup of usb2phy

GRF_USB20_PHY1_CON0
Address: Operational Base + offset (0x0e460)
USB20 PHY1 GRF Register 0

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:13	RO	0x0	reserved
12	RW	0x0	vdm_src_en vdm_src_en 1: enable vdm_src for battery charge for usb3otg1
11	RW	0x0	vdp_src_en vdp_src_en 1:enable vdp_src for battery charge for usb3otg1
10	RW	0x0	rdm_pdwn_en rdm_pdwn_en 1: enable rdm_pdwn for battery charge for usb3otg1

Bit	Attr	Reset Value	Description
9	RW	0x0	idp_src_en idp_src_en 1: enable idp_src for battery charge for usb3otg1
8	RW	0x0	idm_sink_en idm_sink_en 1: enable idm_sink for battery charge for usb3otg1
7	RW	0x0	idp_sink_en idp_sink_en 1: enable idp_sink for battery charge for usb3otg1
6:5	RO	0x0	reserved
4	RW	0x0	otg_commononn otg_commononn configure pll clock output in suspend mode
3	RW	0x0	bypasssel bypasssel 1: bypass DP/DM as uart sin/sout for usb3otg1 0: Normal USB function for usb3otg1
2	RW	0x0	bypassdmen bypassdmen 1: enable bypass uart_sout to DM for usb3otg1 0: disable bypass uart_sout to DM for usb3otg1
1	RW	0x0	otg_disable_1 otg_disable_1 1:disable otg function of usb2 host1 0:enable otg function of usb2 host1
0	RW	0x0	otg_disable_0 otg_disable_0 1:disable otg function of usb3otg1 0:enable otg function of usb3otg1

GRF_USB20_PHY1_CON1

Address: Operational Base + offset (0x0e464)

USB20 PHY1GRF Register 1

Bit	Attr	Reset Value	Description
31:29	RO	0x0	reserved

Bit	Attr	Reset Value	Description
28:13	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
12	RW	0x1	suspend_n_sel1 suspend_n-_sel1 Pls see suspend_n.
11	RW	0x0	suspend_n_sel suspend_n_sel Pls see suspend_n.
10	RW	0x1	iddig iddig Select the value of this register to usb3otg1 register
9	RW	0x0	iddig_sel iddig_sel 1: select the value of bit10 of USB20_PHY0_CON1 to usb3otg1 controller 0: select the iddig from usb2phy to usb3otg1 controller
8	RW	0x0	dmpulldown dmpulldown Select the value of this register to usb2phy when utmi_sel=1
7	RW	0x0	dppulldown dppulldown Select the value of this register to usb2phy when utmi_sel=1
6	RW	0x1	termselect termselect Select the value of this register to usb2phy when utmi_sel=1

Bit	Attr	Reset Value	Description
5:4	RW	0x1	xcvrselect xcvrselect Select the value of this register to usb2phy when utmi_sel=1
3:2	RW	0x0	opmode opmode Select the value of this register to usb2phy when utmi_sel=1
1	RW	0x1	suspend_n suspend_n utmi_sel=1, select the value of this register to usb2phy utmi_sel=0 and bit11 of USB20_PHY0_CON1=0, and bit 12 of USB20_PHY0_CON1=0 select the value of the value of this bit to usb2phy utmi_sel=0 and bit11 of USB20_PHY0_CON1=0, and bit 12 of USB20_PHY0_CON1=1 select suspend_n signals from usb3otg0 controller to usb2phy for free running utmi clock utmi_sel=0 and bit11 of USB20_PHY0_CON1=1, select suspend_com_n signals from usb3otg0 controller to usb2phy for not free running utmi clock
0	RW	0x0	utmi_sel utmi_sel 1: select utmi interface signals from GRF reister to usb2phy 0: select utmi interface signals from utmi interface of usb3otg1 controller to usb2phy

GRF_USB20_PHY1_CON2
Address: Operational Base + offset (0x0e468)
USB20 PHY1 GRF Register 2

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:10	RO	0x0	reserved
9	RW	0x1	idpullup idpullup Use the value of this register input to idpullup of usb2phy
8	RW	0x1	dmpulldown dmpulldown Use the value of this register input to dmpulldown of usb2phy
7	RW	0x1	dppulldown dppulldown Use the value of this register input to dppulldown of usb2phy
6	RW	0x1	termselect termselect Select the value of this register to usb2phy when utmi_sel=1
5:4	RW	0x1	xcvrselect xcvrselect Select the value of this register to usb2phy when utmi_sel=1
3:2	RW	0x0	opmode opmode Select the value of this register to usb2phy when utmi_sel=1
1	RW	0x1	suspend_n suspend_n Select the value of this register to usb2phy when utmi_sel=1

Bit	Attr	Reset Value	Description
0	RW	0x0	utmi_sel utmi_sel 1: select utmi interface signals from GRF reister to usb2phy 0: select utmi interface signals from utmi interface of usb20 host1 controller to usb2phy

GRF_USB20_PHY1_CON3
Address: Operational Base + offset (0x0e46c)
USB20 PHY1 GRF Register 3

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:5	RO	0x0	reserved
4	RW	0x0	dischrgvbus dischrgvbus Use the value of this register input to chrgvbus of usb2phy
3	RW	0x0	chrgvbus chrgvbus Use the value of this register input to chrgvbus of usb2phy
2	RW	0x0	drvvbus drvvbus Pls see drvvbus_sel.

Bit	Attr	Reset Value	Description
1	RW	0x0	drvvbus_sel drvvbus_sel 1: select the value of bit2 of USB20_PHY1_CON3 to drvvbus of usb2phy and GPIO to external PMIC 0: select drvvbus from usb3otg1 controller to drvvbus of usb2phy and GPIO to external PMIC
0	RW	0x1	idpullup idpullup Use the value of this register input to idpullup of usb2phy

GRF_USB3PHY0_CON0
Address: Operational Base + offset (0x0e580)
TypeC PHY/TCPD PHY/TCPC Control register0

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:14	RW	0x2	vbus_src_sel vbus source select to IOMUX 0: select vbus_source_en of TCPC0 to IOMUX 1: select vbus_source_en of TCPC1 to IOMUX 2: select host0_drvvbus, host1_drvvbus, otg0_drvvbus, otg1_drvvbus to IOMUX
13	RO	0x0	reserved
12	RW	0x1	cc2_overcurrent_n cc2 overcurrent 0: cc2 overcurrent 1: cc2 not overcurrent

Bit	Attr	Reset Value	Description
11	RW	0x1	cc1_overcurrent_n cc1 overcurrent 0: cc1 overcurrent 1: cc1 not overcurrent
10	RW	0x0	vbus_valid_sel vbus valid select 0: use bvalid from usb2phy to usb3 controller 1: usb vbus_valid from TCPC to usb3 controller
9	RW	0x0	tcpc_vbus_on TCPC Vbus On 0: disable TCPC vbus supply 1: enable TCPC vbus supply
8	RW	0x1	typec_conn_dir_sel TypeC connect direction select 0: select typec_conn_dir (bit0 of this register) to TypeC PHY 1: select TCPC ouput typec_con_dir to TypeC PHY
7	RW	0x1	dead_battery_n dead_battery_n 1: no dead battery 0: dead battery
6	RW	0x1	dead_battery_sel dead_battery_sel 0: select external dead_battery_n from IOMUX 1: select internal bit7 of this register
5:4	RW	0x0	tcpc_role_strap TCPC role trap 01: TCPC default as DFP 10: TCPC default as UFP 11: TCPC default as DRP
3	RW	0x1	usb3tousb2_en force usb3 to usb2 enable control 1: force usb3 controller work as usb2. 0: not force usb3 controller work as usb2.
2:1	RW	0x0	pipe_data_bus_width Pipe interface data bus width 0: 32bit data bus width, only support 32bit data bus width.

Bit	Attr	Reset Value	Description
0	RW	0x0	typec_conn_dir TypeC PHY connect direction 0: normal orientation 1: flip orientation

GRF_USB3PHY0_CON1

Address: Operational Base + offset (0x0e584)

TypeC PHY/TCPD PHY/TCPC Control register1

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15	RW	0x0	JTAG_select JTAG_select 0: select TDO of TCPC0 to IOMUX 1: select TDO of TCPC1 to IOMUX
14:13	RO	0x0	reserved
12	RW	0x1	vbus_overvoltage_n vbus overvoltage 0: vbus over voltage 1: vbus not over voltage
11	RW	0x0	JTRST JTRST TCPC extensa core JTAG JTRST reset control
10	RW	0x0	DReset DReset TCPC extensa core JTAG DReset
9	RW	0x0	BRreset BRreset TCPC extensa core JTAG BRreset
8	RW	0x0	OCDHaltOnReset OCDHaltOnReset TCPC extensa core JTAG OCDHaltOnReset

Bit	Attr	Reset Value	Description
7:6	RO	0x0	reserved
5	RW	0x0	txdetectrxloopbk txdetectrxloopbk pipe_sel=1, select this bit to TypeC PHY
4:3	RW	0x0	powerdown powerdown pipe_sel: select this two bit to TypeC PHY
2	RW	0x0	txelecidle txelecidle pipe_sel=1, select this bit to TypeC PHY
1	RW	0x0	rxtermination rx termination pipe_sel=1, select this bit to TypeC PHY
0	RW	0x0	pipe_sel pipe interface select 0: select pipe interface from usb3otg 1: select pipe interface from grf controller register

GRF_USB3PHY0_CON2
Address: Operational Base + offset (0x0e588)
TypeC PHY/TCPD PHY/TCPC Control register2

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:14	RO	0x0	reserved
13:10	RW	0xf	vbus_overcurrent_n vbus source overcurrent 0: vbus source over current 1: vbus source not over current

Bit	Attr	Reset Value	Description
9:0	RW	0x0c8	vbus_voltage TCPC vbus voltage TCPC vbus voltage

GRF_USB3PHY1_CON0

Address: Operational Base + offset (0x0e58c)

TypeC PHY/TCPD PHY/TCPC Control register0

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:13	RO	0x0	reserved
12	RW	0x1	cc2_overcurrent_n cc2 overcurrent 0: cc2 overcurrent 1: cc2 not overcurrent
11	RW	0x1	cc1_overcurrent_n cc1 overcurrent 0: cc1 overcurrent 1: cc1 not overcurrent
10	RW	0x0	vbus_valid_sel vbus valid select 0: use bvalid from usb2phy to usb3 controller 1: usb vbus_valid from TCPC to usb3 controller
9	RW	0x0	tcpc_vbus_on TCPC Vbus On 0: disable TCPC vbus supply 1: enable TCPC vbus supply

Bit	Attr	Reset Value	Description
8	RW	0x1	typec_conn_dir_sel TypeC connect direction select 0: select typec_conn_dir (bit0 of this register) to TypeC PHY 1: select TCPC ouput typec_con_dir to TypeC PHY
7	RW	0x1	dead_battery_n dead_battery_n 1: no dead battery 0: dead battery
6	RW	0x1	dead_battery_sel dead_battery_sel 0: select external dead_battery_n from IOMUX 1: select internal bit7 of this register
5:4	RW	0x0	tcpc_role_strap TCPC role trap 01: TCPC default as DFP 10: TCPC default as UFP 11: TCPC default as DRP
3	RW	0x1	usb3tousb2_en force usb3 to usb2 enable control 1: force usb3 controller work as usb2. 0: not force usb3 controller work as usb2.
2:1	RW	0x0	pipe_data_bus_width Pipe interface data bus width 0: 32bit data bus width, only support 32bit data bus width.
0	RW	0x0	typec_conn_dir TypeC PHY connect direction 0: normal orientation 1: flip orientation

GRF_USB3PHY1_CON1
Address: Operational Base + offset (0x0e590)
TypeC PHY/TCPD PHY/TCPC Control register1

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:13	RO	0x0	reserved
12	RW	0x1	vbus_overvoltage_n vbus overvoltage 0: vbus over voltage 1: vbus not over voltage
11	RW	0x0	JTRST JTRST TCPC extensa core JTAG JTRST reset control
10	RW	0x0	DReset DReset TCPC extensa core JTAG DReset
9	RW	0x0	BRreset BRreset TCPC extensa core JTAG BRreset
8	RW	0x0	OCDHaltOnReset OCDHaltOnReset TCPC extensa core JTAG OCDHaltOnReset
7:6	RO	0x0	reserved
5	RW	0x0	txdetectrxloopbk txdetectrxloopbk pipe_sel=1, select this bit to TypeC PHY
4:3	RW	0x0	powerdown powerdown pipe_sel: select this two bit to TypeC PHY
2	RW	0x0	txelecidle txelecidle pipe_sel=1, select this bit to TypeC PHY
1	RW	0x0	rxtermination rx termination pipe_sel=1, select this bit to TypeC PHY

Bit	Attr	Reset Value	Description
0	RW	0x0	pipe_sel pipe interface select 0: select pipe interface from usb3otg 1: select pipe interface from grf controller register

GRF_USB3PHY1_CON2
Address: Operational Base + offset (0x0e594)
TypeC PHY/TCPD PHY/TCPC Control register2

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:14	RO	0x0	reserved
13:10	RW	0xf	vbus_overcurrent_n vbus source overcurrent 0: vbus source over current 1: vbus source not over current
9:0	RW	0x0c8	vbus_voltage TCPC vbus voltage TCPC vbus voltage

GRF_USB3PHY_STATUS0
Address: Operational Base + offset (0x0e5c0)
USB3PHY_STATUS0

Bit	Attr	Reset Value	Description
31:29	RO	0x0	reserved
28:25	RO	0x0	TCPC1_vbus_source_en TCPC1_vbus_source_en 1: select corresponding vbus source
24	RW	0x0	TCPC1_sink_en TCPC1_sink_en 1: TCPC1 enable to sink vbus

Bit	Attr	Reset Value	Description
23	RW	0x0	TCPC1_bdis_en TCPC1_bdis_en 1: TCPC1 bleed discharge enable
22	RW	0x0	TCPC1_fdis_en TCPC1_fdis_en 1: TCPC1 force discharge enable
21	RO	0x0	TCPC1_vconn_to_cc2 TCPC1 supply VCONN to CC2 1: TCPC1 supply VCONN to CC2
20	RO	0x0	TCPC1_vconn_to_cc1 TCPC1 vconn supply to CC1 1: support voconn to CC1
19	RO	0x0	TCPC1_vbus_overcurrent TCPC1 vbus overcurrent ouput 1: vbus over current
18	RO	0x0	TCPC1_JTAG_XOCDMode TCPC1 JTAG XOCDMode
17	RO	0x0	typec_pd_phy1_ready TypeC PD PHY 1 ready 1: TypeC PD PHY ready
16	RO	0x0	typec_phy1_pipe_status TypeC PHY 0 pipe status 0: indicate TypeC PHY pipe ready after release TypeC PHY pipe reset.
15:13	RO	0x0	reserved
12:9	RO	0x0	TCPC0_vbus_source_en TCPC0_vbus_source_en 1: select corresponding vbus source
8	RO	0x0	TCPC0_sink_en TCPC0_sink_en 1: TCPC0 enable to sink vbus
7	RO	0x0	TCPC0_bdis_en TCPC0_bdis_en 1: TCPC0 bleed discharge enable
6	RO	0x0	TCPC0_fdis_en TCPC0_fdis_en 1: TCPC0 force discharge enable
5	RO	0x0	TCPC0_vconn_to_cc2 TCPC0 supply VCONN to CC2 1: TCPC0 supply VCONN to CC2
4	RO	0x0	TCPC0_vconn_to_cc1 TCPC0 vconn supply to CC1 1: support voconn to CC1

Bit	Attr	Reset Value	Description
3	RO	0x0	TCPC0_vbus_overcurrent TCPC0 vbus overcurrent ouput 1: vbus over current
2	RO	0x0	TCPC0_JTAG_XOCDMode TCPC0 JTAG XOCDMode
1	RO	0x0	typec_pd_phy0_ready TypeC PD PHY 0 ready 1: TypeC PD PHY ready
0	RO	0x0	typec_phy0_pipe_status TypeC PHY 0 pipe status 0: indicate TypeC PHY pipe ready after release TypeC PHY pipe reset.

GRF_USB3PHY_STATUS1
Address: Operational Base + offset (0x0e5c4)
USB3PHY_STATUS1

Bit	Attr	Reset Value	Description
31	RO	0x0	cc_dead_battery_n CC dead battery indicator from IOMUX 0: dead battery happen 1: No dead battery happen
30:28	RO	0x0	reserved
27	RW	0x0	TCPC1_vbus_overcurrent_en TCPC vbus over current enable 1: enable
26	RO	0x0	TCPC1_vconn_overcurrent_en TCPC vconn overcurrent enable 1: enable
25	RO	0x0	TCPC1_vbus_voltage_en TCPC vbus voltage enable 1: enable
24	RO	0x0	TCPC1_vbus_overvoltage_en TCPC vbus overvoltage enable 1: enable
23	RO	0x0	TCPC1_outs_to_hiz TCPC outs to hiz
22	RO	0x0	TCPC1_dbg_acc_conn_n TCPC debug accessory connect 0: Debug accessory connected 1: No debug accessory connected

Bit	Attr	Reset Value	Description
21	RO	0x0	TCPC1_audio_acc_conn_n TCPC audio accessory connect 0: audio accessory connected 1: No audio accessory connected
20	RO	0x0	TCPC1_act_cable_conn_n TCPC active cable connect 0: No connected 1: Active cable connected
19:18	RO	0x0	TCPC1_mux_ctrl TCPC MUX CTRL 0: No connect 1: USB3.1 connect 2: DP 4 lanes 3: USB3.1 and DP 2 lanes
17	RO	0x0	TCPC1_conn_present TCPC connect present 0: No connect 1: Connected
16	RO	0x0	TCPC1_conn_orientation TCPC connect orientation 0: normal 1: flip
15:12	RO	0x0	reserved
11	RW	0x0	TCPC0_vbus_overcurrent_en TCPC vbus over current enable 1: enable
10	RO	0x0	TCPC0_vconn_overcurrent_en TCPC vconn overcurrent enable 1: enable
9	RO	0x0	TCPC0_vbus_voltage_en TCPC vbus voltage enable 1: enable
8	RO	0x0	TCPC0_vbus_overvoltage_en TCPC vbus overvoltage enable 1: enable
7	RO	0x0	TCPC0_outs_to_hiz TCPC outs to hiz
6	RO	0x0	TCPC0_dbg_acc_conn_n TCPC debug accessory connect 0: Debug accessory connected 1: No debug accessory connected

Bit	Attr	Reset Value	Description
5	RO	0x0	TCPC0_audio_acc_conn_n TCPC audio accessory connect 0: audio accessory connected 1: No audio accessory connected
4	RO	0x0	TCPC0_act_cable_conn_n TCPC active cable connect 0: No connected 1: Active cable connected
3:2	RO	0x0	TCPC0_mux_ctrl TCPC MUX CTRL 0: No connect 1: USB3.1 connect 2: DP 4 lanes 3: USB3.1 and DP 2 lanes
1	RO	0x0	TCPC0_conn_present TCPC connect present 0: No connect 1: Connected
0	RO	0x0	TCPC0_conn_orientation TCPC connect orientation 0: normal 1: flip

GRF_DLL_CON0
Address: Operational Base + offset (0x0e600)
pvtm control register

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:14	RW	0x0	pvtm_gpu_osc_ring_sel gpu PVT monitor oscilator ring select

Bit	Attr	Reset Value	Description
13	RW	0x0	pvtm_gpu_osc_en gpu PVT monitor oscilator enable 1'b1: enable 1'b0: disable
12	RW	0x0	pvtm_gpu_start gpu PVT monitor start control
11:10	RW	0x0	pvtm_ddr_osc_ring_sel ddr PVT monitor oscilator ring select
9	RW	0x0	pvtm_ddr_osc_en ddr PVT monitor oscilator enable 1'b1: enable 1'b0: disable
8	RW	0x0	pvtm_ddr_start ddr PVT monitor start control
7:6	RW	0x0	pvtm_core_b_osc_sel pd_core_l PVT monitor oscilator select pvtm_core_b_osc_sel[1:0]
5	RW	0x0	pvtm_core_b_osc_en pd_core_b PVT monitor oscilator enable 1'b1: enable 1'b0: disable
4	RW	0x0	pvtm_core_b_start pd_core_b PVT monitor start control
3:2	RW	0x0	pvtm_core_l_osc_sel pd_core_l PVT monitor oscilator select
1	RW	0x0	pvtm_core_l_osc_en pd_core_l PVT monitor oscilator enable 1'b1: enable 1'b0: disable
0	RW	0x0	pvtm_core_l_start pd_core_l PVT monitor start control

GRF_DLL_CON1
Address: Operational Base + offset (0x0e604)
pvtm control register

Bit	Attr	Reset Value	Description
31:0	RW	0x016e3600	pvtm_core_l_cal_cnt pd_core_l pvtm calculator counter

GRF_DLL_CON2
Address: Operational Base + offset (0x0e608)
pvtm control register

Bit	Attr	Reset Value	Description
31:0	RW	0x016e3600	pvtm_core_b_cal_cnt pd_core_b pvtm calculator counter

GRF_DLL_CON3
Address: Operational Base + offset (0x0e60c)
pvtm control register

Bit	Attr	Reset Value	Description
31:0	RW	0x016e3600	pvtm_ddr_cal_cnt ddr pvtm calculator counter

GRF_DLL_CON4
Address: Operational Base + offset (0x0e610)
pvtm control register

Bit	Attr	Reset Value	Description
31:0	RW	0x016e3600	pvtm_gpu_cal_cnt gpu pvtm calculator counter

GRF_DLL_CON5
Address: Operational Base + offset (0x0e614)
pvtm control register

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:1	RO	0x0	reserved
0	RW	0x0	pvtm_core_b_osc_sel pvtm_core_b_osc_sel[2]

GRF_DLL_STATUS0
Address: Operational Base + offset (0x0e620)
pvtm status register

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:4	RO	0x0	reserved
3	RW	0x0	pvtm_ddr_freq_done ddr pvtm frequency calculate done stutus
2	RW	0x0	pvtm_gpu_freq_done gpu pvtm frequency calculate done stutus
1	RW	0x0	pvtm_core_b_freq_done pd_core_b pvtm frequency calculate done stutus
0	RW	0x0	pvtm_core_l_freq_done pd_core_l pvtm frequency calculate done stutus

GRF_DLL_STATUS1
Address: Operational Base + offset (0x0e624)
pvtm status register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	pvtm_core_l_freq_cnt pd_core_l pvtm frequency count

GRF_DLL_STATUS2
Address: Operational Base + offset (0x0e628)
pvtm status register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	pvtm_core_b_freq_cnt pd_core_b pvtm frequency count

GRF_DLL_STATUS3
Address: Operational Base + offset (0x0e62c)
pvtm status register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	pvtm_gpu_freq_cnt gpu pvtm frequency count

GRF_DLL_STATUS4
Address: Operational Base + offset (0x0e630)
pvtm status register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	pvtm_ddr_freq_cnt ddr pvtm frequency count

GRF_IO_VSEL
Address: Operational Base + offset (0x0e640)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:4	RO	0x0	reserved
3	RW	0x0	gpio1830_gpio4cd_ms 1'b0: IO domain works as 3.0V; 1'b1: IO domain works as 1.8V;
2	RW	0x0	sdmmc_gpio4b_ms 1'b0: IO domain works as 3.0V; 1'b1: IO domain works as 1.8V;
1	RW	0x0	audio_gpio3d4a_ms 1'b0: IO domain works as3.0V; 1'b1: IO domain works as 1.8V;
0	RW	0x0	bt656_gpio2ab_ms 1'b0: IO domain works as 3.0V; 1'b1: IO domain works as 1.8V;

GRF_SARADC_TESTBIT
Address: Operational Base + offset (0x0e644)

saradc test bit control register

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:1	RO	0x0	reserved
0	RW	0x0	saradc_testbit saradc test bit control

GRF_TSADC_TESTBIT_L

Address: Operational Base + offset (0x0e648)

saradc test bit control register

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:4	RO	0x0	reserved
3	RW	0x0	grf_tsadc_dig_bypass
2	RW	0x0	grf_tsadc_clk_sel

Bit	Attr	Reset Value	Description
1	RW	0x0	grf_tsadc_tsen_pd_1
0	RW	0x0	grf_tsadc_tsen_pd_0

GRF_TSADC_TESTBIT_H
Address: Operational Base + offset (0x0e64c)
tsadc test bit control register

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:1	RO	0x0	reserved
0	RW	0x0	tsadc_testbit_h tsadc test bit control

GRF_CHIP_ID_ADDR
Address: Operational Base + offset (0x0e800)
chip id register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	chip_id 3399

GRF_FAST_BOOT_ADDR
Address: Operational Base + offset (0x0e880)
faster boot address register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	fast_boot_addr fast boot address

GRF_EMMCCORE_CON0
Address: Operational Base + offset (0x0f000)
emmc core control register

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x0000	emmccore_con0 emmc controller control register 0. Please refer to emmc chapter for detail description.

GRF_EMMCCORE_CON1
Address: Operational Base + offset (0x0f004)
emmc core control register

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x0000	emmccore_con1 emmc controller control register 1. Please refer to emmc chapter for detail description.

GRF_EMMCCORE_CON2
Address: Operational Base + offset (0x0f008)
emmc core control register

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x0000	emmccore_con2 emmc controller control register 2. Please refer to emmc chapter for detail description.

GRF_EMMCCORE_CON3
Address: Operational Base + offset (0x0f00c)
emmc core control register

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x0000	emmccore_con3 emmc controller control register 3. Please refer to emmc chapter for detail description.

GRF_EMMCCORE_CON4
Address: Operational Base + offset (0x0f010)
emmc core control register

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x0000	emmccore_con4 emmc controller control register 4. Please refer to emmc chapter for detail description.

GRF_EMMCCORE_CON5
Address: Operational Base + offset (0x0f014)
emmc core control register

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x0000	emmccore_con5 emmc controller control register 5. Please refer to emmc chapter for detail description.

GRF_EMMCCORE_CON6
Address: Operational Base + offset (0x0f018)
emmc core control register

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x0000	emmccore_con6 emmc controller control register 6. Please refer to emmc chapter for detail description.

GRF_EMMCCORE_CON7
Address: Operational Base + offset (0x0f01c)
emmc core control register

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x0000	emmccore_con7 emmc controller control register 7. Please refer to emmc chapter for detail description.

GRF_EMMCCORE_CON8
Address: Operational Base + offset (0x0f020)
emmc core control register

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x0000	emmccore_con8 emmc controller control register 8. Please refer to emmc chapter for detail description.

GRF_EMMCCORE_CON9
Address: Operational Base + offset (0x0f024)
emmc core control register

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x0000	emmccore_con9 emmc controller control register 9. Please refer to emmc chapter for detail description.

GRF_EMMCCORE_CON10
Address: Operational Base + offset (0x0f028)
emmc core control register

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x0000	emmccore_con10 emmc controller control register 10. Please refer to emmc chapter for detail description.

GRF_EMMCCORE_CON11
Address: Operational Base + offset (0x0f02c)
emmc core control register

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x0000	emmccore_con11 emmc controller control register 11. Please refer to emmc chapter for detail description.

GRF_EMMCCORE_STATUS0
Address: Operational Base + offset (0x0f040)
emmc core status register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	emmccore_status0 emmc controller status register 0. Please refer to emmc chapter for detail description.

GRF_EMMCCORE_STATUS1
Address: Operational Base + offset (0x0f044)
emmc core status register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	emmccore_status1 emmc controller status register 1. Please refer to emmc chapter for detail description.

GRF_EMMCCORE_STATUS2
Address: Operational Base + offset (0x0f048)
emmc core status register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	emmccore_status2 emmc controller status register 2. Please refer to emmc chapter for detail description.

GRF_EMMCCORE_STATUS3
Address: Operational Base + offset (0x0f04c)
emmc core status register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	emmccore_status3 emmc controller status register 3. Please refer to emmc chapter for detail description.

GRF_EMMCPHY_CON0
Address: Operational Base + offset (0x0f780)
emmc phy control register

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x0000	emmcphy_con0 emmc phy control register 0. Please refer to emmc chapter for detail description.

GRF_EMMCPHY_CON1
Address: Operational Base + offset (0x0f784)
emmc phy control register

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x0000	emmcphy_con1 emmc phy control register 1. Please refer to emmc chapter for detail description.

GRF_EMMCPHY_CON2
Address: Operational Base + offset (0x0f788)
emmc phy control register

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x0000	emmcphy_con2 emmc phy control register 2. Please refer to emmc chapter for detail description.

GRF_EMMCPHY_CON3
Address: Operational Base + offset (0x0f78c)
emmc phy control register

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x0000	emmcphy_con3 emmc phy control register 3. Please refer to emmc chapter for detail description.

GRF_EMMCPHY_CON4
Address: Operational Base + offset (0x0f790)
emmc phy control register

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x0000	emmcphy_con4 emmc phy control register 4. Please refer to emmc chapter for detail description.

GRF_EMMCPHY_CON5
Address: Operational Base + offset (0x0f794)
emmc phy control register

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x0000	emmcphy_con5 emmc phy control register 5. Please refer to emmc chapter for detail description.

GRF_EMMCPHY_CON6
Address: Operational Base + offset (0x0f798)
emmc phy control register

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	emmcphy_con6 emmc phy control register 5. Please refer to emmc chapter for detail description.

GRF_EMMCPHY_STATUS
Address: Operational Base + offset (0x0f7a0)
emmc phy status register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	emmcphy_status emmc phy status register. Please refer to emmc chapter for detail description.

4.4 PMU GRF Register description

4.4.1 Register Summary

Name	Offset	Size	Reset Value	Description
PMUGRF_GPIO0A_IOMUX	0x00000	W	0x00000000	GPIO0A iomux control
PMUGRF_GPIO0B_IOMUX	0x00004	W	0x00000014	GPIO0B iomux control
PMUGRF_GPIO1A_IOMUX	0x00010	W	0x00000000	GPIO1A iomux control
PMUGRF_GPIO1B_IOMUX	0x00014	W	0x00000000	GPIO1B iomux control
PMUGRF_GPIO1C_IOMUX	0x00018	W	0x00000000	GPIO1C iomux control
PMUGRF_GPIO1D_IOMUX	0x0001c	W	0x00000000	GPIO1D iomux control
PMUGRF_GPIO0A_P	0x00040	W	0x0000dd5f	GPIO0A PU/PD control
PMUGRF_GPIO0B_P	0x00044	W	0x00000557	GPIO0B PU/PD control
PMUGRF_GPIO1A_P	0x00050	W	0x00006aaa	GPIO1A PU/PD control
PMUGRF_GPIO1B_P	0x00054	W	0x00006955	GPIO1B PU/PD control
PMUGRF_GPIO1C_P	0x00058	W	0x0000a599	GPIO1C PU/PD control
PMUGRF_GPIO1D_P	0x0005c	W	0x00000002	GPIO0D PU/PD control
PMUGRF_GPIO0A_E	0x00080	W	0x00000000	GPIO0A drive strength control
PMUGRF_GPIO0B_E	0x00088	W	0x00000000	GPIO0D drive strength control
PMUGRF_GPIO1A_E	0x000a0	W	0x00004000	GPIO1A drive strength control
PMUGRF_GPIO1B_E	0x000a8	W	0x00000015	GPIO1D drive strength control

Name	Offset	Size	Reset Value	Description
PMUGRF_GPIO1C_E	0x000b0	W	0x00005000	GPIO1C drive strength control
PMUGRF_GPIO1D_E	0x000b8	W	0x00000001	GPIO1D drive strength control
PMUGRF_GPIO0L_SR	0x00100	W	0x00000000	GPIO0 A/B SR control
PMUGRF_GPIO1L_SR	0x00108	W	0x00000000	GPIO1 A/B SR control
PMUGRF_GPIO1H_SR	0x0010c	W	0x0000000f	GPIO1C/D SR control
PMUGRF_GPIO0A_SMT	0x00120	W	0x00000000	GPIO0A smit control
PMUGRF_GPIO0B_SMT	0x00124	W	0x00000000	GPIO0B smit control
PMUGRF_GPIO1A_SMT	0x00130	W	0x00000000	GPIO1A smit control
PMUGRF_GPIO1B_SMT	0x00134	W	0x00000000	GPIO1B smit control
PMUGRF_GPIO1C_SMT	0x00138	W	0x00000000	GPIO1C smit control
PMUGRF_GPIO1D_SMT	0x0013c	W	0x00000000	GPIO1D smit control
PMUGRF_GPIO0L_HE	0x00160	W	0x00000000	GPIO0 A/B HE control
PMUGRF_SOC_CON0	0x00180	W	0x00000320	SoC control register 0
PMUGRF_SOC_CON10	0x001a8	W	0x000061a8	SoC control register 10
PMUGRF_SOC_CON11	0x001ac	W	0x00000000	SoC control register 11
PMUGRF_PMUPVTM_CON0	0x00240	W	0x00000000	pmu pvtm configuration register0
PMUGRF_PMUPVTM_CON1	0x00244	W	0x00000000	pmu pvtm configuration register1
PMUGRF_PMUPVTM_STATUS0	0x00248	W	0x00000000	pmu pvtm status register
PMUGRF_PMUPVTM_STATUS1	0x0024c	W	0x00000000	pmu pvtm status register
PMUGRF_OSC_E	0x00250	W	0x00000006	OSC control register
PMUGRF_OS_REG0	0x00300	W	0x00000000	os register
PMUGRF_OS_REG1	0x00304	W	0x00000000	os register
PMUGRF_OS_REG2	0x00308	W	0x00000000	os register
PMUGRF_OS_REG3	0x0030c	W	0x00000000	os register

*Notes:Size:**B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access*

4.4.2 Detail Register Description

PMUGRF_GPIO0A_IOMUX

Address: Operational Base + offset (0x00000)

GPIO0A iomux control

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:14	RW	0x0	gpio0a7_sel GPIO0A[7] iomux select 2'b00: gpio 2'b01: sdmmc_dectn 2'b10: pmu_debug5 2'b11: reserved
13:12	RW	0x0	gpio0a6_sel GPIO0A[6] iomux select 2'b00: gpio 2'b01: pwm_3a 2'b10: pmu_debug4 2'b11: reserved
11:10	RW	0x0	gpio0a5_sel GPIO0A[5] iomux select 2'b00: gpio 2'b01: emmc_pwren 2'b10: pmu_debug3 2'b11: reserved
9:8	RW	0x0	gpio0a4_sel GPIO0A[4] iomux select 2'b00: gpio 2'b01: sdio_intn 2'b10: pmu_debug2 2'b11: reserved
7:6	RW	0x0	gpio0a3_sel GPIO0A[3] iomux select 2'b00: gpio 2'b01: sdio_wrprt 2'b10: pmu_debug1 2'b11: reserved

Bit	Attr	Reset Value	Description
5:4	RW	0x0	gpio0a2_sel GPIO0A[2] iomux select 2'b00: gpio 2'b01: wifi_26m 2'b10: pmu_debug0 2'b11: reserved
3:2	RW	0x0	gpio0a1_sel GPIO0A[1] iomux select 2'b00: gpio 2'b01: ddrio_pwroff 2'b10: tcpd_ccdben 2'b11: reserved
1:0	RW	0x0	gpio0a0_sel GPIO0A[0] iomux select 2'b00: gpio 2'b01: test_clkout0 2'b10: clk_32k 2'b11: reserved

PMUGRF_GPIO0B_IOMUX
Address: Operational Base + offset (0x00004)
GPIO0B iomux control

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:12	RO	0x0	reserved
11:10	RW	0x0	gpio0b5_sel GPIO0B[5] iomux select 2'b00: gpio 2'b01: tcpd_vbusfdis 2'b10: tcpdusb2_vbussource3 2'b11: reserved

Bit	Attr	Reset Value	Description
9:8	RW	0x0	gpio0b4_sel GPIO0B4] iomux select 2'b00: gpio 2'b01: tcpd_vbusbdis 2'b10: reserved 2'b11: reserved
7:6	RW	0x0	gpio0b3_sel GPIO0B[3] iomux select 2'b00: gpio 2'b01: reserved 2'b10: reserved 2'b11: reserved
5:4	RW	0x1	gpio0b2_sel GPIO0B[2] iomux select 2'b00: gpio 2'b01: reserved 2'b10: reserved 2'b11: reserved
3:2	RW	0x1	gpio0b1_sel GPIO0B[1] iomux select 2'b00: gpio 2'b01: pmu1830_volsel 2'b10: reserved 2'b11: reserved
1:0	RW	0x0	gpio0b0_sel GPIO0B[0] iomux select 2'b00: gpio 2'b01: sdmmc_wrprt 2'b10: pmum0_wfi 2'b11: test_clkout2

PMUGRF_GPIO1A_IOMUX
Address: Operational Base + offset (0x00010)
GPIO1A iomux control

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:14	RW	0x0	gpio1a7_sel GPIO1A[7] iomux select 2'b00: gpio 2'b01: uart4m0_sin 2'b10: spi1ec_rxd 2'b11: reserved
13:12	RW	0x0	gpio1a6_sel GPIO1A[6] iomux select 2'b00: gpio 2'b01: tsadc_int 2'b10: reserved 2'b11: reserved
11:10	RW	0x0	gpio1a5_sel GPIO1A[5] iomux select 2'b00: gpio 2'b01: ap_pwroff 2'b10: reserved 2'b11: reserved
9:8	RW	0x0	gpio1a4_sel GPIO1A[4] iomux select 2'b00: gpio 2'b01: isp0_prelighttrig 2'b10: isp1_prelighttrig 2'b11: reserved
7:6	RW	0x0	gpio1a3_sel GPIO1A[3] iomux select 2'b00: gpio 2'b01: isp0_flashtrigout 2'b10: isp1_flashtrigout 2'b11: reserved

Bit	Attr	Reset Value	Description
5:4	RW	0x0	gpio1a2_sel GPIO1A[2] iomux select 2'b00: gpio 2'b01: isp0_flashtrigin 2'b10: isp1_flashtrigin 2'b11: tcpd_cc1vconn
3:2	RW	0x0	gpio1a1_sel GPIO1A[1] iomux select 2'b00: gpio 2'b01: isp0_shuttertrig 2'b10: isp1_shuttertrig 2'b11: tcpd_cc0vconn
1:0	RW	0x0	gpio1a0_sel GPIO1A[0] iomux select 2'b00: gpio 2'b01: isp0_shutteren 2'b10: isp1_shutteren 2'b11: tcpd_vbussink

PMUGRF_GPIO1B_IOMUX
Address: Operational Base + offset (0x00014)
GPIO1B iomux control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:14	RW	0x0	gpio1b7_sel GPIO1B[7] iomux select 2'b00: gpio 2'b01: spi3pmu_rxd 2'b10: i2c0pmu_scl 2'b11: reserved

Bit	Attr	Reset Value	Description
13:12	RW	0x0	gpio1b6_sel GPIO1B[6] iomux select 2'b00: gpio 2'b01: pwm_3b 2'b10: reserved 2'b11: reserved
11:10	RW	0x0	gpio1b5_sel GPIO1B[5] iomux select 2'b00: gpio 2'b01: reserved 2'b10: reserved 2'b11: reserved
9:8	RW	0x0	gpio1b4_sel GPIO1B[4] iomux select 2'b00: gpio 2'b01: i2c4sensor_scl 2'b10: reserved 2'b11: reserved
7:6	RW	0x0	gpio1b3_sel GPIO1B[3] iomux select 2'b00: gpio 2'b01: i2c4sensor_sda 2'b10: reserved 2'b11: reserved
5:4	RW	0x0	gpio1b2_sel GPIO1B[2] iomux select 2'b00: gpio 2'b01: pmum0jtag_tms 2'b10: spi1ec_csn0 2'b11: reserved
3:2	RW	0x0	gpio1b1_sel GPIO1B[1] iomux select 2'b00: gpio 2'b01: pmum0jtag_tck 2'b10: spi1ec_clk 2'b11: reserved
1:0	RW	0x0	gpio1b0_sel GPIO1B[0] iomux select 2'b00: gpio 2'b01: uart4m0_sout 2'b10: spi1ec_txd 2'b11: reserved

PMUGRF_GPIO1C_IOMUX
Address: Operational Base + offset (0x00018)

GPIO1C iomux control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:14	RW	0x0	gpio1c7_sel GPIO1C[7] iomux select 2'b00: gpio 2'b01: tcpdusb2_vbussource1 2'b10: reserved 2'b11: reserved
13:12	RW	0x0	gpio1c6_sel GPIO1C[6] iomux select 2'b00: gpio 2'b01: tcpdusb2_vbussource0 2'b10: reserved 2'b11: reserved
11:10	RW	0x0	gpio1c5_sel GPIO1C[5] iomux select 2'b00: gpio 2'b01: i2c8dcdc_scl 2'b10: reserved 2'b11: reserved
9:8	RW	0x0	gpio1c4_sel GPIO1C[4] iomux select 2'b00: gpio 2'b01: i2c8dcdc_sda 2'b10: reserved 2'b11: reserved
7:6	RW	0x0	gpio1c3_sel GPIO1C[3] iomux select 2'b00: gpio 2'b01: pwm_2 2'b10: reserved 2'b11: reserved

Bit	Attr	Reset Value	Description
5:4	RW	0x0	gpio1c2_sel GPIO1C[2] iomux select 2'b00: gpio 2'b01: spi3pmu_csn0 2'b10: reserved 2'b11: reserved
3:2	RW	0x0	gpio1c1_sel GPIO1C[1] iomux select 2'b00: gpio 2'b01: spi3pmu_clk 2'b10: reserved 2'b11: reserved
1:0	RW	0x0	gpio1c0_sel GPIO1C[0] iomux select 2'b00: gpio 2'b01: spi3pmu_txd 2'b10: i2c0pmu_scl 2'b11: reserved

PMUGRF_GPIO1D_IOMUX
Address: Operational Base + offset (0x0001c)
GPIO1D iomux control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:2	RO	0x0	reserved
1:0	RW	0x0	gpio1d0_sel GPIO1D[0] iomux select 2'b00: gpio 2'b01: tcpdusb2_vbussource2 2'b10: reserved 2'b11: reserved

PMUGRF_GPIO0A_P
Address: Operational Base + offset (0x00040)
GPIO0A PU/PD control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0xdd5f	gpio0a_p GPIO0A PE/PS programmation section, every GPIO bit corresponding to 2bits[PS:PE] 2'b00: Z(Normal operation); 2'b01: weak 0(pull-down); 2'b10: Z(Normal operation); 2'b11: weak 1(pull-up);

PMUGRF_GPIO0B_P
Address: Operational Base + offset (0x00044)
GPIO0B PU/PD control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;

Bit	Attr	Reset Value	Description
15:0	RW	0x0557	gpio0b_p GPIO0A PE/PS programmation section, every GPIO bit corresponding to 2bits[PS:PE] 2'b00: Z(Normal operation); 2'b01: weak 0(pull-down); 2'b10: Z(Normal operation); 2'b11: weak 1(pull-up);

PMUGRF_GPIO1A_P
Address: Operational Base + offset (0x00050)
GPIO1A PU/PD control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x6aaa	gpio1a_p GPIO1A PU/PD programmation section, every GPIO bit corresponding to 2bits 2'b00: Z(Normal operaton); 2'b01: weak 1(pull-up); 2'b10: weak 0(pull-down); 2'b11: Reserved;

PMUGRF_GPIO1B_P
Address: Operational Base + offset (0x00054)
GPIO1B PU/PD control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x6955	gpio1b_p GPIO1B PU/PD programmation section, every GPIO bit corresponding to 2bits 2'b00: Z(Normal operaton); 2'b01: weak 1(pull-up); 2'b10: weak 0(pull-down); 2'b11: Reserved;

PMUGRF_GPIO1C_P
Address: Operational Base + offset (0x00058)
GPIO1C PU/PD control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;

Bit	Attr	Reset Value	Description
15:0	RW	0xa599	gpio1c_p GPIO1C PU/PD programmation section, every GPIO bit corresponding to 2bits 2'b00: Z(Normal operaton); 2'b01: weak 1(pull-up); 2'b10: weak 0(pull-down); 2'b11: Reserved;

PMUGRF_GPIO1D_P
Address: Operational Base + offset (0x0005c)
GPIO0D PU/PD control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x0002	gpio1d_p GPIO1D PU/PD programmation section, every GPIO bit corresponding to 2bits 2'b00: Z(Normal operaton); 2'b01: weak 1(pull-up); 2'b10: weak 0(pull-down); 2'b11: Reserved;

PMUGRF_GPIO0A_E
Address: Operational Base + offset (0x00080)
GPIO0A drive strength control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x0000	gpio0a_e GPIO0A drive strength control, every GPIO bit corresponding to 2bits 2'b00: 5mA 2'b01: 10mA 2'b10: 15mA 2'b11: 20mA

PMUGRF_GPIO0B_E
Address: Operational Base + offset (0x00088)
GPIO0D drive strength control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;

Bit	Attr	Reset Value	Description
15:0	RW	0x0000	gpio0b_e GPIO0B drive strength control, every GPIO bit corresponding to 2bits 2'b00: 5mA 2'b01: 10mA 2'b10: 15mA 2'b11: 20mA

PMUGRF_GPIO1A_E
Address: Operational Base + offset (0x000a0)
GPIO1A drive strength control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x4000	gpio1a_e GPIO1A drive strength control, every GPIO bit corresponding to 2bits 2'b00: 3mA 2'b01: 6mA 2'b10: 9mA 2'b11: 12mA

PMUGRF_GPIO1B_E
Address: Operational Base + offset (0x000a8)
GPIO1D drive strength control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x0015	gpio1b_e GPIO1B drive strength control, every GPIO bit corresponding to 2bits 2'b00: 3mA 2'b01: 6mA 2'b10: 9mA 2'b11: 12mA

PMUGRF_GPIO1C_E

Address: Operational Base + offset (0x000b0)

GPIO1C drive strength control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;

Bit	Attr	Reset Value	Description
15:0	RW	0x5000	gpio1c_e GPIO1C drive strength control, every GPIO bit corresponding to 2bits 2'b00: 3mA 2'b01: 6mA 2'b10: 9mA 2'b11: 12mA

PMUGRF_GPIO1D_E
Address: Operational Base + offset (0x000b8)
GPIO1D drive strength control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x0001	gpio1d_e GPIO1D drive strength control, every GPIO bit corresponding to 2bits 2'b00: 3mA 2'b01: 6mA 2'b10: 9mA 2'b11: 12mA

PMUGRF_GPIO0L_SR
Address: Operational Base + offset (0x00100)
GPIO0 A/B SR control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:8	RW	0x00	gpio0b_sr GPIO0B slew rate control for each bit 1'b0: slow (half frequency) 1'b1: fast
7:0	RW	0x00	gpio0a_sr GPIO0A slew rate control for each bit 1'b0: slow (half frequency) 1'b1: fast

PMUGRF_GPIO1L_SR
Address: Operational Base + offset (0x00108)
GPIO1 A/B SR control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;

Bit	Attr	Reset Value	Description
15:8	RW	0x00	gpio1b_sr GPIO1B slew rate control for each bit 1'b0: slow (half frequency) 1'b1: fast
7:0	RW	0x00	gpio1a_sr GPIO1A slew rate control for each bit 1'b0: slow (half frequency) 1'b1: fast

PMUGRF_GPIO1H_SR
Address: Operational Base + offset (0x0010c)
GPIO1C/D SR control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:8	RW	0x00	gpio0d_sr GPIO0D slew rate control for each bit 1'b0: slow (half frequency) 1'b1: fast
7:0	RW	0x0f	gpio1c_sr GPIO1C slew rate control for each bit 1'b0: slow (half frequency) 1'b1: fast

PMUGRF_GPIO0A_SMT
Address: Operational Base + offset (0x00120)
GPIO0A smit control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x0000	gpio0a_smt GPIO0A drive strength control, every GPIO bit corresponding to 2bits 2'b00: smit disable 2'b01: smit enable 2'b10: reserved 2'b11: reserved

PMUGRF_GPIO0B_SMT
Address: Operational Base + offset (0x00124)
GPIO0B smit control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;

Bit	Attr	Reset Value	Description
15:0	RW	0x0000	gpio0b_smt GPIO0B drive strength control, every GPIO bit corresponding to 2bits 2'b00: smit disable 2'b01: smit enable 2'b10: reserved 2'b11: reserved

PMUGRF_GPIO1A_SMT
Address: Operational Base + offset (0x00130)
GPIO1A smit control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x0000	gpio1a_smt GPIO1A drive strength control, every GPIO bit corresponding to 2bits 2'b00: smit disable 2'b01: smit enable 2'b10: reserved 2'b11: reserved

PMUGRF_GPIO1B_SMT
Address: Operational Base + offset (0x00134)
GPIO1B smit control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x0000	gpio1b_smt GPIO1B drive strength control, every GPIO bit corresponding to 2bits 2'b00: smit disable 2'b01: smit enable 2'b10: reserved 2'b11: reserved

PMUGRF_GPIO1C_SMT
Address: Operational Base + offset (0x00138)
GPIO1C smit control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;

Bit	Attr	Reset Value	Description
15:0	RW	0x0000	gpio1c_smt GPIO1C drive strength control, every GPIO bit corresponding to 2bits 2'b00: smit disable 2'b01: smit enable 2'b10: reserved 2'b11: reserved

PMUGRF_GPIO1D_SMT
Address: Operational Base + offset (0x0013c)
GPIO1D smit control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x0000	gpio1d_smt GPIO1D drive strength control, every GPIO bit corresponding to 2bits 2'b00: smit disable 2'b01: smit enable 2'b10: reserved 2'b11: reserved

PMUGRF_GPIO0L_HE
Address: Operational Base + offset (0x00160)
GPIO0 A/B HE control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:14	RO	0x0	reserved
13:8	RW	0x00	gpio0b_he GPIO0B gpio keep privous state control, every GPIO bit corresponding to 1bit 1'b0: disable 1'b1: enable
7:0	RW	0x00	gpio0a_he GPIO0A gpio keep privous state control, every GPIO bit corresponding to 1bit 1'b0: disable 1'b1: enable

PMUGRF_SOC_CON0
Address: Operational Base + offset (0x00180)
SoC control register 0

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:10	RO	0x0	reserved
9	RW	0x1	pmu1830_vol pmu IO 1.8v/3.0v select. 0: IO domain works as3.0v ; 1: IO domain works as 1.8v ;
8	RW	0x1	pmu1830_volsel pmu GPIO1 1.8v/3.0v control source select. 0: controlled by IO_GPIO0B1 ; 1: controlled by PMUGRF.SOC_CON0.pmu1830_vol
7	RO	0x0	reserved
6	RW	0x0	pclk_alive_niu_en pd_alive pclk_niu gating. 1: gating ; 0: not gating .
5	RW	0x1	pwm3_sel Use 2 optional IOs for pwm3. 0: pwm3a 1: pwm3b
4	RW	0x0	cru_pmu_pclk_gate 1: gate clock ; 0: not gate .
3	RW	0x0	pmu_noc_obsrv
2	RW	0x0	pmu_mcu_niu_obsrv

Bit	Attr	Reset Value	Description
1	RW	0x0	pmu_noc_stall When pmu noc meet illegal access, the noc will 0: error reponse 1: stall
0	RW	0x0	chip_32k_src chip 32K clock source select 0: from external 1: from internal, pvtm

PMUGRF_SOC_CON10
Address: Operational Base + offset (0x001a8)
SoC control register 10

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x61a8	sdmmc_dettime0 sdmmc_dettime[15:0]

PMUGRF_SOC_CON11
Address: Operational Base + offset (0x001ac)
SoC control register 11

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:4	RO	0x0	reserved
3:0	RW	0x0	sdmmc_dettime1 sdmmc_dettime[19:16]

PMUGRF_PMUPVTM_CON0
Address: Operational Base + offset (0x00240)
pmu pvtm configuration register0

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:2	RW	0x0000	pvtm_clkout_div clk_pvtm_out_div=clk_pvtm_out/pvtm_clkout_div
1	RW	0x0	pvtm_osc_en pmu pvtm osc enable
0	RW	0x0	pvtm_start pmu pvtm start

PMUGRF_PMUPVTM_CON1
Address: Operational Base + offset (0x00244)
pmu pvtm configuration register1

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	pvtm_core_cal_cnt pd_core pvtm calculator counter

PMUGRF_PMUPVTM_STATUS0
Address: Operational Base + offset (0x00248)
pmu pvtm status register

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	pvtm_freq_done pvtm frequency calculate done status

PMUGRF_PMUPVTM_STATUS1
Address: Operational Base + offset (0x0024c)
pmu pvtm status register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	pvtm_freq_cnt pvtm frequency count

PMUGRF_OSC_E
Address: Operational Base + offset (0x00250)
OSC control register

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved
18:16	RW	0x0	write_enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; When bit 18=1, bit 2 can be written by software . When bit 18=0, bit 2 cannot be written by software;
15:3	RO	0x0	reserved
2:0	RW	0x6	osc_e 24M OSC drive strenth

PMUGRF_OS_REG0
Address: Operational Base + offset (0x00300)

os register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	os_reg0 os register

PMUGRF_OS_REG1

Address: Operational Base + offset (0x00304)

os register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	os_reg1 os register

PMUGRF_OS_REG2

Address: Operational Base + offset (0x00308)

os register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	os_reg2 os register

PMUGRF_OS_REG3

Address: Operational Base + offset (0x0030c)

os register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	os_reg3 os register

Chapter 5 Cortex-A72

5.1 Overview

The RK3399Pro has a dual-core Cortex-A72 cluster with 1M L2 memory. Cortex-A72 processor, which is a high_performance, low-power processor that implements the ARMv8-A architecture.

The Cortex-A72 processor includes following features:

- Full implementation of the ARMv8-A architecture instruction set
- Support for both AArch32 and AArch64 Execution status.
- Support for all exception levels, EL0, EL1, EL2, and EL3, in each execution states.
- Support A32 instruction set, previously called the ARM instruction set.
- Support T32 instruction set, previously called the Thumb instruction set.
- Support A64 instruction set.
- Superscalar, variable-length, out-of-order pipeline.
- Dynamic branch prediction with Branch Target Buffer (BTB) and Global History Buffer (GHB) RAMs, a return stack, and an indirect predictor.
- 48-entry fully-associative L1 instruction Translation Lookaside Buffer (TLB) with native support for 4KB, 64KB, and 1MB page sizes.
- 32-entry fully-associative L1 data TLB with native support for 4KB, 64KB, and 1MB page sizes.
- Level 2 (L2) memory system providing cluster memory coherency, with L2 cache.
- Support advanced SIMD and Floating-point Extension for integer and floating-point vector operations.
- Support ARMv8 Cryptography Extensions.
- Support AMBA 4 ACE bus architecture.

The configuration details of little cluster and big cluster are shown in following tables

Table 5-1 CPU Configuration

Number of CPU	2
L1 I cache size	48K
L1 D cache size	32K
L2 cache size	1M
L2 data RAM output latency	3 cycles
L2 data RAM input latency	2 cycles
CPU cache protection	No
SCU L2 cache protection	No
BUS master interface	ACE
NEON and floating point support	Yes
Cryptography extension	Yes

5.2 Block Diagram

The Cortex-A72 sub system is shown in Figure 8-1. As illustrated, dual-core Cortex-A72 connects to system bus through asynchronous bridges which can handle with CDC (clock domain crossing) issue.

The Cortex-A72 is connected with system counter, which can run under a constant frequency clock, for PPI interrupt generation.

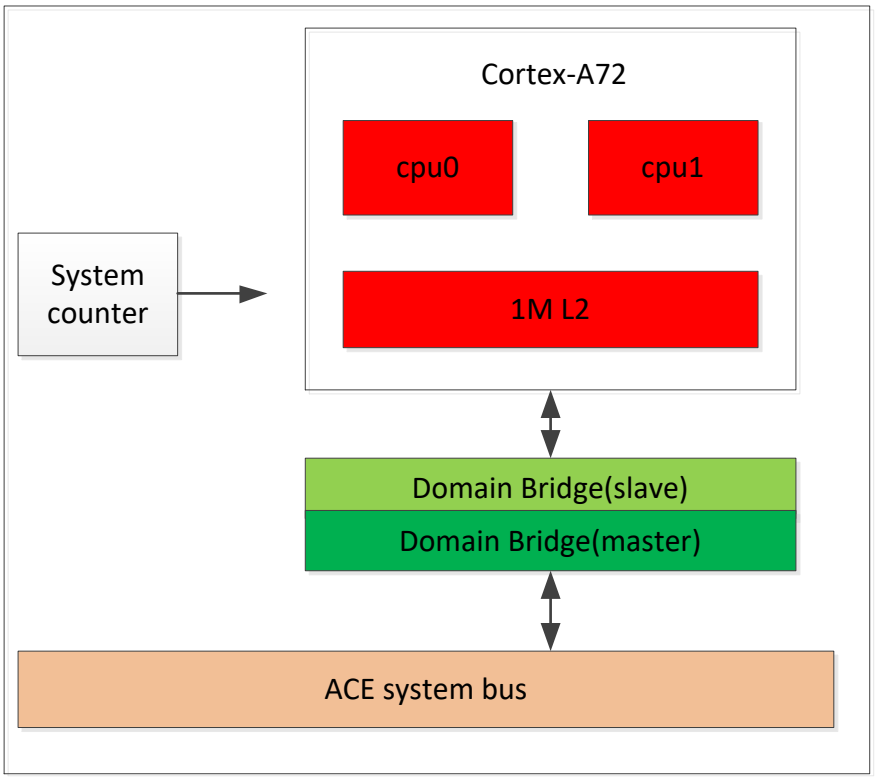


Fig. 5-1 Block Diagram

Chapter 6 Cortex-A53

6.1 Overview

The RK3399Pro has a quad-core Cortex-A53 cluster with 512K L2 memory. Cortex-A53 processor, which is a mid-range, low-power processor that implements the ARMv8-A architecture.

The Cortex-A53 processor includes following features:

- Full implementation of the ARMv8-A architecture instruction set
- Support for both AArch32 and AArch64 Execution status.
- Support for all exception levels, EL0, EL1, EL2, and EL3, in each execution states.
- Support A32 instruction set, previously called the ARM instruction set.
- Support T32 instruction set, previously called the Thumb instruction set.
- Support A64 instruction set.
- In-order pipeline with symmetric dual-issue of most instructions.
- Harvard Level 1(L1) memory system with a Memory Management Unit (MMU).
- Level 2 (L2) memory system providing cluster memory coherency, with L2 cache.
- Support advanced SIMD and Floating-point Extension for integer and floating-point vector operations.
- Support ARMv8 Cryptography Extensions.
- Support AMBA 4 ACE bus architecture.

The configuration details of little cluster and big cluster are shown in following tables

Table 6-1 CPU Configuration

Number of CPU	4
L1 I cache size	32K
L1 D cache size	32K
L2 cache size	512K
L2 data RAM output latency	3 cycles
L2 data RAM input latency	2 cycles
CPU cache protection	No
SCU L2 cache protection	No
BUS master interface	ACE
NEON and floating point support	Yes
Cryptography extension	Yes

6.2 Block Diagram

The Cortex-A53 sub system is shown in Figure 9-1. As illustrated, dual-core Cortex-A53 connects to system bus through asynchronous bridges which can handle with CDC (clock domain crossing) issue.

The Cortex-A53 is connected with system counter, which can run under a constant frequency clock, for PPI interrupt generation.

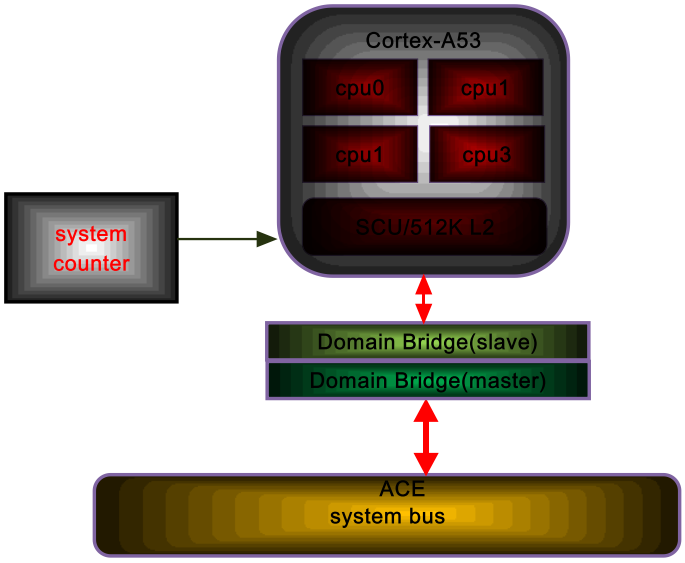


Fig. 6-1 Block Diagram

Chapter 7 Cortex-M0
7

7.1 Overview

The Cortex-M0 processor is a very low gate count, highly energy efficient processor that is intended for microcontroller and deeply embedded applications that require an area optimized processor.
The processor features and benefits are:

- A low gate count processor that features
  - The ARMv6-M Thumb instruction set
  - Thumb-2 technology
  - Compliant 24-bit SysTick timer
  - Load/store-multiples and multicycle-multiplies that can be abandoned and restarted to facilitate rapid interrupt handling
  - Low power sleep-mode entry using Wait For Interrupt (WFI), Wait For Event (WFE) instructions, or the return from interrupt sleep-on-exit feature
- NVIC that features
  - 32 external interrupt inputs, each with four levels of priority
  - Dedicated Non-Maskable Interrupt (NMI) input
  - Optional Wake-up Interrupt Controller (WIC), providing ultra-low power sleep mode support
- Optional debug support
  - Four hardware breakpoints
  - Two watch points
  - Support Serial Wire debug connection
  - single 32-bit AMBA-3 AHB-Lite system interface

7.2 Block Diagram

Cortex-M0 Integration architecture is shown below.

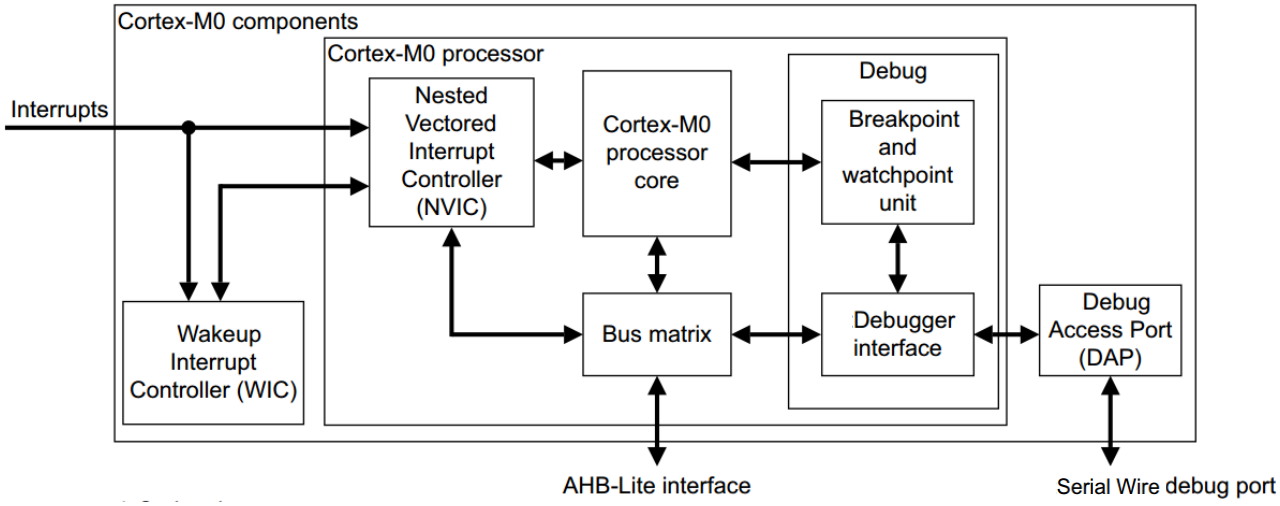


Fig. 7-1 Cortex-M0 Integration Architecture

There are two Cortex-M0 Integration instances in the SOC system, one in PERILP power domain, mainly for normal access, named “PERILPM0”; another is in PMU power domain, mainly for power management, named “PMUM0”.

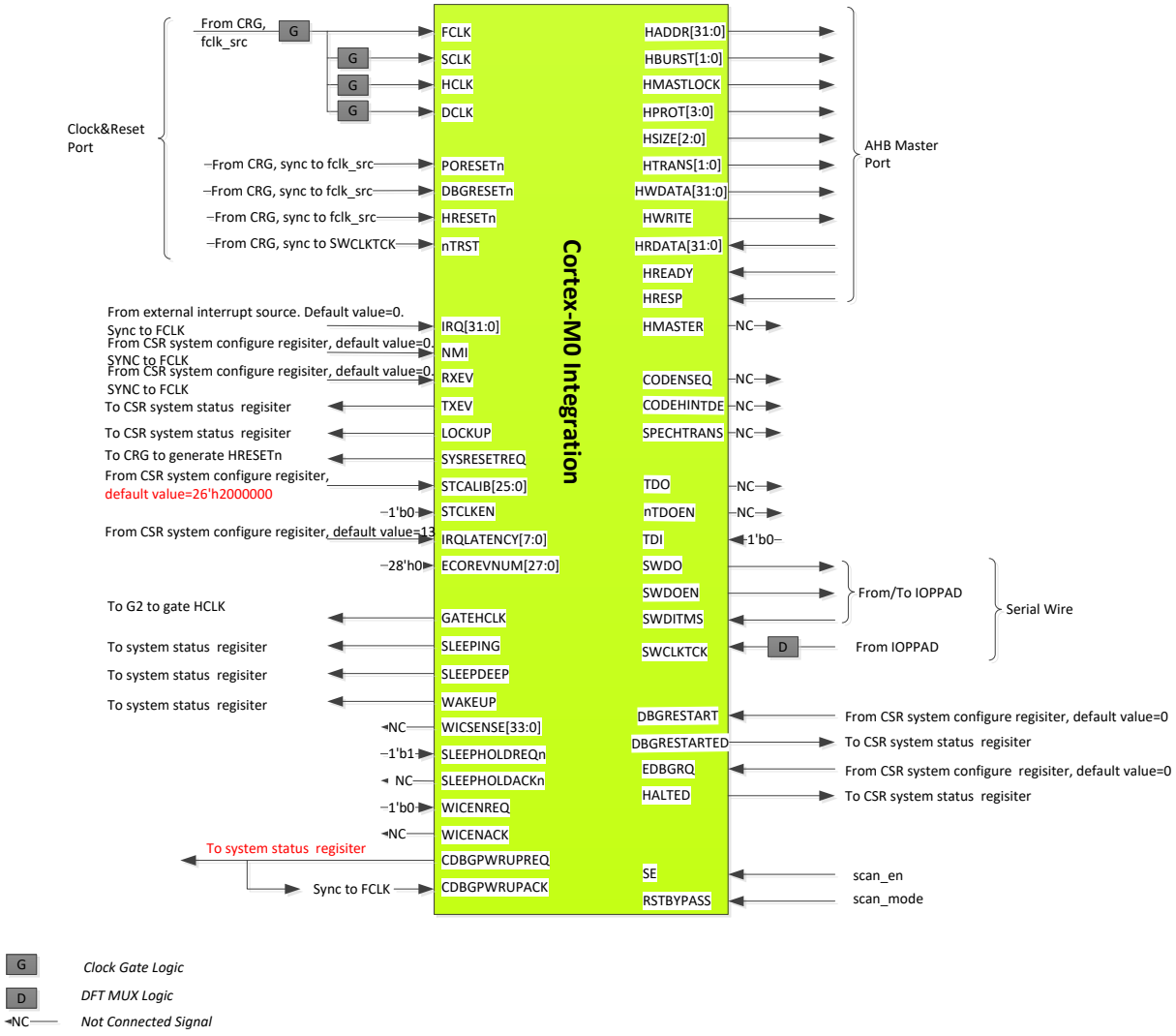


Fig. 7-2 PERILPM0 Architecture

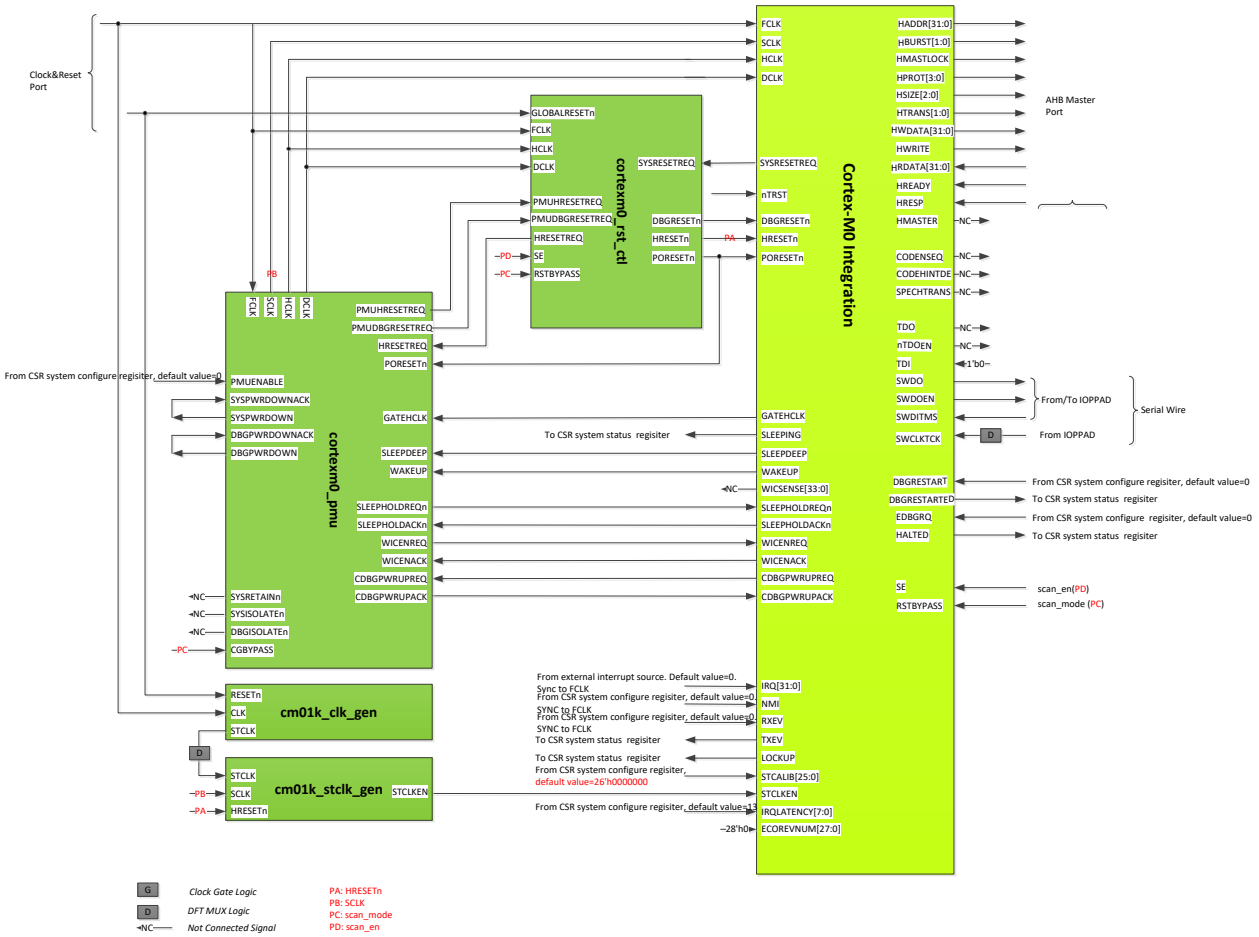


Fig. 7-3 PMUM0 Architecture

7.3 Interface Description

Table 7-1 Cortex-M0 Interface Description

Module Pin	Direction	Pad Name	IOMUX Setting
perilpm0_jtag_tck	I	IO_SDMMCclkout_MCUJTAGtck_HDCPJTAGtck_SDMMCgpio4b4	GRF_GPIO4B_IOMUX[9:8]=2'b10
perilpm0_jtag_tms	I/O	IO_SDMMCcmd_MCUJTAGtms_HDCPJTAGtms_SDMMCgpio4b5	GRF_GPIO4B_IOMUX[11:10]=2'b10
pmum0_jtag_tck	I	IO_PMUM0JTAGtck_SPI1ECclk_PMU1830gpio1b1	PMUGRF_GPIO1B_IOMUX[3:2]=2'b01
pmum0_jtag_tms	I/O	IO_PMUM0JTAGtms_SPI1ECcsn0_PMU1830gpio1b2	PMUGRF_GPIO1B_IOMUX[5:4]=2'b01
perilpm0_sleeping	O	IO_I2Sclk_TRACEctl_LPM0wfi_AUDIOgpio4a0	GRF_GPIO4A_IOMUX[1:0]=2'b11
pmum0_sleeping	O	IO_SDMMCwrprt_PMUM0wfi_TESTclkout2_PMU18gpio0b0	GRF_GPIO0B_IOMUX[1:0]=2'b10

Notes: I=input, O=output, I/O=input/output, bidirectional

7.4 Application Notes

7.4.1 Clock and Reset Generation

The clock and reset generation for PERILPM0 and PMUM0 are different, please refer to "Chapter CRU" for more detailed information.

7.4.2 Memory Remap for PERILPM0

The memory map is divided into different memory types for different usage.
To facilitate the memory remap between different processors, the source space can be remap to another. In addition, these remap operation could not take effect until the CortexM0 is soft reset.

Table 7-2 PERILPM0 Address Remap

Before Remap	Memory Type	Usage	After Remap
0xA0000000 - 0xDFFFFFFF	Device XN	Peripherals	[31:28]: sgrf_perilp_m0_con7[15:12] [27:12]: sgrf_perilp_m0_con6[15:0] [11:00]: 0
0x80000000 - 0x9FFFFFFF	Normal WT	Off chip RAM	[31:28]: sgrf_perilp_m0_con7[11:8] [27:12]: sgrf_perilp_m0_con5[15:0] [11:00]: 0
0x60000000 - 0x7FFFFFFF	Normal WBWA	Off chip RAM	[31:28]: sgrf_perilp_m0_con7[11:8] [27:12]: sgrf_perilp_m0_con5[15:0] [11:00]: 0
0x40000000 - 0x5FFFFFFF	Device XN	Peripherals	Before Remap + 0xB8000000
0x20000000 - 0x3FFFFFFF	Normal WBWA	On chip RAM	[31:28]: sgrf_perilp_m0_con7[7:4] [27:12]: sgrf_perilp_m0_con4[15:0] [11:00]: 0
0x00000000 - 0x1FFFFFFF	Normal WT	ROM or flash	[31:28]: sgrf_perilp_m0_con7[3:0] [27:12]: sgrf_perilp_m0_con3[15:0] [11:00]: 0

Notes:

- *XN means execute-never.*
- *WT means write-through.*
- *WBWA means write-back-write-allocate.*
- *WT means write-through.*

7.4.3 Memory Remap for PMUM0

The memory map is divided into different memory types for different usage.
To facilitate the memory remap between different processors, the source space can be remap to another. In addition, these remap operation could not take effect until the CortexM0 is soft reset.

Table 7-3 PMUM0 Address Remap

Before Remap	Memory Type	Usage	After Remap
0xA0000000 - 0xDFFFFFFF	Device XN	Peripherals	[31:28]: sgrf_pmu_con7[15:12] [27:12]: sgrf_pmu_con6[15:0] [11:00]: 0
0x80000000 - 0x9FFFFFFF	Normal WT	Off chip RAM	[31:28]: sgrf_pmu_con7[11:8] [27:12]: sgrf_pmu_con5[15:0] [11:00]: 0
0x60000000 - 0x7FFFFFFF	Normal WBWA	Off chip RAM	[31:28]: sgrf_pmu_con7[11:8] [27:12]: sgrf_pmu_con5[15:0] [11:00]: 0
0x40000000 - 0x5FFFFFFF	Device XN	Peripherals	Before Remap + 0xB8000000
0x20000000 - 0x3FFFFFFF	Normal WBWA	On chip RAM	[31:28]: sgrf_pmu_con7[7:4] [27:12]: sgrf_pmu_con4[15:0] [11:00]: 0
0x00000000 - 0x1FFFFFFF	Normal WT	ROM or flash	[31:28]: sgrf_pmu_con7[3:0] [27:12]: sgrf_pmu_con3[15:0] [11:00]: 0

Notes:

- *XN means execute-never.*
- *WT means write-through.*
- *WBWA means write-back-write-allocate.*
- *WT means write-through.*

7.4.4 Miscellaneous Signals for PERILPM0

System Configure Signals

Table 7-4 PERILPM0 System Configure Signals

Signal Name	Source	Def	Description
sgrf_perilp_cm0s_rsthold	sgrf_perilp_m0_con0[5]	0	Reset hold control. 0: reset can be asserted 1: reset cannot be asserted, so M0 RESETn will be high
sgrf_con_dbgen_m0_perilp	sgrf_perilp_m0_con0[1]	0	SerialWire Debug enable. 0: disable 1: enable

Signal Name	Source	Def	Description
sgrf_con_perilp_m0_jtag_rstreqn	sgrf_soc_con0[13]	0	Always be 0
sgrf_con_m0_perilp_sysrstreq_en	sgrf_soc_con0[14]	0	Enable for SYSRESETREQ
sgrf_con_perim0_secure_ctrl	sgrf_soc_con6[13]	1	Master security attribute: 0: secure 1: no-secure
grf_con_m0_perilp_stcalib[25:0]	{sgrf_perilp_m0_con2[9:0], sgrf_perilp_m0_con1[15:0]}	0	[25]: NOREF. Indicates that no alternative reference clock source has been integrated. Tie HIGH if STCLKEN has been tied off. [24]: SKEW. Tie this LOW if the system timer clock, the external reference clock, or SCLK as indicated by STCALIB[25], can guarantee an exact multiple of 10ms. Otherwise, tie this signal HIGH. [23:0]: TENMS. Provides an integer value to compute a 10ms (100Hz) delay from either the reference clock, or SCLK if the reference clock is not implemented.
grf_con_m0_perilp_irqlatency[7:0]	sgrf_perilp_m0_con8[7:0]	0	Minimum number of cycles between an interrupt that becomes pended in the NVIC, and the vector fetch for that interrupt being issued
sgrf_con_nmi_m0_perilp	sgrf_perilp_m0_con0[0]	0	Non-maskable interrupt
sgrf_con_dbgrestart_m0_perilp	sgrf_perilp_m0_con0[2]	0	External restart request
sgrf_con_edbgrq_m0_perilp	sgrf_perilp_m0_con0[3]	0	External debug request
sgrf_con_rxev_m0_perilp	sgrf_perilp_m0_con0[4]	0	A HIGH level on this input causes the architecture defined Event Register to be set in the Cortex-M0 processor. This causes a WFE instruction to complete. It also awakens the processor if it is sleeping as the result of encountering a WFE instruction when the Event Register is clear.

System Status Signals

Table 7-5 PERILPM0 System Status Signals

Signal Name	Destination	Def	Description
m0_perilp_sysresetreq	grf_soc_status2[9]	0	System reset request 0: no effect 1: requests a system level reset.
grf_stat_txev_m0_perilp	grf_soc_status2[6]	0	A single SCLK cycle HIGH level is generated on this output every time an SEV instruction is executed on the Cortex-M0 processor.
grf_stat_m0_perilp_dbgrestarted	grf_soc_status2[5]	1	Handshake for DBGRESTART
grf_stat_m0_perilp_core_halted	grf_soc_status2[4]	0	Indicates that the processor is in debug state. HALTED remains asserted for as long as the processor remains in debug state.
grf_stat_m0_perilp_core_lockup	grf_soc_status2[3]	0	Indicates that the processor is in the architected lock-up state, as the result of an unrecoverable exception.
grf_stat_m0_perilp_sleepdeep	grf_soc_status2[2]	0	Active only when SLEEPING is HIGH. Indicates that the SLEEPDEEP bit in the NVIC is set to 1.
grf_stat_m0_perilp_sleeping	grf_soc_status2[1]	0	Indicates the processor is idle, waiting for an interrupt on either the IRQ, NMI, or internal SysTick, or HIGH level on RXEV.
grf_stat_m0_perilp_wakeup	grf_soc_status2[0]	0	Active HIGH signal to the PMU that indicates a wake-up event has occurred and the processor system domain requires its clocks and power restored.

7.4.5 Miscellaneous Signals for PMUM0

System Configure Signals

Table 7-6 PMuM0 System Configure Signals

Signal Name	Source	Def	Description
sgrf_con_m0_stcalib[25:0]	{sgrf_pmu_con2[9:0],sgrf_pmu_con1[15:0]}	0	[25]: NOREF.Indicates that no alternative reference clock source has been integrated. Tie HIGH if STCLKEN has been tied off. [24]: SKEW. Tie this LOW if the system timer clock, the external reference clock, or SCLK as indicated by STCALIB[25], can guarantee an exact multiple of 10ms. Otherwise, tie this signal HIGH. [23:0]: TENMS. Provides an integer value to compute a 10ms (100Hz) delay from either the reference clock, or SCLK if the reference clock is not implemented.
sgrf_con_irqlatency_m0[7:0]	sgrf_pmu_con8[7:0]	0	Minimum number of cycles between an interrupt that becomes pended in the NVIC, and the vector fetch for that interrupt being issued
sgrf_pmu_cm0_nmi	sgrf_pmu_con0[0]	0	Non-maskable interrupt
sgrf_pmu_enable	sgrf_pmu_con0[1]	0	Internal PMU enable 0: disable 1: enable
sgrf_mcu_dbgrestart	sgrf_pmu_con0[2]	0	External restart request.
sgrf_mcu_edbgrq	sgrf_pmu_con0[3]	0	External debug request.
sgrf_mcu_rxev	sgrf_pmu_con0[4]	0	A HIGH level on this input causes the architecture defined Event Register to be set in the Cortex-M0 processor. This causes a WFE instruction to complete. It also awakens the processor if it is sleeping as the result of encountering a WFE instruction when the Event Register is clear.
sgrf_mcu_dbgen	sgrf_pmu_con0[5]	0	SerialWire Debug enable. 0: disable 1: enable
sgrf_pmu_cm0s_rsthold	sgrf_pmu_con0[6]	0	Reset hold control. 0: reset can be asserted 1: reset cannot be asserted, so M0 RESETn will be high
sgrf_pmu_cm0_mst_ctrl	sgrf_pmu_con0[7]	1	Master security attribute: 0: secure 1: no-secure

System Status Signals

Table 7-7 PMUM0 System Status Signals

Signal Name	Destination	Def	Description
grf_stat_mcu_txev	pmugrf_soc_status0[5]	0	A single SCLK cycle HIGH level is generated on this output every time an SEV instruction is executed on the Cortex-M0 processor.
grf_stat_mcu_dbgrestarted	pmugrf_soc_status0[4]	1	Handshake for DBGRESTART
grf_stat_mcu_core_halted	pmugrf_soc_status0[3]	0	Indicates that the processor is in debug state. HALTED remains asserted for as long as the processor remains in debug state.
grf_stat_mcu_core_lockup	pmugrf_soc_status0[2]	0	Indicates that the processor is in the architected lock-up state, as the result of an unrecoverable exception.
grf_stat_mcu_sleeping	pmugrf_soc_status0[1]	0	Active only when SLEEPING is HIGH. Indicates that the SLEEPDEEP bit in the NVIC is set to 1.
mcu_sleepdeep	pmugrf_soc_status0[0]	0	Indicates the processor is idle, waiting for an interrupt on either the IRQ, NMI, or internal SysTick, or HIGH level on RXEV.

7.4.6 Interrupt Source Arbiter for PERILPM0

The processor supports 32 external interrupt inputs, IRQ[31:0].
Every interrupt input has 8 interrupt sources, except IRQ[18]~IRQ[31] which has no interrupt source.
For IRQ[0]~IRQ[17]: Every interrupt source will be active and output to IRQ[i](i=0~17) by asserting the corresponding bit in INTR_ARB_MASKi(i=0~17). For example, if INTR_ARB_MASK1[1] is asserted to 1, then IRQ[1] is determined by dp_irq. The software can find out the acting interrupt source by reading INTR_ARB_FLAGi. For example, if INTR_ARB_FLAG1[1] is equal to 1, then it is considered that the dp_irq is active now.
INTR_ARB_MASKi(i=0~17) is write/read available, and the base address is (0xff798000 + 0x00+4*i).
INTR_ARB_FLAGi (i=0~17) is read only, and the base address is (0xff798000 + 0x80+4*i).
The relationship between IRQ[31:0], interrupt source, INTR_ARB_MASKi and INTR_ARB_FLAGi is shown below.

Table 7-8 Interrupt Source for PERILPM0

IRQ [31:0]	Int Source ID	Int Source	Int Arbiter Mask	Int Arbiter Flag
IRQ[0]	0	crypto0_int	INTR_ARB_MASK0[0]	INTR_ARB_FLAG0[0]
	1	dcf_done_int	INTR_ARB_MASK0[1]	INTR_ARB_FLAG0[1]
	2	dcf_error_int	INTR_ARB_MASK0[2]	INTR_ARB_FLAG0[2]
	3	ddrc0_int	INTR_ARB_MASK0[3]	INTR_ARB_FLAG0[3]
	4	ddrc1_int	INTR_ARB_MASK0[4]	INTR_ARB_FLAG0[4]
	5	dmac0_perilp_irq_abort	INTR_ARB_MASK0[5]	INTR_ARB_FLAG0[5]
	6	dmac0_perilp_irq	INTR_ARB_MASK0[6]	INTR_ARB_FLAG0[6]
	7	dmac1_perilp_irq_abort	INTR_ARB_MASK0[7]	INTR_ARB_FLAG0[7]
IRQ[1]	8	dmac1_perilp_irq	INTR_ARB_MASK1[0]	INTR_ARB_FLAG1[0]
	9	dp_irq	INTR_ARB_MASK1[1]	INTR_ARB_FLAG1[1]
	10	edp_irq	INTR_ARB_MASK1[2]	INTR_ARB_FLAG1[2]
	11	emmccore_int	INTR_ARB_MASK1[3]	INTR_ARB_FLAG1[3]
	12	gmac_int	INTR_ARB_MASK1[4]	INTR_ARB_FLAG1[4]
	13	gmac_pmt_int	INTR_ARB_MASK1[5]	INTR_ARB_FLAG1[5]
	14	gpio0_int	INTR_ARB_MASK1[6]	INTR_ARB_FLAG1[6]
	15	gpio1_int	INTR_ARB_MASK1[7]	INTR_ARB_FLAG1[7]
IRQ[2]	16	gpio2_intr	INTR_ARB_MASK2[0]	INTR_ARB_FLAG2[0]
	17	gpio3_intr	INTR_ARB_MASK2[1]	INTR_ARB_FLAG2[1]
	18	gpio4_intr	INTR_ARB_MASK2[2]	INTR_ARB_FLAG2[2]
	19	gpu_irqgpu	INTR_ARB_MASK2[3]	INTR_ARB_FLAG2[3]
	20	gpu_irqjob	INTR_ARB_MASK2[4]	INTR_ARB_FLAG2[4]
	21	gpu_irqmmu	INTR_ARB_MASK2[5]	INTR_ARB_FLAG2[5]
	22	hdcp22_irq	INTR_ARB_MASK2[6]	INTR_ARB_FLAG2[6]
	23	hdmi_irq	INTR_ARB_MASK2[7]	INTR_ARB_FLAG2[7]
IRQ[3]	24	hdmi_wakeup_irq	INTR_ARB_MASK3[0]	INTR_ARB_FLAG3[0]
	25	host0_arb_int	INTR_ARB_MASK3[1]	INTR_ARB_FLAG3[1]
	26	host0_ehci_int	INTR_ARB_MASK3[2]	INTR_ARB_FLAG3[2]
	27	host0_linestate_irq	INTR_ARB_MASK3[3]	INTR_ARB_FLAG3[3]
	28	host0_ohci_int	INTR_ARB_MASK3[4]	INTR_ARB_FLAG3[4]
	29	host1_arb_int	INTR_ARB_MASK3[5]	INTR_ARB_FLAG3[5]
	30	host1_ehci_int	INTR_ARB_MASK3[6]	INTR_ARB_FLAG3[6]
	31	host1_linestate_irq	INTR_ARB_MASK3[7]	INTR_ARB_FLAG3[7]
IRQ[4]	32	host1_ohci_int	INTR_ARB_MASK4[0]	INTR_ARB_FLAG4[0]
	33	hsic_int	INTR_ARB_MASK4[1]	INTR_ARB_FLAG4[1]
	34	i2c3_int	INTR_ARB_MASK4[2]	INTR_ARB_FLAG4[2]
	35	i2c2_int	INTR_ARB_MASK4[3]	INTR_ARB_FLAG4[3]
	36	i2c7_int	INTR_ARB_MASK4[4]	INTR_ARB_FLAG4[4]
	37	i2c6_int	INTR_ARB_MASK4[5]	INTR_ARB_FLAG4[5]
	38	i2c5_int	INTR_ARB_MASK4[6]	INTR_ARB_FLAG4[6]
	39	i2s0_int	INTR_ARB_MASK4[7]	INTR_ARB_FLAG4[7]
IRQ[5]	40	i2s1_int	INTR_ARB_MASK5[0]	INTR_ARB_FLAG5[0]
	41	i2s2_int	INTR_ARB_MASK5[1]	INTR_ARB_FLAG5[1]
	42	iep_intr	INTR_ARB_MASK5[2]	INTR_ARB_FLAG5[2]
	43	isp0_irq	INTR_ARB_MASK5[3]	INTR_ARB_FLAG5[3]

IRQ [31:0]	Int Source ID	Int Source	Int Arbiter Mask	Int Arbiter Flag
	44	isp1_irq	INTR_ARB_MASK5[4]	INTR_ARB_FLAG5[4]
	45	mipi_dsi_host0_irq	INTR_ARB_MASK5[5]	INTR_ARB_FLAG5[5]
	46	mipi_dsi_host1_irq	INTR_ARB_MASK5[6]	INTR_ARB_FLAG5[6]
	47	errirq_cci	INTR_ARB_MASK5[7]	INTR_ARB_FLAG5[7]
IRQ[6]	48	noc_intr	INTR_ARB_MASK6[0]	INTR_ARB_FLAG6[0]
	49	pcie_sys_int	INTR_ARB_MASK6[1]	INTR_ARB_FLAG6[1]
	50	pcie_legacy_int	INTR_ARB_MASK6[2]	INTR_ARB_FLAG6[2]
	51	pcie_client_int	INTR_ARB_MASK6[3]	INTR_ARB_FLAG6[3]
	52	spi2_int	INTR_ARB_MASK6[4]	INTR_ARB_FLAG6[4]
	53	spi1_int	INTR_ARB_MASK6[5]	INTR_ARB_FLAG6[5]
	54	pmu_int	INTR_ARB_MASK6[6]	INTR_ARB_FLAG6[6]
	55	rga_intr	INTR_ARB_MASK6[7]	INTR_ARB_FLAG6[7]
IRQ[7]	56	i2c4_int	INTR_ARB_MASK7[0]	INTR_ARB_FLAG7[0]
	57	i2c0_int	INTR_ARB_MASK7[1]	INTR_ARB_FLAG7[1]
	58	i2c8_int	INTR_ARB_MASK7[2]	INTR_ARB_FLAG7[2]
	59	i2c1_int	INTR_ARB_MASK7[3]	INTR_ARB_FLAG7[3]
	60	spi3_int	INTR_ARB_MASK7[4]	INTR_ARB_FLAG7[4]
	61	pwm_int	INTR_ARB_MASK7[5]	INTR_ARB_FLAG7[5]
	62	saradc_int	INTR_ARB_MASK7[6]	INTR_ARB_FLAG7[6]
	63	sd_detectn_irq	INTR_ARB_MASK7[7]	INTR_ARB_FLAG7[7]
IRQ[8]	64	sdio_int	INTR_ARB_MASK8[0]	INTR_ARB_FLAG8[0]
	65	sdmmc_int	INTR_ARB_MASK8[1]	INTR_ARB_FLAG8[1]
	66	spdif_int	INTR_ARB_MASK8[2]	INTR_ARB_FLAG8[2]
	67	spi4_int	INTR_ARB_MASK8[3]	INTR_ARB_FLAG8[3]
	68	spi0_int	INTR_ARB_MASK8[4]	INTR_ARB_FLAG8[4]
	69	stimer_intr0	INTR_ARB_MASK8[5]	INTR_ARB_FLAG8[5]
	70	stimer_intr1	INTR_ARB_MASK8[6]	INTR_ARB_FLAG8[6]
	71	stimer_intr2	INTR_ARB_MASK8[7]	INTR_ARB_FLAG8[7]
IRQ[9]	72	stimer_intr3	INTR_ARB_MASK9[0]	INTR_ARB_FLAG9[0]
	73	stimer_intr4	INTR_ARB_MASK9[1]	INTR_ARB_FLAG9[1]
	74	stimer_intr5	INTR_ARB_MASK9[2]	INTR_ARB_FLAG9[2]
	75	stimer_intr6	INTR_ARB_MASK9[3]	INTR_ARB_FLAG9[3]
	76	stimer_intr7	INTR_ARB_MASK9[4]	INTR_ARB_FLAG9[4]
	77	stimer_intr8	INTR_ARB_MASK9[5]	INTR_ARB_FLAG9[5]
	78	stimer_intr9	INTR_ARB_MASK9[6]	INTR_ARB_FLAG9[6]
	79	stimer_intr10	INTR_ARB_MASK9[7]	INTR_ARB_FLAG9[7]
IRQ[10]	80	stimer_intr11	INTR_ARB_MASK10[0]	INTR_ARB_FLAG10[0]
	81	timer_intr0	INTR_ARB_MASK10[1]	INTR_ARB_FLAG10[1]
	82	timer_intr1	INTR_ARB_MASK10[2]	INTR_ARB_FLAG10[2]
	83	timer_intr2	INTR_ARB_MASK10[3]	INTR_ARB_FLAG10[3]
	84	timer_intr3	INTR_ARB_MASK10[4]	INTR_ARB_FLAG10[4]
	85	timer_intr4	INTR_ARB_MASK10[5]	INTR_ARB_FLAG10[5]
	86	timer_intr5	INTR_ARB_MASK10[6]	INTR_ARB_FLAG10[6]
	87	timer_intr6	INTR_ARB_MASK10[7]	INTR_ARB_FLAG10[7]
IRQ[11]	88	timer_intr7	INTR_ARB_MASK11[0]	INTR_ARB_FLAG11[0]
	89	timer_intr8	INTR_ARB_MASK11[1]	INTR_ARB_FLAG11[1]
	90	timer_intr9	INTR_ARB_MASK11[2]	INTR_ARB_FLAG11[2]
	91	timer_intr10	INTR_ARB_MASK11[3]	INTR_ARB_FLAG11[3]
	92	timer_intr11	INTR_ARB_MASK11[4]	INTR_ARB_FLAG11[4]
	93	perf_int_a53	INTR_ARB_MASK11[5]	INTR_ARB_FLAG11[5]
	94	perf_int_a72	INTR_ARB_MASK11[6]	INTR_ARB_FLAG11[6]
	95	pmutimer_int0	INTR_ARB_MASK11[7]	INTR_ARB_FLAG11[7]
IRQ[12]	96	pmutimer_int1	INTR_ARB_MASK12[0]	INTR_ARB_FLAG12[0]
	97	tsadc_int	INTR_ARB_MASK12[1]	INTR_ARB_FLAG12[1]
	98	uart1_int	INTR_ARB_MASK12[2]	INTR_ARB_FLAG12[2]
	99	uart0_int	INTR_ARB_MASK12[3]	INTR_ARB_FLAG12[3]
	100	uart2_int	INTR_ARB_MASK12[4]	INTR_ARB_FLAG12[4]
	101	uart3_int	INTR_ARB_MASK12[5]	INTR_ARB_FLAG12[5]
	102	uart4_int	INTR_ARB_MASK12[6]	INTR_ARB_FLAG12[6]
	103	usb3otg0_bvalid_irq	INTR_ARB_MASK12[7]	INTR_ARB_FLAG12[7]
IRQ[13]	104	usb3otg0_id_irq	INTR_ARB_MASK13[0]	INTR_ARB_FLAG13[0]

IRQ [31:0]	Int Source ID	Int Source	Int Arbiter Mask	Int Arbiter Flag
	105	usb3otg0_int	INTR_ARB_MASK13[1]	INTR_ARB_FLAG13[1]
	106	usb3otg0_linestate_irq	INTR_ARB_MASK13[2]	INTR_ARB_FLAG13[2]
	107	usb3otg0_rxdet_irq	INTR_ARB_MASK13[3]	INTR_ARB_FLAG13[3]
	108	usb3otg1_bvalid_irq	INTR_ARB_MASK13[4]	INTR_ARB_FLAG13[4]
	109	usb3otg1_id_irq	INTR_ARB_MASK13[5]	INTR_ARB_FLAG13[5]
	110	usb3otg1_int	INTR_ARB_MASK13[6]	INTR_ARB_FLAG13[6]
	111	usb3otg1_linestate_irq	INTR_ARB_MASK13[7]	INTR_ARB_FLAG13[7]
IRQ[14]	112	usb3otg1_rxdet_irq	INTR_ARB_MASK14[0]	INTR_ARB_FLAG14[0]
	113	vcodec_dec_int	INTR_ARB_MASK14[1]	INTR_ARB_FLAG14[1]
	114	vcodec_enc_int	INTR_ARB_MASK14[2]	INTR_ARB_FLAG14[2]
	115	vcodec_mmu_int	INTR_ARB_MASK14[3]	INTR_ARB_FLAG14[3]
	116	vdu_dec_irq	INTR_ARB_MASK14[4]	INTR_ARB_FLAG14[4]
	117	vdu_mmu_irq	INTR_ARB_MASK14[5]	INTR_ARB_FLAG14[5]
	118	vopbig_irq	INTR_ARB_MASK14[6]	INTR_ARB_FLAG14[6]
	119	voplit_irq	INTR_ARB_MASK14[7]	INTR_ARB_FLAG14[7]
IRQ[15]	120	wdt0_intr	INTR_ARB_MASK15[0]	INTR_ARB_FLAG15[0]
	121	wdt1_intr	INTR_ARB_MASK15[1]	INTR_ARB_FLAG15[1]
	122	wdt2_int	INTR_ARB_MASK15[2]	INTR_ARB_FLAG15[2]
	123	usb3otg0_pme_generation	INTR_ARB_MASK15[3]	INTR_ARB_FLAG15[3]
	124	usb3otg0_host_legacy_smi_interrupt	INTR_ARB_MASK15[4]	INTR_ARB_FLAG15[4]
	125	usb3otg0_host_sys_err	INTR_ARB_MASK15[5]	INTR_ARB_FLAG15[5]
	126	usb3otg1_pme_generation	INTR_ARB_MASK15[6]	INTR_ARB_FLAG15[6]
	127	usb3otg1_host_legacy_smi_interrupt	INTR_ARB_MASK15[7]	INTR_ARB_FLAG15[7]
IRQ[16]	128	usb3otg1_host_sys_err	INTR_ARB_MASK16[0]	INTR_ARB_FLAG16[0]
	129	vopbig_irq_ddr	INTR_ARB_MASK16[1]	INTR_ARB_FLAG16[1]
	130	voplit_irq_ddr	INTR_ARB_MASK16[2]	INTR_ARB_FLAG16[2]
	131	ddr_mon_intr	INTR_ARB_MASK16[3]	INTR_ARB_FLAG16[3]
	132	spi5_int	INTR_ARB_MASK16[4]	INTR_ARB_FLAG16[4]
	133	tcpd_int0	INTR_ARB_MASK16[5]	INTR_ARB_FLAG16[5]
	134	tcpd_int1	INTR_ARB_MASK16[6]	INTR_ARB_FLAG16[6]
	135	crypto1_int	INTR_ARB_MASK16[7]	INTR_ARB_FLAG16[7]
IRQ[17]	136	gasket_irq	INTR_ARB_MASK17[0]	INTR_ARB_FLAG17[0]
	137	pcie_rc_mode_elec_idle_irq	INTR_ARB_MASK17[1]	INTR_ARB_FLAG17[1]
	138	\		
	139	\		
	140	perilp_mailbox_int[0]	INTR_ARB_MASK17[4]	INTR_ARB_FLAG17[4]
	141	perilp_mailbox_int[1]	INTR_ARB_MASK17[5]	INTR_ARB_FLAG17[5]
	142	perilp_mailbox_int[2]	INTR_ARB_MASK17[6]	INTR_ARB_FLAG17[6]
	143	perilp_mailbox_int[3]	INTR_ARB_MASK17[7]	INTR_ARB_FLAG17[7]
IRQ[18]~ IRQ[31]	\	Connect to 0	\	\

7.4.7 Interrupt Source Arbiter for PMUM0

The processor supports 32 external interrupt inputs, IRQ[31:0].
Every interrupt input has 8 interrupt sources, except IRQ[18]~IRQ[30] which has only one interrupt source and IRQ[31] which has no interrupt source.
For IRQ[0]~IRQ[17]: Every interrupt source will be active and output to IRQ[i](i=0~17) by asserting the corresponding bit in INTR_ARB_MASKi(i=0~17). For example, if INTR_ARB_MASK1[1] is asserted to 1, then IRQ[1] is determined by dp_irq. The software can find out the acting interrupt source by reading INTR_ARB_FLAGi. For example, if INTR_ARB_FLAG1[1] is equal to 1, then it is considered that the dp_irq is active now.
INTR_ARB_MASKi(i=0~17) is write/read available, and the base address is (0xff79c000 + 0x00+4*i).
INTR_ARB_FLAGi (i=0~17) is read only, and the base address is (0xff79c000 + 0x80+4*i).
The relationship between IRQ[31:0], interrupt source, INTR_ARB_MASKi and

INTR_ARB_FLAGi is shown below.

Table 7-9 Interrupt Source for PERILPM0

IRQ [31:0]	Int Arbiter ID	Int Arbiter Source	Int Arbiter Mask	Int Arbiter Flag
IRQ[0]	0	crypto0_int	INTR_ARB_MASK0[0]	INTR_ARB_FLAG0[0]
	1	dcf_done_int	INTR_ARB_MASK0[1]	INTR_ARB_FLAG0[1]
	2	dcf_error_int	INTR_ARB_MASK0[2]	INTR_ARB_FLAG0[2]
	3	ddrc0_int	INTR_ARB_MASK0[3]	INTR_ARB_FLAG0[3]
	4	ddrc1_int	INTR_ARB_MASK0[4]	INTR_ARB_FLAG0[4]
	5	dmac0_perilp_irq_abort	INTR_ARB_MASK0[5]	INTR_ARB_FLAG0[5]
	6	dmac0_perilp_irq	INTR_ARB_MASK0[6]	INTR_ARB_FLAG0[6]
	7	dmac1_perilp_irq_abort	INTR_ARB_MASK0[7]	INTR_ARB_FLAG0[7]
IRQ[1]	8	dmac1_perilp_irq	INTR_ARB_MASK1[0]	INTR_ARB_FLAG1[0]
	9	dp_irq	INTR_ARB_MASK1[1]	INTR_ARB_FLAG1[1]
	10	edp_irq	INTR_ARB_MASK1[2]	INTR_ARB_FLAG1[2]
	11	emmccore_int	INTR_ARB_MASK1[3]	INTR_ARB_FLAG1[3]
	12	gmac_int	INTR_ARB_MASK1[4]	INTR_ARB_FLAG1[4]
	13	gmac_pmt_int	INTR_ARB_MASK1[5]	INTR_ARB_FLAG1[5]
	14	gpio0_int	INTR_ARB_MASK1[6]	INTR_ARB_FLAG1[6]
	15	gpio1_int	INTR_ARB_MASK1[7]	INTR_ARB_FLAG1[7]
IRQ[2]	16	gpio2_intr	INTR_ARB_MASK2[0]	INTR_ARB_FLAG2[0]
	17	gpio3_intr	INTR_ARB_MASK2[1]	INTR_ARB_FLAG2[1]
	18	gpio4_intr	INTR_ARB_MASK2[2]	INTR_ARB_FLAG2[2]
	19	gpu_irqgpu	INTR_ARB_MASK2[3]	INTR_ARB_FLAG2[3]
	20	gpu_irqjob	INTR_ARB_MASK2[4]	INTR_ARB_FLAG2[4]
	21	gpu_irqmmu	INTR_ARB_MASK2[5]	INTR_ARB_FLAG2[5]
	22	hdcp22_irq	INTR_ARB_MASK2[6]	INTR_ARB_FLAG2[6]
	23	hdmi_irq	INTR_ARB_MASK2[7]	INTR_ARB_FLAG2[7]
IRQ[3]	24	hdmi_wakeup_irq	INTR_ARB_MASK3[0]	INTR_ARB_FLAG3[0]
	25	host0_arb_int	INTR_ARB_MASK3[1]	INTR_ARB_FLAG3[1]
	26	host0_ehci_int	INTR_ARB_MASK3[2]	INTR_ARB_FLAG3[2]
	27	host0_linestate_irq	INTR_ARB_MASK3[3]	INTR_ARB_FLAG3[3]
	28	host0_ohci_int	INTR_ARB_MASK3[4]	INTR_ARB_FLAG3[4]
	29	host1_arb_int	INTR_ARB_MASK3[5]	INTR_ARB_FLAG3[5]
	30	host1_ehci_int	INTR_ARB_MASK3[6]	INTR_ARB_FLAG3[6]
	31	host1_linestate_irq	INTR_ARB_MASK3[7]	INTR_ARB_FLAG3[7]
IRQ[4]	32	host1_ohci_int	INTR_ARB_MASK4[0]	INTR_ARB_FLAG4[0]
	33	hsic_int	INTR_ARB_MASK4[1]	INTR_ARB_FLAG4[1]
	34	i2c3_int	INTR_ARB_MASK4[2]	INTR_ARB_FLAG4[2]
	35	i2c2_int	INTR_ARB_MASK4[3]	INTR_ARB_FLAG4[3]
	36	i2c7_int	INTR_ARB_MASK4[4]	INTR_ARB_FLAG4[4]
	37	i2c6_int	INTR_ARB_MASK4[5]	INTR_ARB_FLAG4[5]
	38	i2c5_int	INTR_ARB_MASK4[6]	INTR_ARB_FLAG4[6]
	39	i2s0_int	INTR_ARB_MASK4[7]	INTR_ARB_FLAG4[7]
IRQ[5]	40	i2s1_int	INTR_ARB_MASK5[0]	INTR_ARB_FLAG5[0]
	41	i2s2_int	INTR_ARB_MASK5[1]	INTR_ARB_FLAG5[1]
	42	iep_intr	INTR_ARB_MASK5[2]	INTR_ARB_FLAG5[2]
	43	isp0_irq	INTR_ARB_MASK5[3]	INTR_ARB_FLAG5[3]
	44	isp1_irq	INTR_ARB_MASK5[4]	INTR_ARB_FLAG5[4]
	45	mipi_dsi_host0_irq	INTR_ARB_MASK5[5]	INTR_ARB_FLAG5[5]
	46	mipi_dsi_host1_irq	INTR_ARB_MASK5[6]	INTR_ARB_FLAG5[6]
	47	errirq_cci	INTR_ARB_MASK5[7]	INTR_ARB_FLAG5[7]
IRQ[6]	48	noc_intr	INTR_ARB_MASK6[0]	INTR_ARB_FLAG6[0]
	49	pcie_sys_int	INTR_ARB_MASK6[1]	INTR_ARB_FLAG6[1]
	50	pcie_legacy_int	INTR_ARB_MASK6[2]	INTR_ARB_FLAG6[2]
	51	pcie_client_int	INTR_ARB_MASK6[3]	INTR_ARB_FLAG6[3]
	52	spi2_int	INTR_ARB_MASK6[4]	INTR_ARB_FLAG6[4]
	53	spi1_int	INTR_ARB_MASK6[5]	INTR_ARB_FLAG6[5]
	54	pmu_int	INTR_ARB_MASK6[6]	INTR_ARB_FLAG6[6]
	55	rga_intr	INTR_ARB_MASK6[7]	INTR_ARB_FLAG6[7]
IRQ[7]	56	i2c4_int	INTR_ARB_MASK7[0]	INTR_ARB_FLAG7[0]
	57	i2c0_int	INTR_ARB_MASK7[1]	INTR_ARB_FLAG7[1]
	58	i2c8_int	INTR_ARB_MASK7[2]	INTR_ARB_FLAG7[2]

IRQ [31:0]	Int Arbiter ID	Int Arbiter Source	Int Arbiter Mask	Int Arbiter Flag
	59	i2c1_int	INTR_ARB_MASK7[3]	INTR_ARB_FLAG7[3]
	60	spi3_int	INTR_ARB_MASK7[4]	INTR_ARB_FLAG7[4]
	61	pwm_int	INTR_ARB_MASK7[5]	INTR_ARB_FLAG7[5]
	62	saradc_int	INTR_ARB_MASK7[6]	INTR_ARB_FLAG7[6]
	63	sd_detectn_irq	INTR_ARB_MASK7[7]	INTR_ARB_FLAG7[7]
IRQ[8]	64	sdio_int	INTR_ARB_MASK8[0]	INTR_ARB_FLAG8[0]
	65	sdmmc_int	INTR_ARB_MASK8[1]	INTR_ARB_FLAG8[1]
	66	spdif_int	INTR_ARB_MASK8[2]	INTR_ARB_FLAG8[2]
	67	spi4_int	INTR_ARB_MASK8[3]	INTR_ARB_FLAG8[3]
	68	spi0_int	INTR_ARB_MASK8[4]	INTR_ARB_FLAG8[4]
	69	stimer_intr0	INTR_ARB_MASK8[5]	INTR_ARB_FLAG8[5]
	70	stimer_intr1	INTR_ARB_MASK8[6]	INTR_ARB_FLAG8[6]
	71	stimer_intr2	INTR_ARB_MASK8[7]	INTR_ARB_FLAG8[7]
IRQ[9]	72	stimer_intr3	INTR_ARB_MASK9[0]	INTR_ARB_FLAG9[0]
	73	stimer_intr4	INTR_ARB_MASK9[1]	INTR_ARB_FLAG9[1]
	74	stimer_intr5	INTR_ARB_MASK9[2]	INTR_ARB_FLAG9[2]
	75	stimer_intr6	INTR_ARB_MASK9[3]	INTR_ARB_FLAG9[3]
	76	stimer_intr7	INTR_ARB_MASK9[4]	INTR_ARB_FLAG9[4]
	77	stimer_intr8	INTR_ARB_MASK9[5]	INTR_ARB_FLAG9[5]
	78	stimer_intr9	INTR_ARB_MASK9[6]	INTR_ARB_FLAG9[6]
	79	stimer_intr10	INTR_ARB_MASK9[7]	INTR_ARB_FLAG9[7]
IRQ[10]	80	stimer_intr11	INTR_ARB_MASK10[0]	INTR_ARB_FLAG10[0]
	81	timer_intr0	INTR_ARB_MASK10[1]	INTR_ARB_FLAG10[1]
	82	timer_intr1	INTR_ARB_MASK10[2]	INTR_ARB_FLAG10[2]
	83	timer_intr2	INTR_ARB_MASK10[3]	INTR_ARB_FLAG10[3]
	84	timer_intr3	INTR_ARB_MASK10[4]	INTR_ARB_FLAG10[4]
	85	timer_intr4	INTR_ARB_MASK10[5]	INTR_ARB_FLAG10[5]
	86	timer_intr5	INTR_ARB_MASK10[6]	INTR_ARB_FLAG10[6]
	87	timer_intr6	INTR_ARB_MASK10[7]	INTR_ARB_FLAG10[7]
IRQ[11]	88	timer_intr7	INTR_ARB_MASK11[0]	INTR_ARB_FLAG11[0]
	89	timer_intr8	INTR_ARB_MASK11[1]	INTR_ARB_FLAG11[1]
	90	timer_intr9	INTR_ARB_MASK11[2]	INTR_ARB_FLAG11[2]
	91	timer_intr10	INTR_ARB_MASK11[3]	INTR_ARB_FLAG11[3]
	92	timer_intr11	INTR_ARB_MASK11[4]	INTR_ARB_FLAG11[4]
	93	perf_int_a53	INTR_ARB_MASK11[5]	INTR_ARB_FLAG11[5]
	94	perf_int_a72	INTR_ARB_MASK11[6]	INTR_ARB_FLAG11[6]
	95	pmutimer_int0	INTR_ARB_MASK11[7]	INTR_ARB_FLAG11[7]
IRQ[12]	96	pmutimer_int1	INTR_ARB_MASK12[0]	INTR_ARB_FLAG12[0]
	97	tsadc_int	INTR_ARB_MASK12[1]	INTR_ARB_FLAG12[1]
	98	uart1_int	INTR_ARB_MASK12[2]	INTR_ARB_FLAG12[2]
	99	uart0_int	INTR_ARB_MASK12[3]	INTR_ARB_FLAG12[3]
	100	uart2_int	INTR_ARB_MASK12[4]	INTR_ARB_FLAG12[4]
	101	uart3_int	INTR_ARB_MASK12[5]	INTR_ARB_FLAG12[5]
	102	uart4_int	INTR_ARB_MASK12[6]	INTR_ARB_FLAG12[6]
	103	usb3otg0_bvalid_irq	INTR_ARB_MASK12[7]	INTR_ARB_FLAG12[7]
IRQ[13]	104	usb3otg0_id_irq	INTR_ARB_MASK13[0]	INTR_ARB_FLAG13[0]
	105	usb3otg0_int	INTR_ARB_MASK13[1]	INTR_ARB_FLAG13[1]
	106	usb3otg0_linestate_irq	INTR_ARB_MASK13[2]	INTR_ARB_FLAG13[2]
	107	usb3otg0_rxdet_irq	INTR_ARB_MASK13[3]	INTR_ARB_FLAG13[3]
	108	usb3otg1_bvalid_irq	INTR_ARB_MASK13[4]	INTR_ARB_FLAG13[4]
	109	usb3otg1_id_irq	INTR_ARB_MASK13[5]	INTR_ARB_FLAG13[5]
	110	usb3otg1_int	INTR_ARB_MASK13[6]	INTR_ARB_FLAG13[6]
	111	usb3otg1_linestate_irq	INTR_ARB_MASK13[7]	INTR_ARB_FLAG13[7]
IRQ[14]	112	usb3otg1_rxdet_irq	INTR_ARB_MASK14[0]	INTR_ARB_FLAG14[0]
	113	vcodec_dec_int	INTR_ARB_MASK14[1]	INTR_ARB_FLAG14[1]
	114	vcodec_enc_int	INTR_ARB_MASK14[2]	INTR_ARB_FLAG14[2]
	115	vcodec_mmu_int	INTR_ARB_MASK14[3]	INTR_ARB_FLAG14[3]
	116	vdu_dec_irq	INTR_ARB_MASK14[4]	INTR_ARB_FLAG14[4]
	117	vdu_mmu_irq	INTR_ARB_MASK14[5]	INTR_ARB_FLAG14[5]
	118	vopbig_irq	INTR_ARB_MASK14[6]	INTR_ARB_FLAG14[6]
	119	voplit_irq	INTR_ARB_MASK14[7]	INTR_ARB_FLAG14[7]

IRQ [31:0]	Int Arbiter ID	Int Arbiter Source	Int Arbiter Mask	Int Arbiter Flag
IRQ[15]	120	wdt0_intr	INTR_ARB_MASK15[0]	INTR_ARB_FLAG15[0]
	121	wdt1_intr	INTR_ARB_MASK15[1]	INTR_ARB_FLAG15[1]
	122	wdt2_int	INTR_ARB_MASK15[2]	INTR_ARB_FLAG15[2]
	123	usb3otg0_pme_generation	INTR_ARB_MASK15[3]	INTR_ARB_FLAG15[3]
	124	usb3otg0_host_legacy_smi_interrupt	INTR_ARB_MASK15[4]	INTR_ARB_FLAG15[4]
	125	usb3otg0_host_sys_err	INTR_ARB_MASK15[5]	INTR_ARB_FLAG15[5]
	126	usb3otg1_pme_generation	INTR_ARB_MASK15[6]	INTR_ARB_FLAG15[6]
	127	usb3otg1_host_legacy_smi_interrupt	INTR_ARB_MASK15[7]	INTR_ARB_FLAG15[7]
IRQ[16]	128	usb3otg1_host_sys_err	INTR_ARB_MASK16[0]	INTR_ARB_FLAG16[0]
	129	vopbig_irq_ddr	INTR_ARB_MASK16[1]	INTR_ARB_FLAG16[1]
	130	voplit_irq_ddr	INTR_ARB_MASK16[2]	INTR_ARB_FLAG16[2]
	131	ddr_mon_intr	INTR_ARB_MASK16[3]	INTR_ARB_FLAG16[3]
	132	spi5_int	INTR_ARB_MASK16[4]	INTR_ARB_FLAG16[4]
	133	tcpd_int0	INTR_ARB_MASK16[5]	INTR_ARB_FLAG16[5]
	134	tcpd_int1	INTR_ARB_MASK16[6]	INTR_ARB_FLAG16[6]
	135	crypto1_int	INTR_ARB_MASK16[7]	INTR_ARB_FLAG16[7]
IRQ[17]	136	gasket_irq	INTR_ARB_MASK17[0]	INTR_ARB_FLAG17[0]
	137	pcie_rc_mode_elec_idle_irq	INTR_ARB_MASK17[1]	INTR_ARB_FLAG17[1]
	138	\		
	139	\		
	140	pmu_mailbox_int[0]	INTR_ARB_MASK17[4]	INTR_ARB_FLAG17[4]
	141	pmu_mailbox_int[1]	INTR_ARB_MASK17[5]	INTR_ARB_FLAG17[5]
	142	pmu_mailbox_int[2]	INTR_ARB_MASK17[6]	INTR_ARB_FLAG17[6]
	143	pmu_mailbox_int[3]	INTR_ARB_MASK17[7]	INTR_ARB_FLAG17[7]
IRQ[18]	\	gpio0_int	\	\
IRQ[19]	\	gpio1_int	\	\
IRQ[20]	\	pmu_int	\	\
IRQ[21]	\	i2c_sensor_int	\	\
IRQ[22]	\	i2c_pmu_int	\	\
IRQ[23]	\	i2c_dcdc_int	\	\
IRQ[24]	\	spi_pmu_int	\	\
IRQ[25]	\	rkpwm_pmu_int	\	\
IRQ[26]	\	timer_pmu_int0	\	\
IRQ[27]	\	timer_pmu_int1	\	\
IRQ[28]	\	uartm0_int	\	\
IRQ[29]	\	wdt_m0_pmu_int	\	\
IRQ[30]	\	|pmu_mailbox_int[3:0]	\	\
IRQ[31]	\	\	\	\

Chapter 8 Embedded SRAM

8.1 Overview

The Embedded SRAM supports read and write access to provide system fast access data storage

8.1.1 Features supported

- Provide 8KB access space in PMU
- Provide 192KB access space in perilp
- Support security and non-security access
- Security or non-security space is software programmable for 192KB SRAM
- Security space is nx4KB(up to whole memory space)
- Support 32bit AHB bus for 8KB SRAM
- Support 64bit AXI bus for 192KB SRAM

8.1.2 Features not supported

- Don't support AXI lock transaction
- Don't support AXI exclusive transaction
- Don't support AXI cache function
- Don't support AXI protection function

8.2 Block Diagram

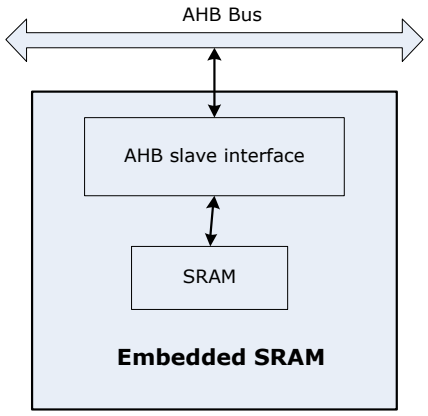


Fig. 8-1 8KB Embedded SRAM block diagram

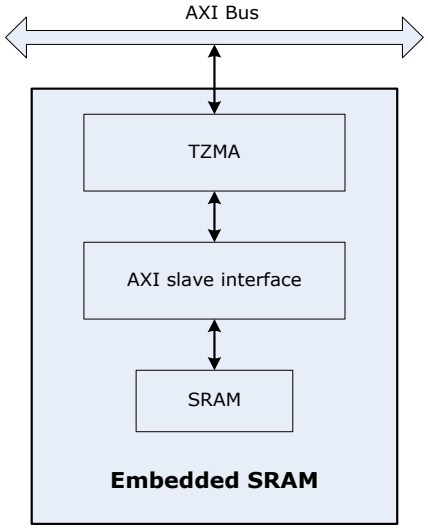


Fig. 8-2 192KB Embedded SRAM block diagram

8.3 Function Description

8.3.1 TZMA

Please refer to secure chapter for TZMA functional description

8.3.2 Embedded SRAM access path

The 8KB Embedded SRAM can only be accessed by Cortex-A72, Cortex-A53, Coresight, PMU Cortex-M0 and perilp Cortex-M0.

The 192KB Embedded SRAM can only be accessed by Cortex-A72, Cortex-A53,Coresight, PMU Cortex-M0,perilp Cortex-M0,DMAC0, DMAC1, CRYPTO0, CRYPTO1, DCF, GIC500, PCIE, USB-HOST and USB-OTG.

8.3.3 Remap

The 192KB Embedded SRAM support remap.

Before remap, the Embedded SRAM address range is 0xff8c_0000~0xff8e_ffff,

After set remap, (ref Security GRF register SGRF_SOC_CON3, bit[7]), the system can still access the Embedded SRAM by the old address. At same time, the system also can access the Embedded SRAM by the new address 0xffff_0000 ~ 0xffff_ffff (include the bootaddr).

Chapter 9 Power Management Unit (PMU)

9.1 Overview

In order to meet low power requirements, a power management unit (PMU) is designed for controlling power resources in RK3399Pro. The RK3399Pro PMU is dedicated for managing the power of the whole chip.

9.1.1 Features

- Support 6 voltage domains including VD_CORE_L, VD_CORE_B, VD_CENTER, VD_GPU, VD_LOGIC and VD_PMU
- Support 31 separate power domains in the whole chip, which can be power up/down by software based on different application scenes
- In low power mode, PMU could power up/down pd_a53_0/pd_a72_0, vd_core_l/vd_core_b, pd_cci, pd_perilp, vd_center by hardware
- Support Cortex-A53/A72 core, pd_center, pd_perilp source clock gating in low power mode
- Support Cortex-A53 L2 flush request by hardware in low power mode
- Support power down/up all power domains by software
- Support core wfi auto power down by hardware

9.2 Block Diagram

9.2.1 power domain partition

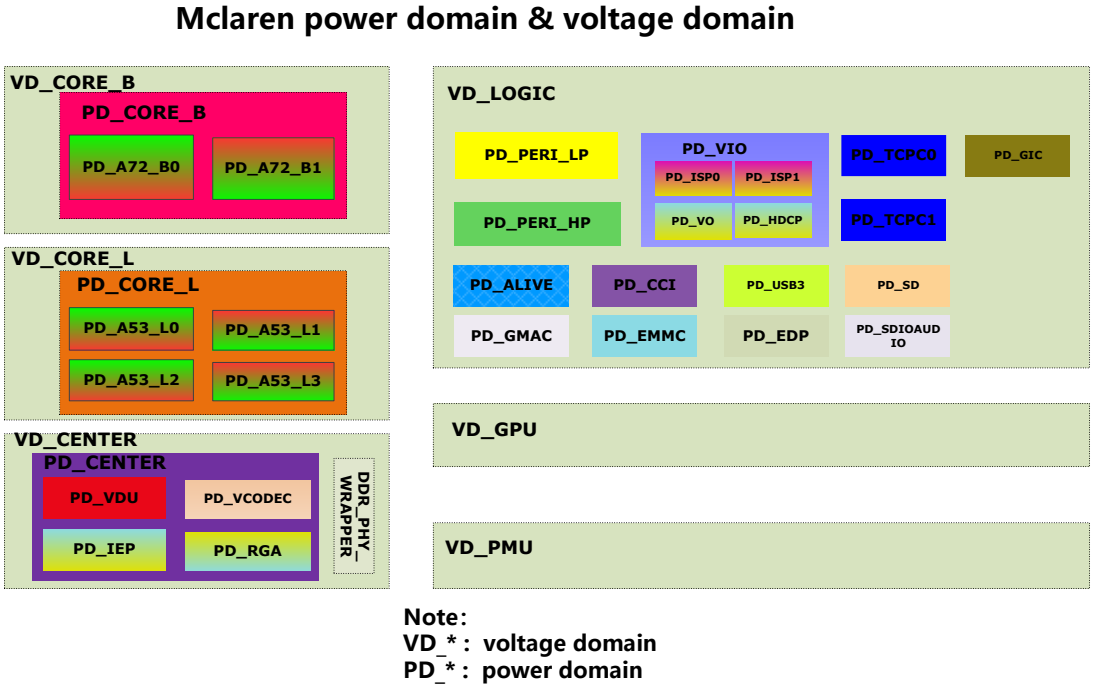


Fig. 9-1 RK3399Pro Power Domain Partition

The above diagram describes the power domain and voltage domain partition, and the following table lists all the power domains.

Table 4-1 RK3399Pro Power Domain and Voltage Domain Summary

Voltage Domain	Power Domain	Description
VD_CORE_L	pd_a53_l0	Cortex-A53 core 0, L1C and Neon
	pd_a53_l1	Cortex-A53 core 1, L1C and Neon
	pd_a53_l2	Cortex-A53 core 2, L1C and Neon
	pd_a53_l3	Cortex-A53 core 3, L1C and Neon

	pd_scu_l	SCU, L2
VD_CORE_B	pd_a72_b0	Cortex-A72 core 0, L1C and Neon
	pd_a72_b1	Cortex-A72 core 1, L1C and Neon
	pd_scu_b	SCU, L2
VD_LOGIC	pd_perilp	cm0, crypto, dcf, imem, dmac, bootrom, efuse_con, spi, i2c, uart, saradc, tsadc
	pd_perihp	pcie, usb2, hsic
	pd_vio	include pd_isp0, pd_isp1, pd_vo, pd_hdcp, mipi dsi
	pd_isp0	isp0
	pd_isp1	isp1
	pd_vo	vopb, vopl
	pd_hdcp	hdcp, hdmi, dptx
	pd_tcpc0	tcpc0
	pd_tcpc1	tcpc1
	pd_alive	cru, grf, timer, gpio, wdt
	pd_gmac	gmac
	pd_cci	cci
	pd_emmc	emmc
	pd_usb3	usb3
	pd_edp	edp
	pd_sd	sdmmc
	pd_sdioaudio	sdio, spi, i2s, spdif
VD_CENTER	pd_vcodec	VDPU,VEPU
	pd_vdu	rkvdec
	pd_rga	RGA
	pd_iep	IEP
	pd_center	DDR
VD_GPU	pd_gpu	GPU
VD_PMU	pd_pmu	cm0, PMU, SRAM(8K), Secure GRF, GPIO0, PVTM,i2c

9.2.2 PMU block diagram

The following figure is the PMU block diagram. The PMU includes the 3 following sections:

- APB interface and register, which can accept the system configuration
- Low Power State Control, which generate low power control signals.
- Power Switch Control, which control all power domain switch

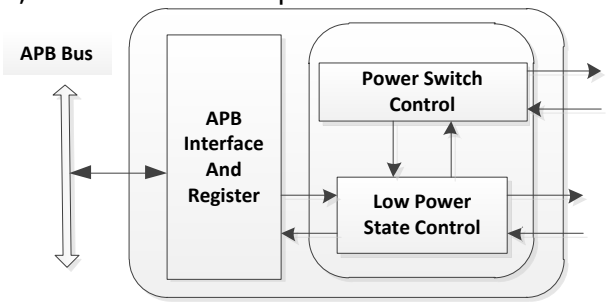


Fig. 9-2 PMU Bock Diagram

9.3 Function Description

First of all, we define two operation modes of PMU, normal mode and low power mode.
When operating at normal mode, that means software can manage power sources directly by accessing PMU register.
For example, Cortex-A53 CPU can write PMU_PWRDN_CON register to determine that power off/on which power domain independently.
When operating at low power mode, software manages power sources indirectly through FSM (Finite States Machine) in PMU and those settings always not take effect immediately.

That means software also can configure PMU registers to power down/up some power resources, but these setting will not be executed immediately after configuration. They will delay to execute after FSM running in particular phase. The low power mode can support some functions that cannot support in normal mode. For example, some components inside RK3399Pro (e.g. Cortex-A53 core 0) can shut down itself through low power mode.
To entering low power mode, after setting some power configurations, the PMU_POWER_MODE[0] bit must be set 1 to enable PMU FSM. Then Cortex-A53 CPU needs to execute a WFI command to perform ready signal. After PMU detects all Cortex-A53 CPUs in WFI status, then the FSM will be fetched. And the specific power sources will be controlled during specific status in FSM. So the low power mode is a "delay affect" way to handle power sources inside the RK3399Pro chip.

9.4 Register Description

9.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
PMU_WAKEUP_CFG0	0x0000	W	0x00000000	pmu wakeup configure register 0
PMU_WAKEUP_CFG1	0x0004	W	0x00000000	pmu wakeup configure register 1
PMU_WAKEUP_CFG2	0x0008	W	0x00000000	pmu wakeup configure register 2
PMU_WAKEUP_CFG3	0x000c	W	0x00000000	pmu wakeup configure register 3
PMU_WAKEUP_CFG4	0x0010	W	0x00000000	pmu wakeup configure register 4
PMU_PWRDN_CON	0x0014	W	0x00000000	pmu power down configure register
PMU_PWRDN_ST	0x0018	W	0x00000000	pmu power down status register
PMU_PLL_CON	0x001c	W	0x00000000	PLL low power control register
PMU_PWRMODE_CON	0x0020	W	0x00000000	pmu power mode configure register of common resource
PMU_SFT_CON	0x0024	W	0x00000000	pmu software configure register
PMU_INT_CON	0x0028	W	0x00000000	pmu interrupt configure register
PMU_INT_ST	0x002c	W	0x00000000	pmu interrupt status register
PMU_GPIO0_POS_INT_CON	0x0030	W	0x00000000	pmu gpio0 posedge interrupt configure register
PMU_GPIO0_NEG_INT_CON	0x0034	W	0x00000000	pmu gpio0 negedge interrupt configure register
PMU_GPIO1_POS_INT_CON	0x0038	W	0x00000000	pmu gpio1 posedge interrupt configure register
PMU_GPIO1_NEG_INT_CON	0x003c	W	0x00000000	pmu gpio1 negedge interrupt configure register
PMU_GPIO0_POS_INT_ST	0x0040	W	0x00000000	pmu gpio0 posedge interrupt status register
PMU_GPIO0_NEG_INT_ST	0x0044	W	0x00000000	pmu gpio0 negedge interrupt status register
PMU_GPIO1_POS_INT_ST	0x0048	W	0x00000000	pmu gpio1 posedge interrupt status register
PMU_GPIO1_NEG_INT_ST	0x004c	W	0x00000000	pmu gpio1 negedge interrupt status register
PMU_PWRDN_INTEN	0x0050	W	0x00000000	pmu power down interrupt enable register

Name	Offset	Size	Reset Value	Description
PMU_PWRDN_STATUS	0x0054	W	0x00000000	pmu power down interrupt status register
PMU_WAKEUP_STATUS	0x0058	W	0x00000000	pmu interrupt wakeup status register
PMU_BUS_CLR	0x005c	W	0x00000000	pmu bus clear register
PMU_BUS_IDLE_REQ	0x0060	W	0x00000000	pmu bus idle request register
PMU_BUS_IDLE_ST	0x0064	W	0x00000000	pmu bus idle status register
PMU_BUS_IDLE_ACK	0x0068	W	0x00000000	pmu bus idle ack status register
PMU_CCI500_CON	0x006c	W	0x00000000	CCI-500 low power control register
PMU_ADB400_CON	0x0070	W	0x00000000	adb-400 low power control register
PMU_ADB400_ST	0x0074	W	0x00000000	adb-400 low power status register
PMU_POWER_ST	0x0078	W	0x00000000	pmu power status register
PMU_CORE_PWR_ST	0x007c	W	0x00000000	pmu core power status register
PMU_OSC_CNT	0x0080	W	0x00000000	pmu osc count register
PMU_PLLLOCK_CNT	0x0084	W	0x00000000	pmu pll lock count register
PMU_PLLRST_CNT	0x0088	W	0x00000000	pmu pll reset count register
PMU_STABLE_CNT	0x008c	W	0x00000000	pmu power stable count register
PMU_DDRIO_PWRON_CNT	0x0090	W	0x00000000	pmu ddrio power on count register
PMU_WAKEUP_RST_CLR_CNT	0x0094	W	0x00000000	pmu wakeup reset clear count register
PMU_DDR_SREF_ST	0x0098	W	0x00000000	pmu ddr self refresh status register
PMU_SCU_L_PWRDN_CNT	0x009c	W	0x00005dc0	pmu scu_l power down count register
PMU_SCU_L_PWRUP_CNT	0x00a0	W	0x00005dc0	pmu scu_l power up count register
PMU_SCU_B_PWRDN_CNT	0x00a4	W	0x00005dc0	pmu scu_b power down count register
PMU_SCU_B_PWRUP_CNT	0x00a8	W	0x00005dc0	pmu scu_b power up count register
PMU_GPU_PWRDN_CNT	0x00ac	W	0x00005dc0	pmu gpu power down count register
PMU_GPU_PWRUP_CNT	0x00b0	W	0x00005dc0	pmu gpu power up count register
PMU_CENTER_PWRDN_CNT	0x00b4	W	0x00005dc0	pmu center power down count register
PMU_CENTER_PWRUP_CNT	0x00b8	W	0x00005dc0	pmu center power up count register
PMU_TIMEOUT_CNT	0x00bc	W	0x00000000	pmu timeout count register
PMU_CPU0APM_CON	0x00c0	W	0x00000000	pmu cpu0 auto power down control register

Name	Offset	Size	Reset Value	Description
PMU_CPU1APM_CON	0x00c4	W	0x00000000	pmu cpu1 auto power down control register
PMU_CPU2APM_CON	0x00c8	W	0x00000000	pmu cpu2 auto power down control register
PMU_CPU3APM_CON	0x00cc	W	0x00000000	pmu cpu3 auto power down control register
PMU_CPU0BPM_CON	0x00d0	W	0x00000000	pmu cluster_b cpu0 auto power down control register
PMU_CPU1BPM_CON	0x00d4	W	0x00000000	pmu cluster_b cpu0 auto power down control register
PMU_NOC_AUTO_ENA	0x00d8	W	0x00000000	NOC auto domain clock gating disable enable register
PMU_PWRDN_CON1	0x00dc	W	0x00000000	pmu power down configure register1
PMU_SYS_REG0	0x00f0	W	0x00000000	pmu system register 0
PMU_SYS_REG1	0x00f4	W	0x00000000	pmu system register 1
PMU_SYS_REG2	0x00f8	W	0x00000000	pmu system register 2
PMU_SYS_REG3	0x00fc	W	0x00000000	pmu system register 3

*Notes:Size:**B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access*

9.4.2 Detail Register Description

PMU_WAKEUP_CFG0

Address: Operational Base + offset (0x0000)
pmu wakeup configure register 0

Bit	Attr	Reset Value	Description
31:24	RW	0x00	gpio0d_posedge_en gpio0d posedge pulse wakeup enable 0: disable 1: enable
23:16	RW	0x00	gpio0c_posedge_en gpio0c posedge pulse wakeup enable 0: disable 1: enable
15:8	RW	0x00	gpio0b_posedge_en gpio0b posedge pulse wakeup enable 0: disable 1: enable
7:0	RW	0x00	gpio0a_posedge_en gpio0a posedge pulse wakeup enable 0: disable 1: enable

PMU_WAKEUP_CFG1

Address: Operational Base + offset (0x0004)
pmu wakeup configure register 1

Bit	Attr	Reset Value	Description
31:24	RW	0x00	gpio0d_negedge_en gpio0d negedge pulse wakeup enable 0: disable 1: enable
23:16	RW	0x00	gpio0c_negedge_en gpio0c negedge pulse wakeup enable 0: disable 1: enable
15:8	RW	0x00	gpio0b_negedge_en gpio0b negedge pulse wakeup enable 0: disable 1: enable
7:0	RW	0x00	gpio0a_negedge_en gpio0a negedge pulse wakeup enable 0: disable 1: enable

PMU_WAKEUP_CFG2
Address: Operational Base + offset (0x0008)
pmu wakeup configure register 2

Bit	Attr	Reset Value	Description
31:24	RW	0x00	gpio1d_posedge_en gpio1d posedge pulse wakeup enable 0: disable 1: enable
23:16	RW	0x00	gpio1c_posedge_en gpio1c posedge pulse wakeup enable 0: disable 1: enable
15:8	RW	0x00	gpio1b_posedge_en gpio1b posedge pulse wakeup enable 0: disable 1: enable
7:0	RW	0x00	gpio1a_posedge_en gpio1a posedge pulse wakeup enable 0: disable 1: enable

PMU_WAKEUP_CFG3
Address: Operational Base + offset (0x000c)
pmu wakeup configure register 3

Bit	Attr	Reset Value	Description
31:24	RW	0x00	gpio1d_negedge_en gpio1d negedge pulse wakeup enable 0: disable 1: enable
23:16	RW	0x00	gpio1c_negedge_en gpio1c negedge pulse wakeup enable 0: disable 1: enable
15:8	RW	0x00	gpio1b_negedge_en gpio1b negedge pulse wakeup enable 0: disable 1: enable
7:0	RW	0x00	gpio1a_negedge_en gpio1a negedge pulse wakeup enable 0: disable 1: enable

PMU_WAKEUP_CFG4
Address: Operational Base + offset (0x0010)
pmu wakeup configure register 4

Bit	Attr	Reset Value	Description
31:14	RO	0x0	reserved
13	RW	0x0	pcie_en pcie interrupt wakeup enable 0: disable 1: enable
12	RO	0x0	reserved
11	RW	0x0	pwm_en pwm interrupt wakeup enable 0: disable 1: enable
10	RW	0x0	timeout_en pmu time out wakeup enable 0: disable 1: enable
9	RW	0x0	wdt_m0_en m3 watch dog wakeup enable 0: disable 1: enable
8	RW	0x0	sft_en software wakeup enable 0: disable 1: enable

Bit	Attr	Reset Value	Description
7	RW	0x0	usbdev_en usb device detect wakeup enable 0: disable 1: enable
6	RW	0x0	timer_en timer wakeup enable 0: disable 1: enable
5	RO	0x0	reserved
4	RW	0x0	sdmmc_en sdmmc detect wakeup enable 0: disable 1: enable
3	RW	0x0	sdio_en sdio detect wakeup enable 0: disable 1: enable
2	RW	0x0	gpio_int_en gpio interrupt wakeup enable 0: disable 1: enable
1	RW	0x0	int_cluster_b_en cluster_b interrupt wakeup enable 0: disable 1: enable
0	RW	0x0	int_cluster_l_en cluster_l interrupt wakeup enable 0: disable 1: enable

PMU_PWRDN_CON

Address: Operational Base + offset (0x0014)

pmu power down configure register

Bit	Attr	Reset Value	Description
31	RW	0x0	pd_sdioaudio_pwrdwn_en pd_sdioaudio power down enable 0: disable 1: enable
30	RW	0x0	pd_sd_pwrdwn_en pd_sd power down enable 0: disable 1: enable

Bit	Attr	Reset Value	Description
29	RW	0x0	pd_gic_pwrdwn_en pd_gic power down enable 0: disable 1: enable
28	RW	0x0	pd_edp_pwrdwn_en pd_edp power down enable 0: disable 1: enable
27	RW	0x0	pd_usb3_pwrdwn_en pd_usb3 power down enable 0: disable 1: enable
26	RW	0x0	pd_emmc_pwrdwn_en pd_emmc power down enable 0: disable 1: enable
25	RW	0x0	pd_gmac_pwrdwn_en pd_gmac power down enable 0: disable 1: enable
24	RW	0x0	pd_hdcp_pwrdwn_en pd_hdcp power down enable 0: disable 1: enable
23	RW	0x0	pd_isp1_pwrdwn_en pd_isp1 power down enable 0: disable 1: enable
22	RW	0x0	pd_isp0_pwrdwn_en pd_isp0 power down enable 0: disable 1: enable
21	RO	0x0	reserved
20	RW	0x0	pd_vo_pwrdwn_en pd_vo power down enable 0: disable 1: enable
19	RW	0x0	pd_iep_pwrdwn_en pd_perihp power down enable 0: disable 1: enable
18	RW	0x0	pd_rga_pwrdwn_en pd_rga power down enable 0: disable 1: enable

Bit	Attr	Reset Value	Description
17	RW	0x0	pd_vdu_pwrdwn_en pd_vdu power down enable 0: disable 1: enable
16	RW	0x0	pd_vcodec_pwrdwn_en pd_perihp power down enable 0: disable 1: enable
15	RW	0x0	pd_gpu_pwrdwn_en pd_gpu power down enable 0: disable 1: enable
14	RW	0x0	pd_vio_pwrdwn_en pd_vio power down enable 0: disable 1: enable
13	RW	0x0	pd_center_pwrdwn_en pd_center power down enable 0: disable 1: enable
12	RW	0x0	pd_perihp_pwrdwn_en pd_perihp power down enable 0: disable 1: enable
11	RW	0x0	pd_perilp_pwrdwn_en pd_perilp power down enable 0: disable 1: enable
10	RW	0x0	pd_cci_pwrdwn_en pd_cci power down enable 0: disable 1: enable
9	RW	0x0	pd_tcpd1_pwrdwn_en pd_tcpd1 power down enable 0: disable 1: enable
8	RW	0x0	pd_tcpd0_pwrdwn_en pd_tcpd0 power down enable 0: disable 1: enable
7	RW	0x0	pd_scu_b_pwrdwn_en pd_scu_b power down enable 0: disable 1: enable

Bit	Attr	Reset Value	Description
6	RW	0x0	pd_scu_l_pwrdwn_en pd_scu_l power down enable 0: disable 1: enable
5	RW	0x0	pd_a72_b1_pwrdwn_en pd_a72_b0 power down enable 0: disable 1: enable
4	RW	0x0	pd_a72_b0_pwrdwn_en pd_a72_b0 power down enable 0: disable 1: enable
3	RW	0x0	pd_a53_l3_pwrdwn pd_a53_l3 power down enable 0: disable 1: enable
2	RW	0x0	pd_a53_l2_pwrdwn pd_a53_l2 power down enable 0: disable 1: enable
1	RW	0x0	pd_a53_l1_pwrdwn pd_a53_l1 power down enable 0: disable 1: enable
0	RW	0x0	pd_a53_l0_pwrdwn_en pd_a53_l0 power down enable 0: disable 1: enable

PMU_PWRDN_ST

Address: Operational Base + offset (0x0018)

pmu power down status register

Bit	Attr	Reset Value	Description
31	RW	0x0	pd_sdioaudio_pwr_stat pd_sdioaudio power state 0: powered up 1: powered down
30	RW	0x0	pd_sd_pwr_stat pd_sd power state 0: powered up 1: powered down
29	RW	0x0	pd_gic_pwr_stat pd_gic power state 0: powered up 1: powered down

Bit	Attr	Reset Value	Description
28	RW	0x0	pd_edp_pwr_stat pd_edp power state 0: powered up 1: powered down
27	RW	0x0	pd_usb3_pwr_stat pd_usb3 power state 0: powered up 1: powered down
26	RW	0x0	pd_emmc_pwr_stat pd_emmc power state 0: powered up 1: powered down
25	RW	0x0	pd_gmac_pwr_stat pd_gmac power state 0: powered up 1: powered down
24	RW	0x0	pd_hdcp_pwr_stat pd_hdcp power state 0: powered up 1: powered down
23	RW	0x0	pd_isp1_pwr_stat pd_isp1 power state 0: powered up 1: powered down
22	RW	0x0	pd_isp0_pwr_stat pd_isp0 power state 0: powered up 1: powered down
21	RO	0x0	reserved
20	RW	0x0	pd_vo_pwr_stat pd_vo power state 0: powered up 1: powered down
19	RW	0x0	pd_iep_pwr_stat pd_iep power state 0: powered up 1: powered down
18	RW	0x0	pd_rga_pwr_stat pd_rga power state 0: powered up 1: powered down
17	RW	0x0	pd_vdu_pwr_stat pd_vdu power state 0: powered up 1: powered down

Bit	Attr	Reset Value	Description
16	RW	0x0	pd_vcodec_pwr_stat pd_vcodec power state 0: powered up 1: powered down
15	RW	0x0	pd_gpu_pwr_stat pd_gpu power state 0: powered up 1: powered down
14	RW	0x0	pd_vio_pwr_stat pd_vio power state 0: powered up 1: powered down
13	RW	0x0	pd_center_pwr_stat pd_center power state 0: powered up 1: powered down
12	RW	0x0	pd_perihp_pwr_stat pd_peri power state 0: powered up 1: powered down
11	RW	0x0	pd_perilp_pwr_stat pd_bus power stat 0: powered up 1: powered down
10	RW	0x0	pd_cci_pwr_stat pd_core power state 0: powered up 1: powered down
9	RW	0x0	pd_tcpd1_pwr_stat pd_tcpd1 power state 0: powered up 1: powered down
8	RW	0x0	pd_tcpd0_pwr_stat pd_tcpd0 power state 0: powered up 1: powered down
7	RO	0x0	pd_scu_b_pwr_stat pd_scu_b power state 0: powered up 1: powered down
6	RO	0x0	pd_scu_l_pwr_stat pd_scu_l power state 0: powered up 1: powered down

Bit	Attr	Reset Value	Description
5	RO	0x0	pd_a72_b1_pwr_stat pd_a72_b1 power state 0: powered up 1: powered down
4	RO	0x0	pd_a72_b0_pwr_stat pd_a72_b0 power state 0: powered up 1: powered down
3	RO	0x0	pd_a53_l3_pwr_stat pd_a53_l3 power state 0: powered up 1: powered down
2	RO	0x0	pd_a53_l2_pwr_stat pd_a53_l2 power state 0: powered up 1: powered down
1	RO	0x0	pd_a53_l1_pwr_stat pd_a53_l1 power state 0: powered up 1: powered down
0	RO	0x0	pd_a53_l0_pwr_stat pd_a53_l0 power state 0: powered up 1: powered down

PMU_PLL_CON
Address: Operational Base + offset (0x001c)

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:8	RW	0x00	sft_pll_pd pll power down configured by sftware.
7:0	RW	0x00	pll_pd_cfg pll power down configured by hardware

PMU_PWRMODE_CON
Address: Operational Base + offset (0x0020)
pmu power mode configure register of common resource

Bit	Attr	Reset Value	Description
31	RW	0x0	main_cluster use core big for main cluster. 0: core_l ; 1: core_b.
30	RW	0x0	sleep_output_cfg output pmu_sleep instead of ap_pwroff to IO.

Bit	Attr	Reset Value	Description
29	RW	0x0	ddrio_ret_hw_de_req hardware ddrio retention de-assert request
28	RW	0x0	clk_core_src_gate_en cpu clock gate enable when in power mode 0: disable 1: enable
27	RW	0x0	clk_perilp_src_gate_en pd_perilp clock gate enable when in power mode 0: disable 1: enable
26	RW	0x0	clk_center_src_gate_en pd_center clock gate enable when in power mode 0: disable 1: enable
25:24	RO	0x0	reserved
23	W1C	0x0	ddrio1_ret_de_req ddrio1 retention de-assert request write one clear
22	RW	0x0	ddrio1_ret_en ddrio1 retention enable when in power mode 0: disable 1: enable
21	RW	0x0	ddrc1_gating_en ddr1 controller auto gating when in power mode 0: disable 1: enable
20	RW	0x0	sref1_enter_en ddr1 self_refresh by hardware when in power mode 0: disable 1: enable
19	W1C	0x0	ddrio0_ret_de_req ddrio0 retention de-assert request write one clear
18	RW	0x0	ddrio0_ret_en ddrio0 retention enable when in power mode 0: disable 1: enable
17	RW	0x0	ddrc0_gating_en ddr0 controller auto gating when in power mode 0: disable 1: enable
16	RW	0x0	sref0_enter_en ddr0 self_refresh by hardware when in power mode 0: disable 1: enable

Bit	Attr	Reset Value	Description
15	RW	0x0	center_pd_en power down pd_center when power mode 0: disable 1: enable
14	RW	0x0	perilp_pd_en power down pd_perilp when power mode 0: disable 1: enable
13	RW	0x0	cci_pd_en power down pd_cci when power mode 0: disable 1: enable
12	RW	0x0	scu_pd_en power down main cluster scu when in power mode 0: disable 1: enable
11	RW	0x0	l2_idle_en wait l2 idle when in power mode 0: disable 1: enable
10	RW	0x0	l2_flush_en flush l2 by hardware when in power mode 0: disable 1: enable
9	RW	0x0	cpu0_pd_en power down core0 of cluster_l in power mode 0: disable 1: enable
8	RW	0x0	pll_pd_en power down pll when in power mode 0: disable 1: enable
7	RW	0x0	chip_pd_en chip power down enable 0: disable 1: enable
6	RW	0x0	power_off_req_cfg send power off request to PMIC when in power mode 0: disable 1: enable
5	RW	0x0	pmu_use_lf pmu low frequency mode enable when in power mode 0: disable 1: enable

Bit	Attr	Reset Value	Description
4	RW	0x0	alive_use_lf alive low frequency mode when in power mode 0: disable 1: enable
3	RW	0x0	osc_disable osc disable when in power mode 1: disable 0: enable
2	RW	0x0	input_clamp_en clamp vd_logic when in power mode 0: disable 1: enable
1	RW	0x0	wakeup_reset_en wakeup reset enable when in power mode 0: disable 1: enable
0	RW	0x0	power_mode_en enter power mode enable, will auto self-clear when in power mode 0: disable 1: enable

PMU_SFT_CON
Address: Operational Base + offset (0x0024)
pmu software configure register

Bit	Attr	Reset Value	Description
31	RW	0x0	acinactm_cluster_b_cfg acinactm indicate to cluster_b 0: acinactm to cluster_b is 0 1: acinactm to cluster_b is 1
30	RW	0x0	l2flushreq_cluster_b send l2 flush request to cluster_l by software 0: disable 1: enable
29	RW	0x0	cluster_b_clk_src_gating_cfg cluster_b clock source gating configure 0: disable 1: enable
28	RW	0x0	dbgpwrupreq_b_en dbg powered up request function of cluster_b enable 0: disable 1: enable
27:26	RO	0x0	reserved

Bit	Attr	Reset Value	Description
25:24	RW	0x0	dbgnopwrdwn_b_enable dbgnopowerdown function of cluster_b enable 0: disable 1: enable
23	RW	0x0	acinactm_cluster_l_cfg acinactm indicate to cluster_l 0: acinactm to cluster_l is 0 1: acinactm to cluster_l is 1
22	RW	0x0	l2flushreq_cluster_l send l2 flush request to cluster_l by software 0: disable 1: enable
21	RW	0x0	cluster_l_clk_src_gating_cfg cluster_l clock source gating configure 0: disable 1: enable
20	RW	0x0	dbgpwrupreq_l_en dbg powered up request function of cluster_l enable 0: disable 1: enable
19:16	RW	0x0	dbgnopwrdwn_l_enable dbgnopowerdown function of cluster_l enable 0: disable 1: enable
15	RW	0x0	dbgpwrdup_b0_cfg dbg powered up of pd_a72_b0 enable when in power mode 0: disable 1: enable
14	RO	0x0	reserved
13	RW	0x0	ddr1_io_ret_cfg ddr1 io retention configure by software 0: disable 1: enable
12	RW	0x0	ddrctl1_c_sysreq_cfg ddrctl1 idle request configure 0: disable 1: enable
11:10	RO	0x0	reserved
9	RW	0x0	ddr0_io_ret_cfg ddr0 io retention configure by software 0: disable 1: enable

Bit	Attr	Reset Value	Description
8	RW	0x0	ddrctl0_c_sysreq_cfg ddrctl idle request configure 0: disable 1: enable
7	RW	0x0	wakeup_sft_m0 m0 configure this bit to wakeup PMU state machine.
6	RW	0x0	dbgpwrdup_l0_cfg dbg powered up of pd_a53_l0 enable when in power mode 0: disable 1: enable
5	RW	0x0	pmu_24m_ena_cfg configure PD_PMU use 24M clock
4	RW	0x0	alive_lf_ena_cfg pd_alive low frequency mode configure by software 0: disable 1: enable
3	RW	0x0	pmu_lf_ena_cfg pd_pmu low frequency mode configure by software 0: disable 1: enable
2	RW	0x0	osc_disable_cfg osc disable configure by software 1: disable osc 0: enable psc
1	RW	0x0	input_clamp_cfg software control of input clamp signal
0	RW	0x0	wakeup_sft software wakeup request bit A 0 to 1 pulse posedge will wakeup pmu when in low power mode

PMU_INT_CON

Address: Operational Base + offset (0x0028)

pmu interrupt configure register

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5	RW	0x0	wakeup_gpio1_pos_int_en gpio1 posedge wakeup interrupt enable 0: disable 1: enable
4	RW	0x0	wakeup_gpio1_neg_int_en gpio1 negedge wakeup interrupt enable 0: disable 1: enable

Bit	Attr	Reset Value	Description
3	RW	0x0	wakeup_gpio0_pos_int_en gpio posedge wakeup interrupt enable 0: disable 1: enable
2	RW	0x0	wakeup_gpio0_neg_int_en gpio0 negedge wakeup interrupt enable 0: disable 1: enable
1	RW	0x0	pwrmode_wakeup_int_en power mode wakeup interrupt enable 0: disable 1: enable
0	RW	0x0	pmu_int_en global interrupt enable 0: disable 1: enable

PMU_INT_ST
Address: Operational Base + offset (0x002c)
pmu interrupt status register

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5	RW	0x0	wakeup_gpio1_neg_status gpio1 negedge pulse wakeup status 0: not wakeup by gpio1 negedge pulse 1: wakeup by gpio1 negedge pulse
4	RW	0x0	wakeup_gpio1_pos_status gpio1 posedge pulse wakeup status 0: not wakeup by gpio1 posedge pulse 1: wakeup by gpio1 posedge pulse
3	RW	0x0	wakeup_gpio0_pos_status gpio0 posedge pulse wakeup status 0: not wakeup by gpio0 posedge pulse 1: wakeup by gpio0 posedge pulse
2	RW	0x0	wakeup_gpio0_neg_status gpio0 negedge pulse wakeup status 0: not wakeup by gpio negedge pulse 1: wakeup by gpio negedge pulse
1	RW	0x0	pwrmode_wakeup_status power mode wakeup status 0: not wakeup from power mode 1: wakeup from power mode
0	RO	0x0	reserved

PMU_GPIO0_POS_INT_CON

Address: Operational Base + offset (0x0030)
pmu gpio0 posedge interrupt configure register

Bit	Attr	Reset Value	Description
31:24	RW	0x00	gpio0d_pos_int_en gpio0d posedge pulse interrupt enable 0: disable 1: enable
23:16	RW	0x00	gpio0c_pos_int_en gpio0c posedge pulse interrupt enable 0: disable 1: enable
15:8	RW	0x00	gpio0b_pos_int_en gpio0b posedge pulse interrupt enable 0: disable 1: enable
7:0	RW	0x00	gpio0a_pos_int_en gpio0a posedge pulse interrupt enable 0: disable 1: enable

PMU_GPIO0_NEG_INT_CON
Address: Operational Base + offset (0x0034)
pmu gpio0 negedge interrupt configure register

Bit	Attr	Reset Value	Description
31:24	RW	0x00	gpio0d_neg_int_en gpio0d negedge pulse interrupt enable 0: disable 1: enable
23:16	RW	0x00	gpio0c_neg_int_en gpio0c negedge pulse interrupt enable 0: disable 1: enable
15:8	RW	0x00	gpio0b_neg_int_en gpio0b negedge pulse interrupt enable 0: disable 1: enable
7:0	RW	0x00	gpio0a_neg_int_en gpio0a negedge pulse interrupt enable 0: disable 1: enable

PMU_GPIO1_POS_INT_CON
Address: Operational Base + offset (0x0038)
pmu gpio1 posedge interrupt configure register

Bit	Attr	Reset Value	Description
31:24	RW	0x00	gpio1d_pos_int_en gpio1d posedge pulse interrupt enable 0: disable 1: enable
23:16	RW	0x00	gpio1c_pos_int_en gpio1c posedge pulse interrupt enable 0: disable 1: enable
15:8	RW	0x00	gpio1b_pos_int_en gpio1b posedge pulse interrupt enable 0: disable 1: enable
7:0	RW	0x00	gpio1a_pos_int_en gpio1a posedge pulse interrupt enable 0: disable 1: enable

PMU_GPIO1_NEG_INT_CON
Address: Operational Base + offset (0x003c)
pmu gpio1 negedge interrupt configure register

Bit	Attr	Reset Value	Description
31:24	RW	0x00	gpio1d_neg_int_en gpio1d negedge pulse interrupt enable 0: disable 1: enable
23:16	RW	0x00	gpio1c_neg_int_en gpio1c negedge pulse interrupt enable 0: disable 1: enable
15:8	RW	0x00	gpio1b_neg_int_en gpio1b negedge pulse interrupt enable 0: disable 1: enable
7:0	RW	0x00	gpio1a_neg_int_en gpio1a negedge pulse interrupt enable 0: disable 1: enable

PMU_GPIO0_POS_INT_ST
Address: Operational Base + offset (0x0040)
pmu gpio0 posedge interrupt status register

Bit	Attr	Reset Value	Description
31:24	RW	0x00	gpio0d_pos_int_status gpio0d posedge pulse interrupt status 0: not wakeup by gpio0d posedge pulse 1: wakeup by gpio0d posedge pulse
23:16	RW	0x00	gpio0c_pos_int_status gpio0c posedge pulse interrupt status 0: not wakeup by gpio0c posedge pulse 1: wakeup by gpio0c posedge pulse
15:8	RW	0x00	gpio0b_pos_int_status gpio0b posedge pulse interrupt status 0: not wakeup by gpio0b posedge pulse 1: wakeup by gpio0b posedge pulse
7:0	RW	0x00	gpio0a_pos_int_status gpio0a posedge pulse interrupt status 0: not wakeup by gpio0a posedge pulse 1: wakeup by gpio0a posedge pulse

PMU_GPIO0_NEG_INT_ST
Address: Operational Base + offset (0x0044)
pmu gpio0 negedge interrupt status register

Bit	Attr	Reset Value	Description
31:24	RW	0x00	gpio0d_neg_int_status gpio0d negedge pulse interrupt status 0: not wakeup by gpio0d negedge pulse 1: wakeup by gpio0d negedge pulse
23:16	RW	0x00	gpio0c_neg_int_status gpio0c negedge pulse interrupt status 0: not wakeup by gpio0c negedge pulse 1: wakeup by gpio0c negedge pulse
15:8	RW	0x00	gpio0b_neg_int_status gpio0b negedge pulse interrupt status 0: not wakeup by gpio0b negedge pulse 1: wakeup by gpio0b negedge pulse
7:0	RW	0x00	gpio0a_neg_int_status gpio0a negedge pulse interrupt status 0: not wakeup by gpio0a negedge pulse 1: wakeup by gpio0a negedge pulse

PMU_GPIO1_POS_INT_ST
Address: Operational Base + offset (0x0048)
pmu gpio1 posedge interrupt status register

Bit	Attr	Reset Value	Description
31:24	RW	0x00	gpio1d_pos_int_status gpio1d posedge pulse interrupt status 0: not wakeup by gpio1d posedge pulse 1: wakeup by gpio1d posedge pulse
23:16	RW	0x00	gpio1c_pos_int_status gpio1c posedge pulse interrupt status 0: not wakeup by gpio1c posedge pulse 1: wakeup by gpio1c posedge pulse
15:8	RW	0x00	gpio1b_pos_int_status gpio1b posedge pulse interrupt status 0: not wakeup by gpio1b posedge pulse 1: wakeup by gpio1b posedge pulse
7:0	RW	0x00	gpio1a_pos_int_status gpio1a posedge pulse interrupt status 0: not wakeup by gpio1a posedge pulse 1: wakeup by gpio1a posedge pulse

PMU_GPIO1_NEG_INT_ST
Address: Operational Base + offset (0x004c)
pmu gpio1 negedge interrupt status register

Bit	Attr	Reset Value	Description
31:24	RW	0x00	gpio1d_neg_int_status gpio1d negedge pulse interrupt status 0: not wakeup by gpio1d negedge pulse 1: wakeup by gpio1d negedge pulse
23:16	RW	0x00	gpio1c_neg_int_status gpio1c negedge pulse interrupt status 0: not wakeup by gpio1c negedge pulse 1: wakeup by gpio1c negedge pulse
15:8	RW	0x00	gpio1b_neg_int_status gpio1b negedge pulse interrupt status 0: not wakeup by gpio1b negedge pulse 1: wakeup by gpio1b negedge pulse
7:0	RW	0x00	gpio1a_neg_int_status gpio1a negedge pulse interrupt status 0: not wakeup by gpio1a negedge pulse 1: wakeup by gpio1a negedge pulse

PMU_PWRDN_INTEN
Address: Operational Base + offset (0x0050)
pmu power down configure register

Bit	Attr	Reset Value	Description
31	RW	0x0	pd_sdioaudio_pwr_switch_int_en pd_sdioaudio power switch interrupt enable 0: disable 1: enable
30	RW	0x0	pd_sd_pwr_switch_int_en pd_sd power switch interrupt enable 0: disable 1: enable
29	RW	0x0	pd_gic_pwr_switch_int_en pd_gic power switch interrupt enable 0: disable 1: enable
28	RW	0x0	pd_edp_pwr_switch_int_en pd_edp power switch interrupt enable 0: disable 1: enable
27	RW	0x0	pd_usb3_pwr_switch_interrupt_en pd_usb3 power switch interrupt enable 0: disable 1: enable
26	RW	0x0	pd_emmc_pwr_switch_interrupt_en pd_emmc power switch interrupt enable 0: disable 1: enable
25	RW	0x0	pd_gmac_pwr_switch_int_en pd_gmac power switch interrupt enable 0: disable 1: enable
24	RW	0x0	pd_hdcp_pwr_switch_int_en pd_hdcp power switch interrupt enable 0: disable 1: enable
23	RW	0x0	pd_isp1_pwr_switch_int_en pd_isp1 power switch interrupt enable 0: disable 1: enable
22	RW	0x0	pd_isp0_pwr_switch_int_en pd_isp0 power switch interrupt enable 0: disable 1: enable
21	RO	0x0	reserved
20	RW	0x0	pd_vo_pwr_switch_int_en pd_vo power switch interrupt enable 0: disable 1: enable

Bit	Attr	Reset Value	Description
19	RW	0x0	pd_iep_pwr_switch_int_en pd_perihp power switch interrupt enable 0: disable 1: enable
18	RW	0x0	pd_rga_pwr_switch_int_en pd_rga power switch interrupt enable 0: disable 1: enable
17	RW	0x0	pd_vdu_pwr_switch_int_en pd_vdu power switch interrupt enable 0: disable 1: enable
16	RW	0x0	pd_vcodec_pwr_switch_inten pd_perihp power switch interrupt enable 0: disable 1: enable
15	RW	0x0	pd_gpu_pwr_switch_int_en pd_gpu power interrupt enable 0: disable 1: enable
14	RW	0x0	pd_vio_pwr_switch_int_en pd_vio power switch interrupt enable 0: disable 1: enable
13	RW	0x0	pd_center_pwr_switch_int_en pd_center power switch interrupt enable 0: disable 1: enable
12	RW	0x0	pd_perihp_pwr_switch_int_en pd_perihp power switch interrupt enable 0: disable 1: enable
11	RW	0x0	pd_perilp_pwr_switch_int_en pd_perilp power switch interrupt enable 0: disable 1: enable
10	RW	0x0	pd_cci_pwr_switch_int_en pd_cci power switch interrupt enable 0: disable 1: enable
9	RW	0x0	pd_tcpd1_pwr_switch_int_en pd_tcpd1 power switch interrupt enable 0: disable 1: enable

Bit	Attr	Reset Value	Description
8	RW	0x0	pd_tcpd0_pwr_switch_int_en pd_tcpd0 power switch interrupt enable 0: disable 1: enable
7	RW	0x0	pd_scu_b_pwr_switch_int_en pd_scu_b power switch interrupt enable 0: disable 1: enable
6	RW	0x0	pd_scu_l_pwr_switch_int_en pd_scu_l power switch interrupt enable 0: disable 1: enable
5	RW	0x0	pd_a72_b1_pwr_switch_int_en pd_a72_b1 power switch interrupt enable 0: disable 1: enable
4	RW	0x0	pd_a72_b0_pwr_switch_int_en pd_a72_b0 power enable 0: disable 1: enable
3	RW	0x0	pd_a53_l3_pwr_switch_int_en pd_a53_l3 power switch int enable 0: disable 1: enable
2	RW	0x0	pd_a53_l2_pwr_switch_int_en pd_a53_l2 power switch interrupt enable 0: disable 1: enable
1	RW	0x0	pd_a53_l1_pwr_switch_int_en pd_a53_l1 power switch interrupt enable 0: disable 1: enable
0	RW	0x0	pd_a53_l0_pwr_switch_int_en pd_a53_l0 power switch interrupt enable 0: disable 1: enable

PMU_PWRDN_STATUS
Address: Operational Base + offset (0x0054)
pmu power down status register

Bit	Attr	Reset Value	Description
31	W1C	0x0	pd_sdioaudio_pwr_stat pd_sdioaudio power state 0: powered up 1: powered down

Bit	Attr	Reset Value	Description
30	W1C	0x0	pd_sd_pwr_stat pd_sd power state 0: powered up 1: powered down
29	W1C	0x0	pd_gic_pwr_stat pd_gic power state 0: powered up 1: powered down
28	W1C	0x0	pd_edp_pwr_stat pd_edp power state 0: powered up 1: powered down
27	W1C	0x0	pd_usb3_pwr_stat pd_usb3 power state 0: powered up 1: powered down
26	W1C	0x0	pd_emmc_pwr_stat pd_emmc power state 0: powered up 1: powered down
25	W1C	0x0	pd_gmac_pwr_stat pd_gmac power state 0: powered up 1: powered down
24	W1C	0x0	pd_hdcp_pwr_stat pd_hdcp power state 0: powered up 1: powered down
23	W1C	0x0	pd_isp1_pwr_stat pd_isp1 power state 0: powered up 1: powered down
22	W1C	0x0	pd_isp0_pwr_stat pd_isp0 power state 0: powered up 1: powered down
21	RO	0x0	reserved
20	W1C	0x0	pd_vo_pwr_stat pd_vo power state 0: powered up 1: powered down
19	W1C	0x0	pd_iep_pwr_stat pd_iep power state 0: powered up 1: powered down

Bit	Attr	Reset Value	Description
18	W1C	0x0	pd_rga_pwr_stat pd_rga power state 0: powered up 1: powered down
17	W1C	0x0	pd_vdu_pwr_stat pd_vdu power state 0: powered up 1: powered down
16	W1C	0x0	pd_vcodec_pwr_stat pd_vcodec power state 0: powered up 1: powered down
15	W1C	0x0	pd_gpu_pwr_stat pd_gpu power state 0: powered up 1: powered down
14	W1C	0x0	pd_vio_pwr_stat pd_vio power state 0: powered up 1: powered down
13	W1C	0x0	pd_center_pwr_stat pd_center power state 0: powered up 1: powered down
12	W1C	0x0	pd_perihp_pwr_stat pd_perihp power state 0: powered up 1: powered down
11	W1C	0x0	pd_perilp_pwr_stat pd_perilp power stat 0: powered up 1: powered down
10	W1C	0x0	pd_cci_pwr_stat pd_core power state 0: powered up 1: powered down
9	W1C	0x0	pd_tcpd1_pwr_stat pd_tcpd1 power state 0: powered up 1: powered down
8	W1C	0x0	pd_tcpd0_pwr_stat pd_tcpd0 power state 0: powered up 1: powered down

Bit	Attr	Reset Value	Description
7	W1C	0x0	pd_scu_b_pwr_stat pd_scu_b power state 0: powered up 1: powered down
6	W1C	0x0	pd_scu_l_pwr_stat pd_scu_l power state 0: powered up 1: powered down
5	W1C	0x0	pd_a72_b1_pwr_stat pd_a72_b1 power state 0: powered up 1: powered down
4	W1C	0x0	pd_a72_b0_pwr_stat pd_a72_b0 power state 0: powered up 1: powered down
3	W1C	0x0	pd_a53_l3_pwr_stat pd_a53_l3 power state 0: powered up 1: powered down
2	W1C	0x0	pd_a53_l2_pwr_stat pd_a53_l2 power state 0: powered up 1: powered down
1	W1C	0x0	pd_a53_l1_pwr_stat pd_a53_l1 power state 0: powered up 1: powered down
0	W1C	0x0	pd_a53_l0_pwr_stat pd_a53_l0 power state 0: powered up 1: powered down

PMU_WAKEUP_STATUS

Address: Operational Base + offset (0x0058)

pmu interrupt status register

Bit	Attr	Reset Value	Description
31:14	RO	0x0	reserved
13	RW	0x0	wakeup_pcie_status pcie wakeup status 0: not wakeup by pcie 1: wakeup by pcie
12	RO	0x0	reserved

Bit	Attr	Reset Value	Description
11	RW	0x0	wakeup_pwm_status pwm wakeup status 0: not wakeup by pwm 1: wakeup by pwm
10	RW	0x0	wakeup_timeout_status timeout wakeup status 0: not wakeup by timeout 1: wakeup by timeout
9	RW	0x0	wakeup_wdt_m0_status m0 wdt interrupt wakeup status 0: not wakeup by m0 wdt interrupt 1: wakeup by m0 wdt interrupt
8	RW	0x0	wakeup_sft_m0_status m0 software control wakeup status 0: not wakeup by software 1: wakeup by software
7	RW	0x0	wakeup_usbdev_status usbdev detect wakeup status 0: not wakeup by usbdev detect 1: wakeup by usbdev detect
6	RW	0x0	wakeup_timer_status timer wakeup status 0: not wakeup by timer 1: wakeup by timer
5	RO	0x0	reserved
4	RW	0x0	wakeup_sdmmc_status sdmmc wakeup status 0: not wakeup by sdmmc detect 1: wakeup by sdmmc detect
3	RW	0x0	wakeup_sdio_status sdio wakeup status 0: not wakeup by sdio detect 1: wakeup by sdio detect
2	RW	0x0	wakeup_gpio_int_status gpio interrupt wakeup status 0: not wakeup by gpio int 1: wakeup by gpio int
1	RW	0x0	wakeup_int_cluster_b_status cluster_b interrupt wakeup status 0: not wakeup by interrupt cluster_b 1: wakeup by interrupt cluster_b
0	RW	0x0	wakeup_int_cluster_l_status cluster_l interrupt wakeup status 0: not wakeup by interrupt cluster_l 1: wakeup by interrupt cluster_l

PMU_BUS_CLR
Address: Operational Base + offset (0x005c)
pmu bus clear register

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved
29	RW	0x0	clr_sdioaudio send idle request to sdioaudio low power interface 0: disable 1: enable
28	RW	0x0	clr_sd send idle request to sd low power interface 0: disable 1: enable
27	RW	0x0	clr_gic send idle request to gic low power interface 0: disable 1: enable
26	RW	0x0	clr_pmum0 send idle request to pmu m0 low power interface 0: disable 1: enable
25	RW	0x0	clr_center1 send idle request to center1 low power interface 0: disable 1: enable
24	RW	0x0	clr_emmc send idle request to emmc low power interface 0: disable 1: enable
23	RW	0x0	clr_gmac send idle request to gmac low power interface 0: disable 1: enable
22	RW	0x0	clr_edp send idle request to edp low power interface 0: disable 1: enable
21	RW	0x0	clr_pmu send idle request to pmu low power interface 0: disable 1: enable
20	RW	0x0	clr_alive send idle request to alive low power interface 0: disable 1: enable

Bit	Attr	Reset Value	Description
19	RW	0x0	clr_msch1 send idle request to msch1 low power interface 0: disable 1: enable
18	RW	0x0	clr_msch0 send idle request to msch0 low power interface 0: disable 1: enable
17	RW	0x0	clr_vio send idle request to vio low power interface 0: disable 1: enable
16	RW	0x0	clr_ccim0 send idle request to ccim0 low power interface 0: disable 1: enable
15	RW	0x0	clr_ccim1 send idle request to ccim1 low power interface 0: disable 1: enable
14	RW	0x0	clr_center send idle request to center niu 0: disable 1: enable
13	RW	0x0	clr_perilpm0 send idle request to perilp m0 niu 0: disable 1: enable
12	RW	0x0	clr_usb3 send idle request to usb3 niu 0: disable 1: enable
11	RW	0x0	clr_hdcp send idle request to hdcp niu 0: disable 1: enable
10	RW	0x0	clr_isp1 send idle request to isp1 niu 0: disable 1: enable
9	RW	0x0	clr_isp0 send idle request to isp0 niu 0: disable 1: enable

Bit	Attr	Reset Value	Description
8	RW	0x0	clr_vopl send idle request to vopl niu 0: disable 1: enable
7	RW	0x0	clr_vopb send idle request to vopb niu 0: disable 1: enable
6	RW	0x0	clr_iep send idle request to iep niu 0: disable 1: enable
5	RW	0x0	clr_rga send idle request to rga niu 0: disable 1: enable
4	RW	0x0	clr_vdu send idle request to vdu niu 0: disable 1: enable
3	RW	0x0	clr_vcodec send idle request to vcodec niu 0: disable 1: enable
2	RW	0x0	clr_perihp send idle request to perihp niu 0: disable 1: enable
1	RW	0x0	clr_perilp send idle request to perilp niu 0: disable 1: enable
0	RW	0x0	clr_gpu send idle request to gpu niu 0: disable 1: enable

PMU_BUS_IDLE_REQ
Address: Operational Base + offset (0x0060)
pmu bus idle request register

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved

Bit	Attr	Reset Value	Description
29	RW	0x0	idle_req_sdioaudio send idle request to sdioaudio low power interface 0: disable 1: enable
28	RW	0x0	idle_req_sd send idle request to sd low power interface 0: disable 1: enable
27	RW	0x0	idle_req_gic send idle request to gic low power interface 0: disable 1: enable
26	RW	0x0	idle_req_pmum0 send idle request to pmu m0 low power interface 0: disable 1: enable
25	RW	0x0	idle_req_center1 send idle request to center1 low power interface 0: disable 1: enable
24	RW	0x0	idle_req_emmc send idle request to emmc low power interface 0: disable 1: enable
23	RW	0x0	idle_req_gmac send idle request to gmac low power interface 0: disable 1: enable
22	RW	0x0	idle_req_edp send idle request to edp low power interface 0: disable 1: enable
21	RW	0x0	idle_req_pmu send idle request to pmu low power interface 0: disable 1: enable
20	RW	0x0	idle_req_alive send idle request to alive low power interface 0: disable 1: enable
19	RW	0x0	idle_req_msch1 send idle request to msch1 low power interface 0: disable 1: enable

Bit	Attr	Reset Value	Description
18	RW	0x0	idle_req_msch0 send idle request to msch0 low power interface 0: disable 1: enable
17	RW	0x0	idle_req_vio send idle request to vio low power interface 0: disable 1: enable
16	RW	0x0	idle_req_ccim1 send idle request to ccim1 low power interface 0: disable 1: enable
15	RW	0x0	idle_req_ccim0 send idle request to ccim0 low power interface 0: disable 1: enable
14	RW	0x0	idle_req_center send idle request to center niu 0: disable 1: enable
13	RW	0x0	idle_req_perilpm0 send idle request to perilp m0 niu 0: disable 1: enable
12	RW	0x0	idle_req_usb3 send idle request to usb3 niu 0: disable 1: enable
11	RW	0x0	idle_req_hdcp send idle request to hdcp niu 0: disable 1: enable
10	RW	0x0	idle_req_isp1 send idle request to isp1 niu 0: disable 1: enable
9	RW	0x0	idle_req_isp0 send idle request to isp0 niu 0: disable 1: enable
8	RW	0x0	idle_req_vopl send idle request to vopl niu 0: disable 1: enable

Bit	Attr	Reset Value	Description
7	RW	0x0	idle_req_vopb send idle request to vopb niu 0: disable 1: enable
6	RW	0x0	idle_req_iep send idle request to iep niu 0: disable 1: enable
5	RW	0x0	idle_req_rga send idle request to rga niu 0: disable 1: enable
4	RW	0x0	idle_req_vdu send idle request to vdu niu 0: disable 1: enable
3	RW	0x0	idle_req_vcodec send idle request to vcodec niu 0: disable 1: enable
2	RW	0x0	idle_req_perihp send idle request to perihp niu 0: disable 1: enable
1	RW	0x0	idle_req_perilp send idle request to perilp niu 0: disable 1: enable
0	RW	0x0	idle_req_gpu send idle request to gpu niu 0: disable 1: enable

PMU_BUS_IDLE_ST

Address: Operational Base + offset (0x0064)

pmu bus idle status register

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved
29	RO	0x0	idle_sdioaudio send idle request to sdioaudio low power interface 0: disable 1: enable

Bit	Attr	Reset Value	Description
28	RO	0x0	idle_sd send idle request to sd low power interface 0: disable 1: enable
27	RO	0x0	idle_gic idle status of gic niu 0: idle status of niu is 0 1: idle status of niu is 1
26	RO	0x0	idle_pmum0 idle status of pmu m0 niu 0: idle status of niu is 0 1: idle status of niu is 1
25	RO	0x0	idle_center1 idle status of center1 niu 0: idle status of niu is 0 1: idle status of niu is 1
24	RO	0x0	idle_emmc idle status of emmc niu 0: idle status of niu is 0 1: idle status of niu is 1
23	RO	0x0	idle_gmac idle status of gmac niu 0: idle status of niu is 0 1: idle status of niu is 1
22	RO	0x0	idle_edp idle status of edp niu 0: idle status of niu is 0 1: idle status of niu is 1
21	RO	0x0	idle_pmu idle status of pmu niu 0: idle status of niu is 0 1: idle status of niu is 1
20	RO	0x0	idle_alive idle status of alive niu 0: idle status of niu is 0 1: idle status of niu is 1
19	RO	0x0	idle_msch1 idle status of msch1 niu 0: idle status of niu is 0 1: idle status of niu is 1
18	RO	0x0	idle_msch0 idle status of msch0 niu 0: idle status of niu is 0 1: idle status of niu is 1

Bit	Attr	Reset Value	Description
17	RO	0x0	idle_vio idle status of vio niu 0: idle status of niu is 0 1: idle status of niu is 1
16	RO	0x0	idle_ccim1 idle status of ccim1 niu 0: idle status of niu is 0 1: idle status of niu is 1
15	RO	0x0	idle_ccim0 idle status of ccim0 niu 0: idle status of niu is 0 1: idle status of niu is 1
14	RO	0x0	idle_center idle status of center niu 0: idle status of niu is 0 1: idle status of niu is 1
13	RO	0x0	idle_perilpm0 idle status of perilpm0 niu 0: idle status of niu is 0 1: idle status of niu is 1
12	RO	0x0	idle_usb3 idle status of usb3 niu 0: idle status of niu is 0 1: idle status of niu is 1
11	RO	0x0	idle_hdcp idle status of hdcp niu 0: idle status of niu is 0 1: idle status of niu is 1
10	RO	0x0	idle_isp1 idle status of isp1 niu 0: idle status of niu is 0 1: idle status of niu is 1
9	RO	0x0	idle_isp0 idle status of isp0 niu 0: idle status of niu is 0 1: idle status of niu is 1
8	RO	0x0	idle_vopl idle status of vopl niu 0: idle status of niu is 0 1: idle status of niu is 1
7	RO	0x0	idle_vopb idle status of vopb niu 0: idle status of niu is 0 1: idle status of niu is 1

Bit	Attr	Reset Value	Description
6	RO	0x0	idle_iep idle status of iep niu 0: idle status of niu is 0 1: idle status of niu is 1
5	RO	0x0	idle_rga idle status of rga niu 0: idle status of niu is 0 1: idle status of niu is 1
4	RO	0x0	idle_vdu idle status of vdu niu 0: idle status of niu is 0 1: idle status of niu is 1
3	RO	0x0	idle_vcodec idle status of vcodec niu 0: idle status of niu is 0 1: idle status of niu is 1
2	RO	0x0	idle_perihp idle status of perihp niu 0: idle status of niu is 0 1: idle status of niu is 1
1	RO	0x0	idle_perilp idle status of perilp niu 0: idle status of niu is 0 1: idle status of niu is 1
0	RO	0x0	idle_gpu idle status of gpu niu 0: idle status of gpu_niu is 0 1: idle status of gpu_niu is 1

PMU_BUS_IDLE_ACK

Address: Operational Base + offset (0x0068)

pmu bus idle ack status register

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved
29	RO	0x0	idle_ack_sdioaudio idle acknowledge status from sdioaudio niu 0: idle acknowledge status of niu is 0 1: idle acknowledge status of niu is 1
28	RO	0x0	idle_ack_sd idle acknowledge status from sd niu 0: idle acknowledge status of niu is 0 1: idle acknowledge status of niu is 1

Bit	Attr	Reset Value	Description
27	RO	0x0	idle_ack_gic idle acknowledge status from gic niu 0: idle acknowledge status of niu is 0 1: idle acknowledge status of niu is 1
26	RO	0x0	idle_ack_pmum0 idle acknowledge status from pmu m0 niu 0: idle acknowledge status of niu is 0 1: idle acknowledge status of niu is 1
25	RO	0x0	idle_ack_center1 idle acknowledge status from center1 niu 0: idle acknowledge status of niu is 0 1: idle acknowledge status of niu is 1
24	RO	0x0	idle_ack_emmc idle acknowledge status from emmc niu 0: idle acknowledge status of niu is 0 1: idle acknowledge status of niu is 1
23	RO	0x0	idle_ack_gmac idle acknowledge status from gmac niu 0: idle acknowledge status of niu is 0 1: idle acknowledge status of niu is 1
22	RO	0x0	idle_ack_edp idle acknowledge status from edp niu 0: idle acknowledge status of niu is 0 1: idle acknowledge status of niu is 1
21	RO	0x0	idle_ack_pmu idle acknowledge status from pmu niu 0: idle acknowledge status of niu is 0 1: idle acknowledge status of niu is 1
20	RO	0x0	idle_ack_alive idle acknowledge status from alive niu 0: idle acknowledge status of niu is 0 1: idle acknowledge status of niu is 1
19	RO	0x0	idle_ack_msch1 idle acknowledge status from msch1 niu 0: idle acknowledge status of niu is 0 1: idle acknowledge status of niu is 1
18	RO	0x0	idle_ack_msch0 idle acknowledge status from msch0 niu 0: idle acknowledge status of niu is 0 1: idle acknowledge status of niu is 1
17	RO	0x0	idle_ack_vio idle acknowledge status from vio niu 0: idle acknowledge status of niu is 0 1: idle acknowledge status of niu is 1

Bit	Attr	Reset Value	Description
16	RO	0x0	idle_ack_ccim1 idle acknowledge status from ccim1 niu 0: idle acknowledge status of niu is 0 1: idle acknowledge status of niu is 1
15	RO	0x0	idle_ack_ccim0 idle acknowledge status from ccim0 niu 0: idle acknowledge status of niu is 0 1: idle acknowledge status of niu is 1
14	RO	0x0	idle_ack_center idle acknowledge status from center niu 0: idle acknowledge status of niu is 0 1: idle acknowledge status of niu is 1
13	RO	0x0	idle_ack_perilpm0 idle acknowledge status from perilp m0 niu 0: idle acknowledge status of niu is 0 1: idle acknowledge status of niu is 1
12	RO	0x0	idle_ack_usb3 idle acknowledge status from usb3 niu 0: idle acknowledge status of niu is 0 1: idle acknowledge status of niu is 1
11	RO	0x0	idle_ack_hdcp idle acknowledge status from hdcp niu 0: idle acknowledge status of niu is 0 1: idle acknowledge status of niu is 1
10	RO	0x0	idle_ack_isp1 idle acknowledge status from isp1 niu 0: idle acknowledge status of niu is 0 1: idle acknowledge status of niu is 1
9	RO	0x0	idle_ack_isp0 idle acknowledge status from isp0 niu 0: idle acknowledge status of niu is 0 1: idle acknowledge status of niu is 1
8	RO	0x0	idle_ack_vopl idle acknowledge status from vopl niu 0: idle acknowledge status of niu is 0 1: idle acknowledge status of niu is 1
7	RO	0x0	idle_ack_vopb idle acknowledge status from vopb niu 0: idle acknowledge status of niu is 0 1: idle acknowledge status of niu is 1
6	RO	0x0	idle_ack_iep idle acknowledge status from iep niu 0: idle acknowledge status of niu is 0 1: idle acknowledge status of niu is 1

Bit	Attr	Reset Value	Description
5	RO	0x0	idle_ack_rga idle acknowledge status from rga niu 0: idle acknowledge status of niu is 0 1: idle acknowledge status of niu is 1
4	RO	0x0	idle_ack_vdu idle acknowledge status from vdu niu 0: idle acknowledge status of niu is 0 1: idle acknowledge status of niu is 1
3	RO	0x0	idle_ack_vcodec idle acknowledge status from vcodec niu 0: idle acknowledge status of niu is 0 1: idle acknowledge status of niu is 1
2	RO	0x0	idle_ack_perihp idle acknowledge status from perihp niu 0: idle acknowledge status of niu is 0 1: idle acknowledge status of niu is 1
1	RO	0x0	idle_ack_perilp idle acknowledge status from perilp niu 0: idle acknowledge status of niu is 0 1: idle acknowledge status of niu is 1
0	RO	0x0	idle_ack_gpu idle acknowledge status from gpu niu 0: idle acknowledge status of niu is 0 1: idle acknowledge status of niu is 1

PMU_CCI500_CON
Address: Operational Base + offset (0x006c)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:8	RO	0x0	reserved
7	RW	0x0	qgating_cci500_cfg CCI-500 Q-channel clock gating enable.
6	RW	0x0	clr_qreq_cci500 CCI-500 Q-channel request sent by hardware.
5	RW	0x0	qreq_cci500_cfg CCI-500 Q-channel request sent by software.

Bit	Attr	Reset Value	Description
4:2	RW	0x0	pstate_cci500 CCI-500 P-channel pstate .
1	RW	0x0	clr_preq_cci500 CCI-500 P-channel request sent by hardware .
0	RW	0x0	preq_cci500_cfg CCI-500 P-channel request sent by software

PMU_ADB400_CON
Address: Operational Base + offset (0x0070)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15	RO	0x0	reserved
14	RW	0x0	clr_gic2_core_b hardware send idle request to path from gic to core_b low power interface 0: disable 1: enable
13	RW	0x0	clr_core_b_2gic hardware send idle request to path from core_b to gic low power interface 0: disable 1: enable
12	RW	0x0	clr_core_b hardware send idle request from core_b to cci low power interface 0: disable 1: enable
11	RW	0x0	clr_gic2_core_l hardware send idle request to path from gic to core_l low power interface 0: disable 1: enable
10	RW	0x0	clr_core_l_2gic hardware send idle request to path from core_l to gic low power interface 0: disable 1: enable

Bit	Attr	Reset Value	Description
9	RW	0x0	clr_core_l software send idle request from core_l to cci low power interface 0: disable 1: enable
8	RW	0x0	clr_cxcs hardware send idle request to cxcs low power interface 0: disable 1: enable
7	RO	0x0	reserved
6	RW	0x0	pwrdwn_req_gic2_core_b send idle request to path from gic to core_b low power interface 0: disable 1: enable
5	RW	0x0	pwrdwn_req_core_b_2gic software send idle request to path from core_b to gic low power interface 0: disable 1: enable
4	RW	0x0	pwrdwn_req_core_b software send idle request from core_b to cci low power interface 0: disable 1: enable
3	RW	0x0	pwrdwn_req_gic2_core_l send idle request to path from gic to core_l low power interface 0: disable 1: enable
2	RW	0x0	pwrdwn_req_core_l_2gic software send idle request to path from core_l to gic low power interface 0: disable 1: enable
1	RW	0x0	pwrdwn_req_core_l software send idle request from core_l to cci low power interface 0: disable 1: enable
0	RW	0x0	pwrdwn_req_cxcs software send idle request to cxcs low power interface 0: disable 1: enable

PMU_ADB400_ST
Address: Operational Base + offset (0x0074)

Bit	Attr	Reset Value	Description
31:15	RO	0x0	reserved

Bit	Attr	Reset Value	Description
14	RO	0x0	idle_gic2_core_b active status of cxcs low power interface 0: active status is 0 (inactive) 1: active status is 1 (active)
13	RO	0x0	idle_core_b_2gic active status of cxcs low power interface 0: active status is 0 (inactive) 1: active status is 1 (active)
12	RO	0x0	idle_core_b active status of cxcs low power interface 0: active status is 0 (inactive) 1: active status is 1 (active)
11	WO	0x0	idle_gic2_core_l active status of cxcs low power interface 0: active status is 0 (inactive) 1: active status is 1 (active)
10	RO	0x0	idle_core_l_2gic active status of cxcs low power interface 0: active status is 0 (inactive) 1: active status is 1 (active)
9	RO	0x0	idle_core_l active status of cxcs low power interface 0: active status is 0 (inactive) 1: active status is 1 (active)
8	RO	0x0	active_cxcs active status of cxcs low power interface 0: active status is 0 (inactive) 1: active status is 1 (active)
7	RO	0x0	reserved
6	RO	0x0	pwrdwn_ack_gic2_core_b idle acknowledge status from cxcs 0: idle acknowledge status of adb is 0 1: idle acknowledge status of adb is 1
5	RO	0x0	pwrdwn_ack_core_b_2gic idle acknowledge status from cxcs 0: idle acknowledge status of adb is 0 1: idle acknowledge status of adb is 1
4	RO	0x0	pwrdwn_ack_core_b idle acknowledge status from cxcs 0: idle acknowledge status of adb is 0 1: idle acknowledge status of adb is 1
3	RO	0x0	pwrdwn_ack_gic2_core_l idle acknowledge status from cxcs 0: idle acknowledge status of adb is 0 1: idle acknowledge status of adb is 1

Bit	Attr	Reset Value	Description
2	RO	0x0	pwrdwn_ack_core_l_2gic idle acknowledge status from cxcs 0: idle acknowledge status of adb is 0 1: idle acknowledge status of adb is 1
1	RO	0x0	pwrdwn_ack_core_l idle acknowledge status from cxcs 0: idle acknowledge status of adb is 0 1: idle acknowledge status of adb is 1
0	RO	0x0	pwrdwn_ack_cxcs idle acknowledge status from cxcs 0: idle acknowledge status of adb is 0 1: idle acknowledge status of adb is 1

PMU_POWER_ST
Address: Operational Base + offset (0x0078)
pmu power status register

Bit	Attr	Reset Value	Description
31:7	RO	0x0	reserved
6:0	RW	0x00	power_state power state of pmu FSM

PMU_CORE_PWR_ST
Address: Operational Base + offset (0x007c)
pmu core power status register

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved
29	RO	0x0	qacceptn_cci500 CCI-500 Q-channel accept signal, active low
28	RO	0x0	qdeny_cci500 CCI-500 Q-channel deny signal, active high
27	RO	0x0	qactive_cci500 CCI-500 Q-channel active signal, active high
26	RO	0x0	paccept_cci500 CCI-500 P-channel accept signal, active high
25	RO	0x0	pdeny_cci500 CCI-500 P-channel deny signal, active high
24:20	RO	0x00	pactive_cci500 CCI-500 P-channel active signal, active high
19:18	RO	0x0	reserved
17:16	RO	0x0	standbywfi_cluster_b standbywfi status of cluster_b 0: cluster_b standbywfi status is 0 1: cluster_b standbywfi status is 1
15:14	RO	0x0	reserved

Bit	Attr	Reset Value	Description
13:12	RO	0x0	standbywfe_cluster_b standbywfe status of cluster_b 0: cluster_b standbywfe status is 0 1: cluster_b standbywfe status is 1
11	RO	0x0	standbywfil2_cluster_b standbywfil2 status of cluster_b 0: cluster_b standbywfil2 status is 0 1: cluster_b standbywfil2 status is 1
10	RO	0x0	l2flushdone_cluster_b l2flushdone status of cluster_b 0: cluster_b l2flushdone status is 0 1: cluster_b l2flushdone status is 1
9:6	RO	0x0	standbywfi_cluster_l standbywfi status of cluster_l 0: cluster_l standbywfi status is 0 1: cluster_l standbywfi status is 1
5:2	RO	0x0	standbywfe_cluster_l standbywfe status of cluster_l 0: cluster_l standbywfe status is 0 1: cluster_l standbywfe status is 1
1	RO	0x0	standbywfil2_cluster_l standbywfil2 status of cluster_l 0: cluster_l standbywfil2 status is 0 1: cluster_l standbywfil2 status is 1
0	RO	0x0	l2flushdone_cluster_l l2flushdone status of cluster_l 0: cluster_l l2flushdone status is 0 1: cluster_l l2flushdone status is 1

PMU_OSC_CNT
Address: Operational Base + offset (0x0080)
pmu osc count register

Bit	Attr	Reset Value	Description
31:20	RO	0x0	reserved
19:0	RW	0x00000	pmu_osc_cnt pmu osc stable counter value

PMU_PLLLOCK_CNT
Address: Operational Base + offset (0x0084)
pmu pll lock count register

Bit	Attr	Reset Value	Description
31:20	RO	0x0	reserved
19:0	RW	0x00000	pmu_plllock_cnt pmu pll lock counter value

PMU_PLLRST_CNT
Address: Operational Base + offset (0x0088)
pmu pll reset count register

Bit	Attr	Reset Value	Description
31:20	RO	0x0	reserved
19:0	RW	0x00000	pmu_pllrst_cnt pmu pll reset counter value

PMU_STABLE_CNT
Address: Operational Base + offset (0x008c)
pmu power stable count register

Bit	Attr	Reset Value	Description
31:20	RO	0x0	reserved
19:0	RW	0x00000	pmu_stable_cnt pmu PMIC stable counter value

PMU_DDRIO_PWRON_CNT
Address: Operational Base + offset (0x0090)
pmu ddrio power on count register

Bit	Attr	Reset Value	Description
31:20	RO	0x0	reserved
19:0	RW	0x00000	pmu_ddrio_pwron_cnt pmu ddrio power on counter value

PMU_WAKEUP_RST_CLR_CNT
Address: Operational Base + offset (0x0094)
pmu wakeup reset clear count register

Bit	Attr	Reset Value	Description
31:20	RO	0x0	reserved
19:0	RW	0x00000	pmu_wakeup_rst_cnt pmu wakeup reset counter value

PMU_DDR_SREF_ST
Address: Operational Base + offset (0x0098)
pmu ddr self refresh status register

Bit	Attr	Reset Value	Description
31:3	RO	0x0	reserved
2	RW	0x0	ddrc1_sref_done_ext ddr controller 1 self re-fresh done, active high
1	RO	0x0	reserved
0	RW	0x0	ddrc0_sref_done_ext ddr controller 0 self re-fresh done, active high

PMU_SCU_L_PWRDN_CNT
Address: Operational Base + offset (0x009c)

pmu scu_l power down count register

Bit	Attr	Reset Value	Description
31:20	RO	0x0	reserved
19:0	RW	0x05dc0	pmu_scu_l_pwrdn_cnt pmu scu_l power down counter value

PMU_SCU_L_PWRUP_CNT

Address: Operational Base + offset (0x00a0)

pmu scu_l power up count register

Bit	Attr	Reset Value	Description
31:20	RO	0x0	reserved
19:0	RW	0x05dc0	pmu_scu_l_pwrup_cnt pmu scu_l power up counter value

PMU_SCU_B_PWRDN_CNT

Address: Operational Base + offset (0x00a4)

pmu scu_b power down count register

Bit	Attr	Reset Value	Description
31:20	RO	0x0	reserved
19:0	RW	0x05dc0	pmu_scu_b_pwrdn_cnt pmu scu_b power down counter value

PMU_SCU_B_PWRUP_CNT

Address: Operational Base + offset (0x00a8)

pmu scu_b power up count register

Bit	Attr	Reset Value	Description
31:20	RO	0x0	reserved
19:0	RW	0x05dc0	pmu_scu_b_pwrup_cnt pmu scu_b power up counter value

PMU_GPU_PWRDN_CNT

Address: Operational Base + offset (0x00ac)

pmu gpu power down count register

Bit	Attr	Reset Value	Description
31:20	RO	0x0	reserved
19:0	RW	0x05dc0	pmu_gpu_pwrdn_cnt pmu gpu power down counter value

PMU_GPU_PWRUP_CNT

Address: Operational Base + offset (0x00b0)

pmu gpu power up count register

Bit	Attr	Reset Value	Description
31:20	RO	0x0	reserved
19:0	RW	0x05dc0	pmu_gpu_pwrup_cnt pmu gpu power up counter value

PMU_CENTER_PWRDN_CNT
Address: Operational Base + offset (0x00b4)
pmu center power down count register

Bit	Attr	Reset Value	Description
31:20	RO	0x0	reserved
19:0	RW	0x05dc0	pmu_center_pwrdn_cnt pmu center power down counter value

PMU_CENTER_PWRUP_CNT
Address: Operational Base + offset (0x00b8)
pmu center power up count register

Bit	Attr	Reset Value	Description
31:20	RO	0x0	reserved
19:0	RW	0x05dc0	pmu_center_pwrup_cnt pmu center power up counter value

PMU_TIMEOUT_CNT
Address: Operational Base + offset (0x00bc)
pmu timeout count register

Bit	Attr	Reset Value	Description
31:20	RO	0x0	reserved
19:0	RW	0x00000	timeout_count timeout wakeup counter value

PMU_CPU0APM_CON
Address: Operational Base + offset (0x00c0)
pmu cpu0 auto power down control register

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3	RW	0x0	cpu_l0_sft_wakeup cpu l0 software wakeup source. 1: wakeup ; 0: nothing ;
2	RW	0x0	reserved
1	RW	0x0	cpu_l0_int_wakeup_en cpu l0 interrupt wake enable. 0: disable ; 1: enable ;
0	RW	0x0	cpu_l0_wfi_pwrdn_en cpu_l0 wfi power down enable. 0: disable ; 1: enable ;

PMU_CPU1APM_CON
Address: Operational Base + offset (0x00c4)

pmu cpu1 auto power down control register

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3	RW	0x0	cpu_l1_sft_wakeup cpu l1 software wakeup source. 1: wakeup ; 0: nothing ;
2	RW	0x0	reserved
1	RW	0x0	cpu_l1_int_wakeup_en cpu l1 interrupt wake enable. 0: disable ; 1: enable ;
0	RW	0x0	cpu_l1_wfi_pwrdn_en cpu_l1 wfi power down enable. 0: disable ; 1: enable ;

PMU_CPU2APM_CON

Address: Operational Base + offset (0x00c8)

pmu cpu2 auto power down control register

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3	RW	0x0	cpu_l2_sft_wakeup cpu l2 software wakeup source. 1: wakeup ; 0: nothing ;
2	RW	0x0	reserved
1	RW	0x0	cpu_l2_int_wakeup_en cpu l2 interrupt wake enable. 0: disable ; 1: enable ;
0	RW	0x0	cpu_l2_wfi_pwrdn_en cpu_l2 wfi power down enable. 0: disable ; 1: enable ;

PMU_CPU3APM_CON

Address: Operational Base + offset (0x00cc)

pmu cpu3 auto power down control register

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3	RW	0x0	cpu_l3_sft_wakeup cpu l3 software wakeup source. 1: wakeup ; 0: nothing ;
2	RW	0x0	reserved

Bit	Attr	Reset Value	Description
1	RW	0x0	cpu_l3_int_wakeup_en cpu l3 interrupt wake enable. 0: disable ; 1: enable ;
0	RW	0x0	cpu_l3_wfi_pwrdn_en cpu_l3 wfi power down enable. 0: disable ; 1: enable ;

PMU_CPU0BPM_CON

Address: Operational Base + offset (0x00d0)

pmu cluster_b cpu0 auto power down control register

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3	RW	0x0	cpu_b0_sft_wakeup cpu b0 software wakeup source. 1: wakeup ; 0: nothing ;
2	RW	0x0	reserved
1	RW	0x0	cpu_b0_int_wakeup_en cpu b0 interrupt wake enable. 0: disable ; 1: enable ;
0	RW	0x0	cpu_b0_wfi_pwrdn_en cpu_b0 wfi power down enable. 0: disable ; 1: enable ;

PMU_CPU1BPM_CON

Address: Operational Base + offset (0x00d4)

pmu cluster_b cpu0 auto power down control register

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3	RW	0x0	cpu_b0_sft_wakeup cpu b0 software wakeup source. 1: wakeup ; 0: nothing ;
2	RW	0x0	reserved
1	RW	0x0	cpu_b0_int_wakeup_en cpu b0 interrupt wake enable. 0: disable ; 1: enable ;

Bit	Attr	Reset Value	Description
0	RW	0x0	cpu_b0_wfi_pwrdn_en cpu_b0 wfi power down enable. 0: disable ; 1: enable ;

PMU_NOC_AUTO_ENA
Address: Operational Base + offset (0x00d8)

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved
29	RW	0x0	sdioaudio_gating_disable 0: nothing 1: clock gating disable.
28	RW	0x0	sd_gating_disable 0: nothing 1: clock gating disable.
27	RW	0x0	gic_gating_disable 0: nothing ; 1: clock gating disable.
26	RW	0x0	gpu_gating_disable 0: nothing ; 1: clock gating disable.
25	RW	0x0	perilp_gating_disable 0: nothing ; 1: clock gating disable.
24	RW	0x0	perihp_gating_disable 0: nothing ; 1: clock gating disable.
23	RW	0x0	vcodec_gating_disable 0: nothing 1: clock gating disable.
22	RW	0x0	vdu_gating_disable 0: nothing ; 1: clock gating disable.
21	RW	0x0	rga_gating_disable 0: nothing ; 1: clock gating disable.
20	RW	0x0	iep_gating_disable 0: nothing ; 1: clock gating disable.
19	RW	0x0	vopb_gating_disable 0: nothing ; 1: clock gating disable.

Bit	Attr	Reset Value	Description
18	RW	0x0	vopl_gating_disable 0: nothing ; 1: clock gating disable.
17	RW	0x0	isp0_gating_disable 0: nothing ; 1: clock gating disable.
16	RW	0x0	isp1_gating_disable 0: nothing ; 1: clock gating disable.
15	RW	0x0	hdcp_gating_disable 0: nothing ; 1: clock gating disable.
14	RW	0x0	usb3_gating_disable 0: nothing 1: clock gating disable.
13	RW	0x0	perilpm0_gating_disable 0: nothing ; 1: clock gating disable.
12	RW	0x0	center_gating_disable 0: noting 1: clock gating disable.
11	RW	0x0	ccim0_gating_disable 0: nothing ; 1: clock gating disable.
10	RW	0x0	ccim1_gating_disable 0: nothing ; 1: clock gating disable.
9	RW	0x0	vio_gating_disable 0: nothing ; 1: clock gating disable.
8	RW	0x0	msch0_gating_disable 0: nothing ; 1: clock gating disable.
7	RW	0x0	msch1_gating_disable 0: nothing ; 1: clock gating disable.
6	RW	0x0	alive_gating_disable 0: nothing ; 1: clock gating disable.
5	RW	0x0	pmu_gating_disable 0: nothing ; 1: clock gating disable.
4	RW	0x0	edp_gating_disable 0: nothing ; 1: clock gating disable.

Bit	Attr	Reset Value	Description
3	RW	0x0	gmac_gating_disable 0: nothing ; 1: gmac clock gating disable.
2	RW	0x0	emmc_gating_disable 0: nothing ; 1: clock gating disable.
1	RW	0x0	center1_gating_disable 0: nothing 1: clock gating disable.
0	RW	0x0	pmum0_gating_disable 0: noting ; 1: clock gating disable.

PMU_PWRDN_CON1
Address: Operational Base + offset (0x00dc)
pmu power down configure register1

Bit	Attr	Reset Value	Description
31:3	RO	0x0	reserved
2	RW	0x0	vd_center_pwrdwn vd_center power down enable 0: disable 1: enable
1	RW	0x0	vd_scu_b_pwrdwn vd_scu_b power down enable 0: disable 1: enable
0	RW	0x0	vd_scu_l_enable vd_scu_l power down enable 0: disable 1: enable

PMU_SYS_REG0
Address: Operational Base + offset (0x00f0)
pmu system register 0

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	pmu_sys_reg0 system register 0

PMU_SYS_REG1
Address: Operational Base + offset (0x00f4)
pmu system register 1

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	pmu_sys_reg1 system register 1

PMU_SYS_REG2
Address: Operational Base + offset (0x00f8)
pmu system register 2

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	pmu_sys_reg2 system register 2

PMU_SYS_REG3
Address: Operational Base + offset (0x00fc)
pmu system register 3

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	pmu_sys_reg3 system register 3

9.5 Timing Diagram

9.5.1 Each domain power switch timing

The following figure is the each domain power down and power up timing.

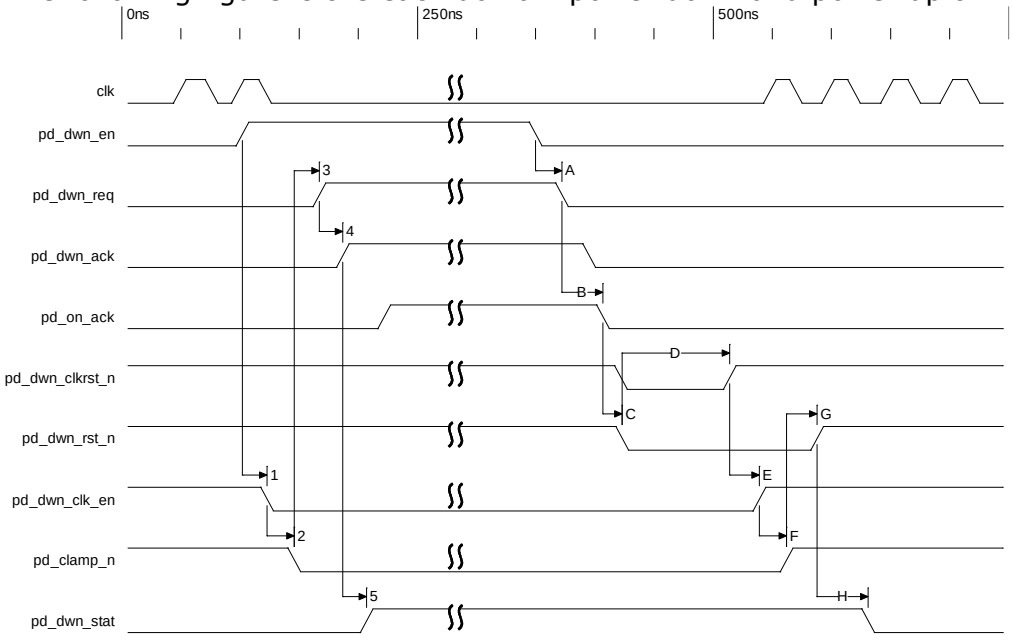


Fig. 4-5 Each Domain Power Switch Timing

9.5.2 External wakeup PAD timing

The PMU supports a lot of external wakeup sources, such as SD/MMDC, USBDEV, SIM detect wakeup, GPIO0 wakeup source and so on. All these external wakeup sources must meet the timing requirement (at least 200us) when the wakeup event is asserted. The following figure gives the timing information.

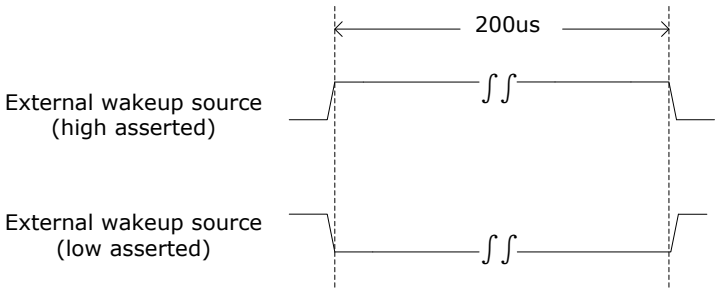


Fig. 4-6 External Wakeup Source PAD Timing

Chapter 10 Memory Management Unit (MMU)

10.1 Overview

An MMU controls address translation, access permissions, memory attribute determination, and checking at a memory system level.

10.2 Block Diagram

The MMU divides memory into 4KB pages, where each page can be individually configured. The MMU uses a 2-level page table structure:
1. The first level, the Page Directory consists of 1024 Directory Table Entries (DTEs), each pointing to a Page Table.
2. The second level, the Page Table consists of 1024 Page Table Entries (PTEs), each pointing to a page in memory.
Fig. 14-1 shows the structure of the two-level page table.

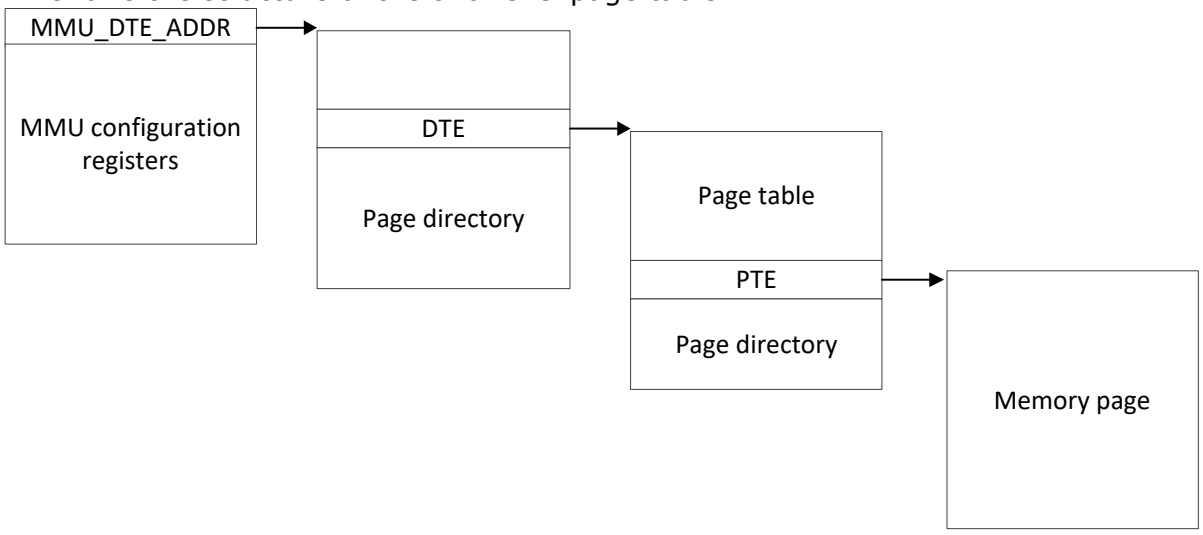


Fig. 10-1 MMU Structure

Fig. 14-2 shows the arrangement of the MMU address bits.

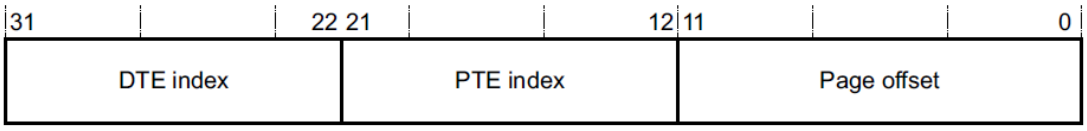


Fig. 10-2 MMU Address Bits

The MMU uses the following algorithm to translate an address:

1. Find the DTE at address given by:
   MMU_DTE_ADDR + (4 * DTE index)
2. Find the PTE at address given by:
   Page table address from DTE + (4 * PTE index)
3. Calculate effective address as follows:
   Page address from PTE + Page offset.

The page directory is a 4KB data structure that contains 1024 32-bit DTEs. The page directory must align at a 4KB boundary in memory.
Each DTE contains the address of a page table and a page table present bit. The system:

- initializes the entire page directory before use
- clear the page table present bit for any DTE that does not point to a valid page table.

The following table shows the page bit assignments.

Table 10-1 Page directory entry detail

Bits	Name	Function
[31:12]	Page table address	This field stores bits [31:12] of the address for a page table
[11:1]	Reserved	Reserved, write as zero
[0]	Page table present	This bit indicates when the page table address points to a valid page table. 0 = page table not valid 1= page table valid.

The page table is a 4KB data structure containing 1024 32-bit PTEs. The page table must be aligned at a 4KB boundary in memory.
Each PTE contains the address of a page of memory, a Page Table present bit, and Read/Write Permission bits. The entire Page Table must be initialized before use, and any PTE not pointing to a valid page must clear the Page Present bit.
The following table shows the page table entry bit assignments.

Table 10-2 Page directory entry detail

Bits	Name	Function
[31:12]	Page table address	This field stores bits [31:12] of the address for a page table
[11:9]	Reserved	Reserved, write as zero
[8]	Read allocate	If set, allocate cache space on read misses. Must not be set if the Read cacheable bit is not set. Only used for reads, if the Override cache attributes bit is set.
[7]	Read cacheable	If set, enable caching or prefetching of data. Only used for reads, if the Override cache attributes bit is set.
[6]	Write bufferable	If set, enable write to be delayed on their way to memory. Only used for writes, if the Override cache attributes bit is set.
[5]	Write allocate	If set, allocate cache space on write misses. Must not be set if the Write cacheable bit is not set. Only used for writes, if the Override cache attributes is set.
[4]	Write cacheable	If set, enable different writes to be merged together. Only used for writes, if the Override cache attributes bit is set.
[3]	Override cache attributes	If set, the cacheability attributes specified in bits [8:4] are used to control the cache attribytes used on the memory bus. If cleared, the default cacheability attributes from the specific processors are used on the system bus.
[2]	Write permission	Enable write accesses to the page, if present.
[1]	Read permission	Enable read accesses from the page, if present.
[0]	Page present	This bit indicates when the page table field points to a valid page. 0 = page not valid 1 = page valid.

Block Descriptions:

- APB Interface

The APB Interface implements the APB slave operation. Its data bus width is 32 bits.

- Register Block

A register block that read coherence for the current count register.

- Interrupt & system reset control

An interrupt/system reset generation block is comprised of a decrementing counter and control logic.

10.3 Register Description

This section describes the control/status registers of the design.

10.3.1 Registers Summary

Name	Offset	Size	Reset Value	Description
MMU_DTE_ADDR	0x0000	W	0x00000000	MMU current page table address
MMU_STATUS	0x0004	W	0x00000018	MMU status register
MMU_CMD	0x0008	W	0x00000000	MMU command register
MMU_PAGE_FAULT_ADDR	0x000c	W	0x00000000	MMU logic address of last page fault register
MMU_ZAP_ONE_LINE	0x0010	W	0x00000000	MMU zap cache line register
MMU_INT_RAWSTAT	0x0014	W	0x00000000	MMU raw interrupt status register
MMU_INT_CLEAR	0x0018	W	0x00000000	MMU interrupt clear register
MMU_INT_MASK	0x001c	W	0x00000000	MMU interrupt mask register
MMU_INT_STATUS	0x0020	W	0x00000000	MMU interrupt status register
MMU_AUTO_GATING	0x0024	W	0x00000000	clock atuo gating register

10.3.2 Detail Register Description

MMU_DTE_ADDR

Address: Operational Base + offset (0x0000)

MMU current page table address

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	mmu_dte_addr page table address

MMU_STATUS

Address: Operational Base + offset (0x0004)

MMU status register

Bit	Attr	Reset Value	Description
31:11	RO	0x0	reserved
10:6	RO	0x00	mmu_page_fault_bus_id Index of master responsible for the last page fault
5	RO	0x0	mmu_page_fault_is_write The direction of access for last page fault: 0: read 1:write
4	RO	0x1	mmu_replay_buffer_empty 1：The MMU replay buffer is empty.

Bit	Attr	Reset Value	Description
3	RO	0x1	mmu_idle the MMu is idle when accesses are being translated and there are no unfinished translated access. The MMU_IDLE signal only reports idle when the MMU processor is idle and accesses are active on the external bus. Note: the MMU can be idle in page fault mode. 1: MMU is idle
2	RO	0x0	mmu_stall_active MMU stall mode currently enabled. The mode is enabled by command. 1: MMU is in stall active status
1	RO	0x0	mmu_page_fault_active MMU page fault mode currently enabled.The mode is enabled by command 1: page fault is active
0	RO	0x0	mmu_paging_enabled 0: paging is disabled 1: Paging is enabled

MMU_CMD
Address: Operational Base + offset (0x0008)
MMU command register

Bit	Attr	Reset Value	Description
31:3	RO	0x0	reserved
2:0	RW	0x0	mmu_cmd 0: MMU_ENABLE_PAGING. enable paging. 1: MMU_DISABLE_PAGING. disable paging. 2: MMU_ENABLE_STALL. turn on stall mode. 3: MMU_DISABLE_STALL. turn off stall mode. 4: MMU_ZAP_CACHE. zap the entire page table cache. 5: MMU_PAGE_FAULT_DONE. leave page fault mode. 6: MMU_FORCE_RESET. reset the mmu. The MMU_ENABLE_STALL command can always be issued. Other commands are ignored unless the MMU is idle or stalled.

MMU_PAGE_FAULT_ADDR
Address: Operational Base + offset (0x000c)
MMU logic address of last page fault register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	mmu_page_fault_addr address of last page fault

MMU_ZAP_ONE_LINE
Address: Operational Base + offset (0x0010)
MMU zap cache line register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	mmu_zap_one_line address to be invalidated from the page table cache.

MMU_INT_RAWSTAT
Address: Operational Base + offset (0x0014)
MMU raw interrupt status register

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1	RO	0x0	read_bus_error read bus error
0	RO	0x0	page_fault page fault

MMU_INT_CLEAR
Address: Operational Base + offset (0x0018)
MMU interrupt clear register

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1	RW	0x0	read_bus_error_clear read bus error interrupt clear. write 1 to this register can clear read bus error interrupt.
0	RW	0x0	page_fault_clear page fault interrupt clear, write 1 to this register can clear page fault interrupt.

MMU_INT_MASK
Address: Operational Base + offset (0x001c)
MMU interrupt mask register

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1	RW	0x0	read_bus_error_int_en read bus error interrupt enable
0	RW	0x0	page_fault_int_en page fault interrupt enable

MMU_INT_STATUS
Address: Operational Base + offset (0x0020)
MMU interrupt status register

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1	RW	0x0	read_bus_error read bus error interrupt

Bit	Attr	Reset Value	Description
0	RO	0x0	page_fault page fault interrupt

MMU_AUTO_GATING
Address: Operational Base + offset (0x0024)
clock atuo gating register

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	mmu_atuo_gating mmu clock auto gating when it is 1, the mmu will auto gating itself

10.4MMU Base Address

The table below shows the MMU base address in different modules.

ISP0_MMU0_BASE	ISP0_BASEADDR + 0x4000
ISP0_MMU1_BASE	ISP0_BASEADDR + 0x5000
ISP1_MMU0_BASE	ISP1_BASEADDR + 0x4000
ISP1_MMU1_BASE	ISP1_BASEADDR + 0x5000
VOPL_MMU_BASE	VOPL_BASEADDR + 0x300
VOPB_MMU_BASE	VOPB_BASEADDR + 0x300
IEP_MMU_BASE	IEP_BASE + 0x800
HDCP_MMU_BASE	0xff930000

Chapter 11 Timer

11.1 Overview

Timer is a programmable timer peripheral. This component is an APB slave device. In RK3399Pro there are 12 Timers(timer0~timer11), 12 Secure Timers(stimer0~stimer11) in ALIVE and 2 Timers(pmutimer0~pmutimer1) in PMU.
All timers count up from a lower programmed value to a higher programmed value and generate an interrupt when the counter reaches the programmed value.
Timer supports the following features:

- Timer0~Timer11 and pmutimer0~pmutimer1 are used for no-secure, stimer0~stimer11 are used for secure.
- Two operation modes: free-running and user-defined count.
- Maskable for each individual interrupt.

11.2 Block Diagram

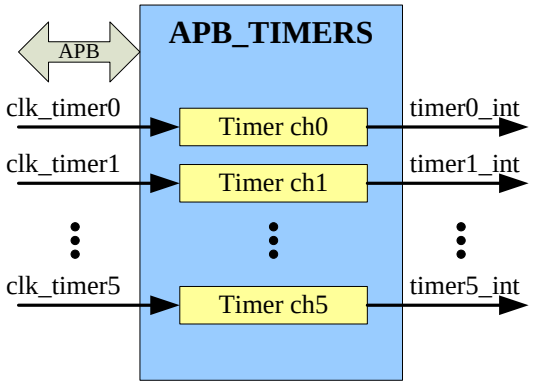


Fig. 11-1 Timer Block Diagram

The above figure shows the architecture of the APB timers (include six programmable timer channels) that in the bus subsystem. The Stimers that in the bus subsystem only include two programmable timer channels.

11.3 Function Description

11.3.1 Timer clock

The timer clock is 24MHz OSC.

11.3.2 Programming sequence

1. Initialize the timer by the TIMERn_CONTROLREG register:
   - Disable the timer by writing a "0" to the timer enable bit (bit 0). Accordingly, the timer_en output signal is de-asserted.
   - Program the timer mode—user-defined or free-running—by writing a "0" or "1" respectively, to the timer mode bit (bit 1).
   - Set the interrupt mask as either masked or not masked by writing a "0" or "1" respectively, to the timer interrupt mask bit (bit 2).
2. Load the timer count value into the TIMERn_LOAD_COUNT0 ~TIMERn_ LOAD_COUNT03 register.
3. Enable the timer by writing a "1" to bit 0 of TIMERn_CONTROLREG.

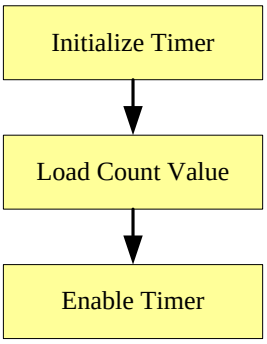


Fig. 11-2 Timer Usage Flow

11.3.3 Loading a timer count value

The initial value for each timer is TIMERn_LOAD_COUNT3 and TIMERn_ LOAD_COUNT2. The count register will count up to the value loaded in the register TIMERn_LOAD_COUNT1 and TIMERn_ LOAD_COUNT0. Two events can cause a timer to load zero:

- Timer is enabled after reset or disabled.
- Timer counts up to the value stored in TIMERn_LOAD_COUNT1 and TIMERn_ LOAD_COUNT0, when timer is configured into free-running mode.

11.3.4 Timer mode selection

- User-defined count mode – Timer loads TIMERn_LOAD_COUNT3 and TIMERn_ LOAD_COUNT2 as initial value. When the timer counts up to the value in TIMERn_LOAD_COUNT1 and TIMERn_LOAD_COUNT0, it will not automatically reload the count register. User need to disable timer firstly and follow the programming sequence to make timer work again.
- Free-running mode – Timer loads the TIMERn_LOAD_COUNT3 and TIMERn_LOAD_COUNT2 register as initial value. Timer will automatically reload the count register, when timer counts up to the value in TIMERn_LOAD_COUNT1 and TIMERn_LOAD_COUNT0.

11.4 Register Description

This section describes the control/status registers of the design. Software should read and write these registers using 32-bits accesses.

11.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
TIMER_n_LOAD_COUNT0	0x0000	W	0x00000000	Timer n higher Load Count Register
TIMER_n_LOAD_COUNT1	0x0004	W	0x00000000	Timer n higher Load Count Register
TIMER_n_CURRENT_VALUE0	0x0008	W	0x00000000	Timer n Current Value Register
TIMER_n_CURRENT_VALUE1	0x000c	W	0x00000000	Timer n Current Value Register
TIMER_n_LOAD_COUNT2	0x0010	W	0x00000000	Timer n lower Load Count Register
TIMER_n_LOAD_COUNT3	0x0014	W	0x00000000	Timer n lower Load Count Register

Name	Offset	Size	Reset Value	Description
TIMER_n_INTSTATUS	0x0018	W	0x00000000	Timer Interrupt Stauts Register
TIMER_n_CONTROLREG	0x001c	W	0x00000000	Timer n Control Register

*Notes:Size:**B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access*

11.4.2 Detail Register Description

TIMER_n_LOAD_COUNT0
Address: Operational Base + offset (0x00)
Timer n High Load Count Register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	load_count_low bits Low 32 bits value to be loaded into Timer n.

TIMER_n_LOAD_COUNT1
Address: Operational Base + offset (0x04)
Timer n High Load Count Register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	load_count_high bits High 32 bits value to be loaded into Timer n.

TIMER_n_CURRENT_VALUE0
Address: Operational Base + offset (0x08)
Timer n Current Value Register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	current_cnt_lowbits Low 32 bits of current value of timer n.

TIMER_n_CURRENT_VALUE1
Address: Operational Base + offset (0x0c)
Timer n Current Value Register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	current_cnt_highbits High 32 bits of current value of timer n.

TIMER_n_LOAD_COUNT2
Address: Operational Base + offset (0x10)
Timer n Low Load Count Register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	load_count_low bits Low 32 bits value to be loaded into Timer n.

TIMER_n_LOAD_COUNT3
Address: Operational Base + offset (0x14)
Timer n Low Load Count Register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	load_count_high bits High 32 bits value to be loaded into Timer n.

TIMER_n_INTSTATUS
Address: Operational Base + offset (0x18)
Timer Interrupt Stauts Register

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	W1C	0x0	int_pd This register contains the interrupt status for timer n. Write 1 to this register will clear the interrupt.

TIMER_n_CONTROLREG
Address: Operational Base + offset (0x1c)
Timer n Control Register

Bit	Attr	Reset Value	Description
31:3	RO	0x0	reserved
2	RW	0x0	int_en Timer interrupt mask 0: mask 1: not mask
1	RW	0x0	timer_mode Timer mode. 0: free-running mode 1: user-defined count mode
0	RW	0x0	timer_en Timer enable. 0: disable 1: enable

11.5 Application Notes

In the chip, the timer_clk is from 24MHz OSC, asynchronous to the pclk. When user disables the timer enables bit, the timer en output signal is de-asserted, and timer_clk will stop. When user enables the timer, the timer_en signal is asserted and timer_clk will start running.
The application is only allowed to re-config registers when timer_en is low.

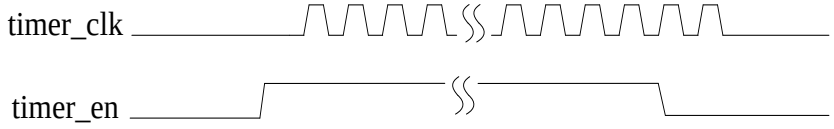


Fig. 11-3 Timing between timer_en and timer_clk

Please refer to function description section for the timer usage flow.

Chapter 12 Generic Interrupt Controller (GIC)

12.1 Overview

The GIC-500 in RK3399Pro provides registers for managing interrupt sources, interrupt behavior, and interrupt routing to one or more cores.
The configuration of GIC500 is shown below:

Configuration item	Value
num_clusters	2
gic_num_rid_bits	4
num_spis	256
disable_security	false
gic_num_wid_bits	4
lpi_support	true
cpus_per_clsuter_0	4
are_option	false
lpi_size	256
cpus_per_cluster_1	2
did_size	16

Table 11-1 GIC500 configuration

The GIC in RK3399Pro supports following feature:

- Support 2 clusters
- Support cluster 0 with 4 cpus
- Support cluster 1 with 2 cpus
- The following interrupt types:
  - Locality-specific Peripheral Interrupts (LPIs). These interrupts are generated by a peripheral writing to a memory-mapped register in the GIC-500. See Configurable options for the GIC-500 RTL on page 1-9.
  - 256 Shared Peripheral Interrupts (SPIs).
  - 16 Private Peripheral Interrupts (PPIs), that are independent for each core and can be programmed to support either edge-triggered or level-sensitive interrupts.
  - 16 SGIs, that are generated either by using software to write to GICD_SGIR or through the GIC CPU interface of a core.
- Interrupt Translation Service (ITS). This provides device isolation and ID translation for message-based interrupts, which allows virtual machines to program devices directly.
- Memory-mapped access to all registers.
- Interrupt masking and prioritization.
- Programmable interrupt routing that is based on affinity.
- Three different interrupt groups, which allow interrupts to target different Exception levels:
  - Group 0.
  - Non-secure Group 1.
  - Secure Group 1.
- A global Disable Security (DS) bit. This allows support for systems with and without security.
- 32 priority values, five bits for each interrupt.

12.2 Block Diagram

The GIC500 in the RK3399Pro is connected with CPU clusters through AXI Stream bus, as shown below:

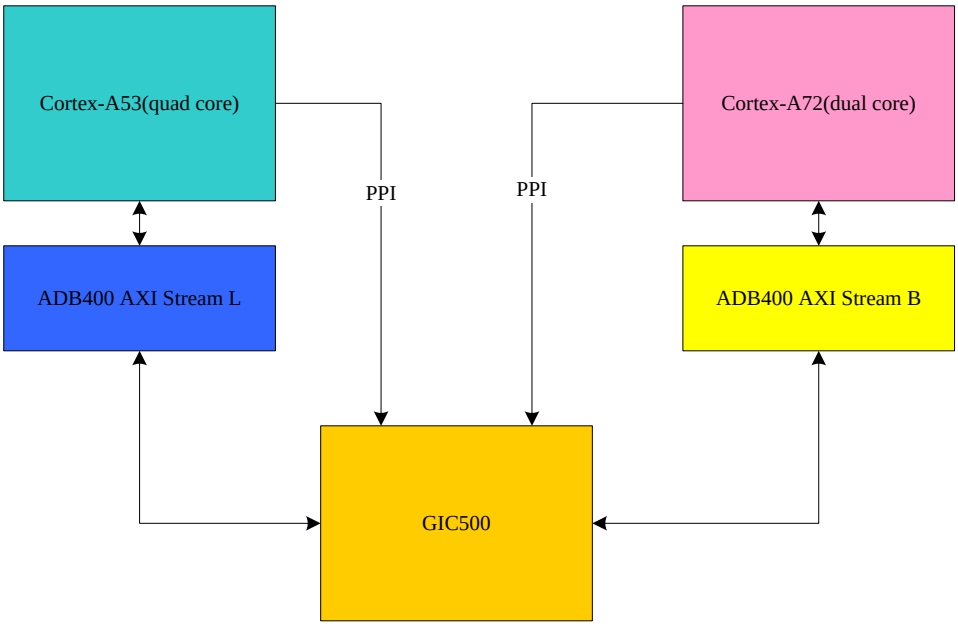


Fig. 12-1 Block Diagram

Chapter 13 DMA Controller (DMAC)

13.1 Overview

This device supports 2 Direct Memory Access (DMA) Controllers, DMAC0 and DMAC1. Both of these two DMAC support transfers between memory and memory, peripheral and memory. DMAC is under Non-secure state after reset, and the secure state can be changed by configuring SGRF module.

DMAC0 supports the following features:

- Supports Trustzone technology
- Supports 12 peripheral request
- Up to 64bits data size
- 6 channel at the same time
- Up to burst 16
- 12 interrupts output and 1 abort output
- Supports 32 MFIFO depth

Following table shows the DMAC0 request mapping scheme.

Table 13-1 DMAC0 Request Mapping Table

Req number	Source	Polarity
0	I2S0 tx	High level
1	I2S0 rx	High level
2	I2S1 tx	High level
3	I2S1 rx	High level
4	I2S2 tx	High level
5	I2S2 rx	High level
6	PWM	High level
7	SPDIF tx	High level
8	SPI5 tx	High level
9	SPI5 rx	High level
10	Reserved	
11	Reserved	

DMAC1 supports the following features:

- Supports Trustzone technology
- Supports 20 peripheral request
- Up to 64bits data size
- 8 channel at the same time
- Up to burst 16
- 16 interrupts output and 1 abort output
- Supports 128 MFIFO depth

Following table shows the DMAC1 request mapping scheme.

Table 13-2 DMAC1 Request Mapping Table

Req number	Source	Polarity
0	UART0 tx	High level
1	UART0 rx	High level
2	UART1 tx	High level
3	UART1 rx	High level
4	UART2 tx	High level
5	UART2 rx	High level
6	UART3 tx	High level
7	UART3 rx	High level
8	UART4 tx	High level
9	UART4 rx	High level
10	SPI0 tx	High level
11	SPI0 rx	High level
12	SPI1 tx	High level
13	SPI1 rx	High level

Req number	Source	Polarity
14	SPI2 tx	High level
15	SPI2 rx	High level
16	SPI3 tx	High level
17	SPI3 rx	High level
18	SPI4 tx	High level
19	SPI4 rx	High level

DMAC support incrementing-address burst and fixed-address burst. But in the case of access SPI and UART at byte or halfword size, DMAC only support fixed-address burst and the address must be aligned to word.

13.2 Block Diagram

Following figure shows the block diagram of DMAC.

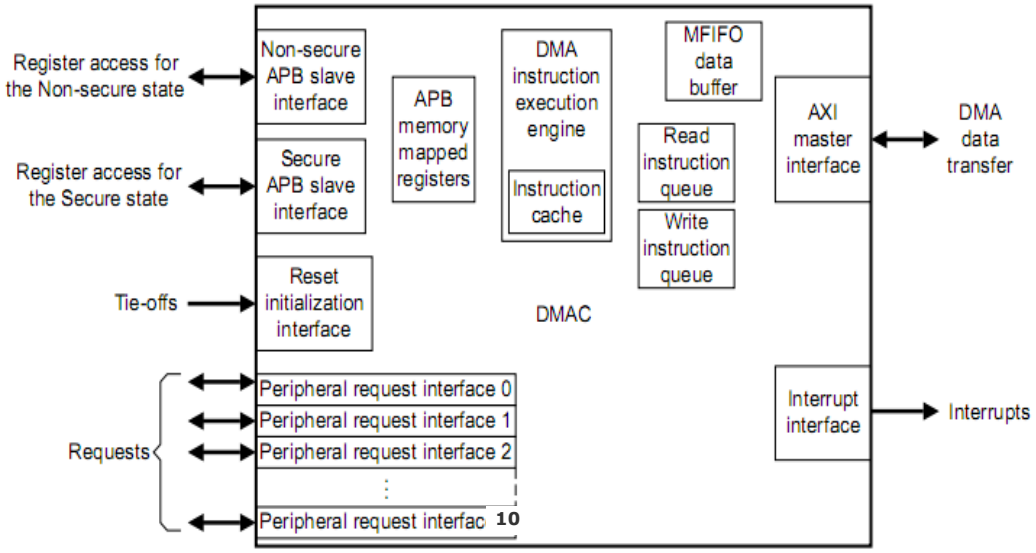


Fig. 13-1 Block diagram of DMAC

As the DMAC supports Trustzone technology, so dual APB interfaces enable the operation of the DMAC to be partitioned into the secure state and Non-secure state. You can use the APB interfaces to access status registers and also directly execute instructions in the DMAC. The default interface after reset is Non-secure apb interface.

13.3 Function Description

13.3.1 Introduction

The DMAC contains an instruction processing block that enables it to process program code that controls a DMA transfer. The program code is stored in a region of system memory that the DMAC accesses using its AXI interface. The DMAC stores instructions temporarily in a cache.

DMAC0 supports 6 channels and DMAC1 supports 8 channels, each channel capable of supporting a single concurrent thread of DMA operation. In addition, a single DMA manager thread exists, and you can use it to initialize the DMA channel threads. The DMAC executes up to one instruction for each AXI clock cycle. To ensure that it regularly executes each active thread, it alternates by processing the DMA manager thread and then a DMA channel thread. It uses a round-robin process when selecting the next active DMA channel thread to execute.

The DMAC uses variable-length instructions that consist of one to six bytes. It provides a separate Program Counter (PC) register for each DMA channel. When a thread requests an instruction from an address, the cache performs a look-up. If a cache hit occurs, then the cache immediately provides the data. Otherwise, the thread is stalled while the DMAC uses the AXI interface to perform a cache line fill. If an instruction is greater than 4 bytes, or spans the end of a cache line, the DMAC performs multiple cache accesses to fetch the instruction.

When a cache line fill is in progress, the DMAC enables other threads to access the cache, but if another cache miss occurs, this stalls the pipeline until the first line fill is complete. When a DMA channel thread executes a load or store instruction, the DMAC adds the instruction to the relevant read or write queue. The DMAC uses these queues as an instruction storage buffer prior to it issuing the instructions on the AXI bus. The DMAC also contains a Multi First-In-First-Out (MFIFO) data buffer that it uses to store data that it reads, or writes, during a DMA transfer.

13.3.2 Operating states

Following figure shows the operating states for the DMA manager thread and DMA channel threads.

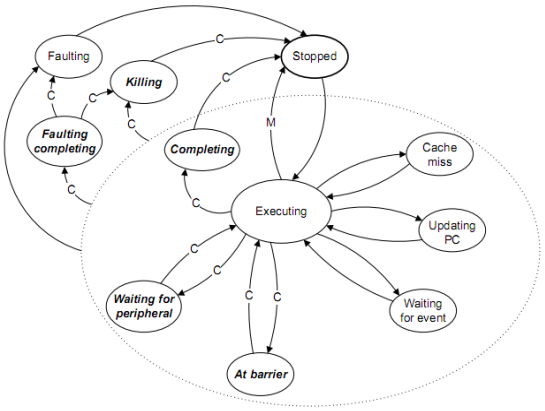


Fig. 13-2 DMAC operation states

Notes:arcs with no letter designator indicate state transitions for the DMA manager and DMA channel threads, otherwise use is restricted as follows:
C DMA channel threads only.
M DMA manager thread only.

After the DMAC exits from reset, it sets all DMA channel threads to the stopped state, and the status of boot_from_pc(tie-off interface of dmac) controls the DMA manager thread state:
boot_from_pc is LOW :DMA manager thread moves to the Stopped state.
boot_from_pc is HIGH :DMA manager thread moves to the Executing state.

13.4 Register Description

13.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
DMAC_DSR	0x0000	W	0x00000000	DMA Manager Status Register
DMAC_DPC	0x0004	W	0x00000000	DMA Program Counter Register
DMAC_INTEN	0x0020	W	0x00000000	Interrupt Enable Register
DMAC_EVENT_RIS	0x0024	W	0x00000000	Event-Interrupt Raw Status Register
DMAC_INTMIS	0x0028	W	0x00000000	Interrupt Status Register
DMAC_INTCLR	0x002c	W	0x00000000	Interrupt Clear Register
DMAC_FSRD	0x0030	W	0x00000000	Fault Status DMA Manager Register
DMAC_FSRC	0x0034	W	0x00000000	Fault Status DMA Channel Register
DMAC_FTRD	0x0038	W	0x00000000	Fault Type DMA Manager Register
DMAC_FTR0	0x0040	W	0x00000000	Fault Type DMA Channel Register
DMAC_FTR1	0x0044	W	0x00000000	Fault Type DMA Channel Register

Name	Offset	Size	Reset Value	Description
DMAC_FTR2	0x0048	W	0x00000000	Fault Type DMA Channel Register
DMAC_FTR3	0x004c	W	0x00000000	Fault Type DMA Channel Register
DMAC_FTR4	0x0050	W	0x00000000	Fault Type DMA Channel Register
DMAC_FTR5	0x0054	W	0x00000000	Fault Type DMA Channel Register
DMAC_FTR6	0x0058	W	0x00000000	Fault Type DMA Channel Register
DMAC_FTR7	0x005c	W	0x00000000	Fault Type DMA Channel Register
DMAC_CSR0	0x0100	W	0x00000000	Channel Status Registers
DMAC_CPC0	0x0104	W	0x00000000	Channel Program Counter Registers
DMAC_CSR1	0x0108	W	0x00000000	Channel Status Registers
DMAC_CPC1	0x010c	W	0x00000000	Channel Program Counter Registers
DMAC_CSR2	0x0110	W	0x00000000	Channel Status Registers
DMAC_CPC2	0x0114	W	0x00000000	Channel Program Counter Registers
DMAC_CSR3	0x0118	W	0x00000000	Channel Status Registers
DMAC_CPC3	0x011c	W	0x00000000	Channel Program Counter Registers
DMAC_CSR4	0x0120	W	0x00000000	Channel Status Registers
DMAC_CPC4	0x0124	W	0x00000000	Channel Program Counter Registers
DMAC_CSR5	0x0128	W	0x00000000	Channel Status Registers
DMAC_CPC5	0x012c	W	0x00000000	Channel Program Counter Registers
DMAC_CSR6	0x0130	W	0x00000000	Channel Status Registers
DMAC_CPC6	0x0134	W	0x00000000	Channel Program Counter Registers
DMAC_CSR7	0x0138	W	0x00000000	Channel Status Registers
DMAC_CPC7	0x013c	W	0x00000000	Channel Program Counter Registers
DMAC_SAR0	0x0400	W	0x00000000	Source Address Registers
DMAC_DAR0	0x0404	W	0x00000000	Destination Address Registers
DMAC_CCR0	0x0408	W	0x00000000	Channel Control Registers
DMAC_LC0_0	0x040c	W	0x00000000	Loop Counter 0 Registers
DMAC_LC1_0	0x0410	W	0x00000000	Loop Counter 1 Registers
DMAC_SAR1	0x0420	W	0x00000000	Source Address Registers
DMAC_DAR1	0x0424	W	0x00000000	Destination Address Registers
DMAC_CCR1	0x0428	W	0x00000000	Channel Control Registers
DMAC_LC0_1	0x042c	W	0x00000000	Loop Counter 0 Registers
DMAC_LC1_1	0x0430	W	0x00000000	Loop Counter 1 Registers
DMAC_SAR2	0x0440	W	0x00000000	Source Address Registers
DMAC_DAR2	0x0444	W	0x00000000	Destination Address Registers
DMAC_CCR2	0x0448	W	0x00000000	Channel Control Registers

Name	Offset	Size	Reset Value	Description
DMAC_LC0_2	0x044c	W	0x00000000	Loop Counter 0 Registers
DMAC_LC1_2	0x0450	W	0x00000000	Loop Counter 1 Registers
DMAC_SAR3	0x0460	W	0x00000000	Source Address Registers
DMAC_DAR3	0x0464	W	0x00000000	Destination Address Registers
DMAC_CCR3	0x0468	W	0x00000000	Channel Control Registers
DMAC_LC0_3	0x046c	W	0x00000000	Loop Counter 0 Registers
DMAC_LC1_3	0x0470	W	0x00000000	Loop Counter 1 Registers
DMAC_SAR4	0x0480	W	0x00000000	Source Address Registers
DMAC_DAR4	0x0484	W	0x00000000	Destination Address Registers
DMAC_CCR4	0x0488	W	0x00000000	Channel Control Registers
DMAC_LC0_4	0x048c	W	0x00000000	Loop Counter 0 Registers
DMAC_LC1_4	0x0490	W	0x00000000	Loop Counter 1 Registers
DMAC_SAR5	0x04a0	W	0x00000000	Source Address Registers
DMAC_DAR5	0x04a4	W	0x00000000	Destination Address Registers
DMAC_CCR5	0x04a8	W	0x00000000	Channel Control Registers
DMAC_LC0_5	0x04ac	W	0x00000000	Loop Counter 0 Registers
DMAC_LC1_5	0x04b0	W	0x00000000	Loop Counter 1 Registers
DMAC_SAR6	0x04c0	W	0x00000000	Source Address Registers
DMAC_DAR6	0x04c4	W	0x00000000	Destination Address Registers
DMAC_CCR6	0x04c8	W	0x00000000	Channel Control Registers
DMAC_LC0_6	0x04cc	W	0x00000000	Loop Counter 0 Registers
DMAC_LC1_6	0x04d0	W	0x00000000	Loop Counter 1 Registers
DMAC_SAR7	0x04e0	W	0x00000000	Source Address Registers
DMAC_DAR7	0x04e4	W	0x00000000	Destination Address Registers
DMAC_CCR7	0x04e8	W	0x00000000	Channel Control Registers
DMAC_LC0_7	0x04ec	W	0x00000000	Loop Counter 0 Registers
DMAC_LC1_7	0x04f0	W	0x00000000	Loop Counter 1 Registers
DMAC_DBGSTATUS	0x0d00	W	0x00000000	Debug Status Register
DMAC_DBGCMD	0x0d04	W	0x00000000	Debug Command Register
DMAC_DBGINST0	0x0d08	W	0x00000000	Debug Instruction-0 Register
DMAC_DBGINST1	0x0d0c	W	0x00000000	Debug Instruction-1 Register
DMAC_CR0	0x0e00	W	0x00047051	Configuration Register 0
DMAC_CR1	0x0e04	W	0x00000057	Configuration Register 1
DMAC_CR2	0x0e08	W	0x00000000	Configuration Register 2
DMAC_CR3	0x0e0c	W	0x00000000	Configuration Register 3
DMAC_CR4	0x0e10	W	0x00000006	Configuration Register 4
DMAC_CRDn	0x0e14	W	0x02094733	DMA Configuration Register
DMAC_WD	0x0e80	W	0x00000000	DMA Watchdog Register

*Notes:Size:**B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access.*
For DMAC0 channel register, only the channel 0~5 is valid.

13.4.2 Detail Register Description

DMAC_DSR

Address: Operational Base + offset (0x0000)

DMA Manager Status Register

Bit	Attr	Reset Value	Description
31:10	RO	0x0	reserved
9	RO	0x0	Provides the security status of the DMA manager thread: 0 = DMA manager operates in the Secure state 1 = DMA manager operates in the Non-secure state.
8:4	RO	0x00	When the DMA manager thread executes a DMAWFE instruction, it waits for the following event to occur: b00000 = event[0] b00001 = event[1] b00010 = event[2] … b11111 = event[31].
3:0	RO	0x0	The operating state of the DMA manager: b0000 = Stopped b0001 = Executing b0010 = Cache miss b0011 = Updating PC b0100 = Waiting for event b0101-b1110 = reserved b1111 = Faulting.

DMAC_DPC
Address: Operational Base + offset (0x0004)
DMA Program Counter Register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	Program counter for the DMA manager thread

DMAC_INTEN
Address: Operational Base + offset (0x0020)
Interrupt Enable Register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	Program the appropriate bit to control how the DMAC responds when it executes DMASEV: Bit [N] = 0 If the DMAC executes DMASEV for the event-interrupt resource N then the DMAC signals event N to all of the threads. Set bit [N] to 0 if your system design does not use irq[N] to signal an interrupt request. Bit [N] = 1 If the DMAC executes DMASEV for the event-interrupt resource N then the DMAC sets irq[N] HIGH. Set bit [N] to 1 if your system designer requires irq[N] to signal an interruptrequest.

DMAC_EVENT_RIS
Address: Operational Base + offset (0x0024)

Event-Interrupt Raw Status Register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	Returns the status of the event-interrupt resources: Bit [N] = 0 Event N is inactive or irq[N] is LOW. Bit [N] = 1 Event N is active or irq[N] is HIGH.

DMAC_INTMIS

Address: Operational Base + offset (0x0028)

Interrupt Status Register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	Provides the status of the interrupts that are active in the DMAC: Bit [N] = 0 Interrupt N is inactive and therefore irq[N] is LOW. Bit [N] = 1 Interrupt N is active and therefore irq[N] is HIGH

DMAC_INTCLR

Address: Operational Base + offset (0x002c)

Interrupt Clear Register

Bit	Attr	Reset Value	Description
31:0	WO	0x00000000	Controls the clearing of the irq outputs: Bit [N] = 0 The status of irq[N] does not change. Bit [N] = 1 The DMAC sets irq[N] LOW if the INTEN Register programs the DMAC to signal an interrupt. Otherwise, the status of irq[N] does not change.

DMAC_FSRD

Address: Operational Base + offset (0x0030)

Fault Status DMA Manager Register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	Provides the fault status of the DMA manager. Read as: 0 = the DMA manager thread is not in the Faulting state 1 = the DMA manager thread is in the Faulting state.

DMAC_FSRC

Address: Operational Base + offset (0x0034)

Fault Status DMA Channel Register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	Each bit provides the fault status of the corresponding channel. Read as: Bit [N] = 0 No fault is present on DMA channel N. Bit [N] = 1 DMA channel N is in the Faulting or Faulting completing state.

DMAC_FTRD

Address: Operational Base + offset (0x0038)

Fault Type DMA Manager Register

Bit	Attr	Reset Value	Description
31	RO	0x0	reserved
30	RO	0x0	If the DMA manager aborts, this bit indicates if the erroneous instruction was read from the system memory or from the debug interface: 0 = instruction that generated an abort was read from system memory 1 = instruction that generated an abort was read from the debug interface.
29:17	RO	0x0	reserved
16	RO	0x0	Indicates the AXI response that the DMAC receives on the RRESP bus, after the DMA manager performs an instruction fetch: 0 = OKAY response 1 = EXOKAY, SLVERR, or DECERR response
15:6	RO	0x0	reserved
5	RO	0x0	Indicates if the DMA manager was attempting to execute DMAWFE or DMASEV with inappropriate security permissions: 0 = DMA manager has appropriate security to execute DMAWFE or DMASEV 1 = a DMA manager thread in the Non-secure state attempted to execute either: DMAWFE to wait for a secure event DMASEV to create a secure event or secure interrupt
4	RO	0x0	Indicates if the DMA manager was attempting to execute DMAGO with inappropriate security permissions: 0 = DMA manager has appropriate security to execute DMAGO 1 = DMA manager thread in the Non-secure state attempted to execute DMAGO to create a DMA channel operating in the Secure state.
3:2	RO	0x0	reserved
1	RO	0x0	Indicates if the DMA manager was attempting to execute an instruction operand that was not valid for the configuration of the DMAC: 0 = valid operand 1 = invalid operand.
0	RW	0x0	Indicates if the DMA manager was attempting to execute an undefined instruction: 0 = defined instruction 1 = undefined instruction.

DMAC_FTR0~DMAC_FTR7

Address: Operational Base + offset (0x0040)
Operational Base+0x44
Operational Base+0x48
Operational Base+0x4C
Operational Base+0x50

Operational Base+0x54
Operational Base+0x58
Operational Base+0x5C
Fault Type DMA Channel Register

Bit	Attr	Reset Value	Description
31	RO	0x0	Indicates if the DMA channel has locked-up because of resource starvation: 0 = DMA channel has adequate resources 1 = DMA channel has locked-up because of insufficient resources. This fault is an imprecise abort
30	RO	0x0	If the DMA channel aborts, this bit indicates if the erroneous instruction was read from the system memory or from the debug interface: 0 = instruction that generated an abort was read from system memory 1 = instruction that generated an abort was read from the debug interface. This fault is an imprecise abort but the bit is only valid when a precise abort occurs.
29:19	RO	0x0	reserved
18	RO	0x0	Indicates the AXI response that the DMAC receives on the RRESP bus, after the DMA channel thread performs a data read: 0 = OKAY response 1 = EXOKAY, SLVERR, or DECERR response. This fault is an imprecise abort
17	RO	0x0	Indicates the AXI response that the DMAC receives on the BRESP bus, after the DMA channel thread performs a data write: 0 = OKAY response 1 = EXOKAY, SLVERR, or DECERR response. This fault is an imprecise abort.
16	RO	0x0	Indicates the AXI response that the DMAC receives on the RRESP bus, after the DMA channel thread performs an instruction fetch: 0 = OKAY response 1 = EXOKAY, SLVERR, or DECERR response. This fault is a precise abort.
15:14	RO	0x0	reserved
13	RO	0x0	Indicates if the MFIFO did not contain the data to enable the DMAC to perform the DMAST: 0 = MFIFO contains all the data to enable the DMAST to complete 1 = previous DMALDs have not put enough data in the MFIFO to enable the DMAST to complete. This fault is a precise abort.

Bit	Attr	Reset Value	Description
12	RO	0x0	Indicates if the MFIFO prevented the DMA channel thread from executing DMALD or DMAST. Depending on the instruction: DMALD 0 = MFIFO contains sufficient space 1 = MFIFO is too small to hold the data that DMALD requires. DMAST 0 = MFIFO contains sufficient data 1 = MFIFO is too small to store the data to enable DMAST to complete. This fault is an imprecise abort
11:8	RO	0x0	reserved
7	RO	0x0	Indicates if a DMA channel thread, in the Non-secure state, attempts to program the CCRn Register to perform a secure read or secure write: 0 = a DMA channel thread in the Non-secure state is not violating the security permissions 1 = a DMA channel thread in the Non-secure state attempted to perform a secure read or secure write. This fault is a precise abort
6	RO	0x0	Indicates if a DMA channel thread, in the Non-secure state, attempts to execute DMAWFP, DMALDP, DMASTP, or DMAFLUSHP with inappropriate security permissions: 0 = a DMA channel thread in the Non-secure state is not violating the security permissions 1 = a DMA channel thread in the Non-secure state attempted to execute either: o DMAWFP to wait for a secure peripheral o DMALDP or DMASTP to notify a secure peripheral o DMAFLUSHP to flush a secure peripheral. This fault is a precise abort.
5	RO	0x0	Indicates if the DMA channel thread attempts to execute DMAWFE or DMASEV with inappropriate security permissions: 0 = a DMA channel thread in the Non-secure state is not violating the security permissions 1 = a DMA channel thread in the Non-secure state attempted to execute either: DMAWFE to wait for a secure event DMASEV to create a secure event or secure interrupt. This fault is a precise abort.
4:2	RO	0x0	reserved
1	RO	0x0	Indicates if the DMA channel thread was attempting to execute an instruction operand that was not valid for the configuration of the DMAC: 0 = valid operand 1 = invalid operand. This fault is a precise abort.

Bit	Attr	Reset Value	Description
0	RO	0x0	Indicates if the DMA channel thread was attempting to execute an undefined instruction: 0 = defined instruction 1 = undefined instruction. This fault is a precise abort

DMAC_CSR0~DMAC_CSR7

Address:Operational Base+0x100
Operational Base+0x108
Operational Base+0x110
Operational Base+0x118
Operational Base+0x120
Operational Base+0x128
Operational Base+0x130
Operational Base+0x138

Channel Status Registers

Bit	Attr	Reset Value	Description
31:22	RO	0x0	reserved
21	RO	0x0	The channel non-secure bit provides the security of the DMA channel: 0 = DMA channel operates in the Secure state 1 = DMA channel operates in the Non-secure state
20:16	RO	0x0	reserved
15	RO	0x0	When the DMA channel thread executes DMAWFP this bit indicates if the periph operand was set: 0 = DMAWFP executed with the periph operand not set 1 = DMAWFP executed with the periph operand set
14	RO	0x0	When the DMA channel thread executes DMAWFP this bit indicates if the burst or single operand were set: 0 = DMAWFP executed with the single operand set 1 = DMAWFP executed with the burst operand set.
13:9	RO	0x0	reserved
8:4	RO	0x00	If the DMA channel is in the Waiting for event state or the Waiting for peripheral state then these bits indicate the event or peripheral number that the channel is waiting for: b00000 = DMA channel is waiting for event, or peripheral, 0 b00001 = DMA channel is waiting for event, or peripheral, 1 b00010 = DMA channel is waiting for event, or peripheral, 2 … b11111 = DMA channel is waiting for event, or peripheral, 31

Bit	Attr	Reset Value	Description
3:0	RO	0x0	The channel status encoding is: b0000 = Stopped b0001 = Executing b0010 = Cache miss b0011 = Updating PC b0100 = Waiting for event b0101 = At barrier b0110 = reserved b0111 = Waiting for peripheral b1000 = Killing b1001 = Completing b1010-b1101 = reserved b1110 = Faulting completing b1111 = Faulting

DMAC_CPC0~DMAC_CPC7

Address:Operational Base+0x104
Operational Base+0x10C
Operational Base+0x114
Operational Base+0x11c
Operational Base+0x124
Operational Base+0x12C
Operational Base+0x134
Operational Base+0x13C

Channel Program Counter Registers

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	Program counter for the DMA channel 0 thread

DMAC_SAR0~DMAC_SAR7

Address:Operational Base+0x400
Operational Base+0x420
Operational Base+0x440
Operational Base+0x460
Operational Base+0x480
Operational Base+0x4A0
Operational Base+0x4C0
Operational Base+0x4E0

Source Address Registers

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	Address of the source data for DMA channel 0

DMAC_DAR0~DMAC_DAR7

Address:Operational Base+0x404
Operational Base+0x424
Operational Base+0x444
Operational Base+0x464
Operational Base+0x484
Operational Base+0x4A4
Operational Base+0x4C4
Operational Base+0x4E4

DestinationAddress Registers

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	Address of the Destination data for DMA channel 0

DMAC_CCR0~DMAC_CCR7

Address: Operational Base+0x408
Operational Base+0x428
Operational Base+0x448
Operational Base+0x468
Operational Base+0x488
Operational Base+0x4A8
Operational Base+0x4C8
Operational Base+0x4E8

Channel Control Registers

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27:25	RO	0x0	Programs the state of AWCACHE[3,1:0]a when the DMAC writes the destination data. Bit [27] 0 = AWCACHE[3] is LOW 1 = AWCACHE[3] is HIGH. Bit [26] 0 = AWCACHE[1] is LOW 1 = AWCACHE[1] is HIGH. Bit [25] 0 = AWCACHE[0] is LOW 1 = AWCACHE[0] is HIGH
24:22	RO	0x0	Programs the state of AWPROT[2:0]a when the DMAC writes the destination data. Bit [24] 0 = AWPROT[2] is LOW 1 = AWPROT[2] is HIGH. Bit [23] 0 = AWPROT[1] is LOW 1 = AWPROT[1] is HIGH. Bit [22] 0 = AWPROT[0] is LOW 1 = AWPROT[0] is HIGH
21:18	RO	0x0	For each burst, these bits program the number of data transfers that the DMAC performs when it writes the destination data: b0000 = 1 data transfer b0001 = 2 data transfers b0010 = 3 data transfers … b1111 = 16 data transfers. The total number of bytes that the DMAC writes out of the MFIFO when it executes a DMAST instruction is the product of dst_burst_len and dst_burst_size

Bit	Attr	Reset Value	Description
17:15	RO	0x0	For each beat within a burst, it programs the number of bytes that the DMAC writes to the destination: b000 = writes 1 byte per beat b001 = writes 2 bytes per beat b010 = writes 4 bytes per beat b011 = writes 8 bytes per beat b100 = writes 16 bytes per beat b101-b111 = reserved. The total number of bytes that the DMAC writes out of the MFIFO when it executes a DMAST instruction is the product of dst_burst_len and dst_burst_size.
14	RO	0x0	Programs the burst type that the DMAC performs when it writes the destination data: 0 = Fixed-address burst. The DMAC signals AWBURST[0] LOW. 1 = Incrementing-address burst. The DMAC signals AWBURST[0] HIGH.
13:11	RO	0x0	Set the bits to control the state of ARCACHE[2:0]a when the DMAC reads the source data. Bit [13] 0 = ARCACHE[2] is LOW 1 = ARCACHE[2] is HIGH. Bit [12] 0 = ARCACHE[1] is LOW 1 = ARCACHE[1] is HIGH. Bit [11] 0 = ARCACHE[0] is LOW 1 = ARCACHE[0] is HIGH.
10:8	RO	0x0	Programs the state of ARPROT[2:0]a when the DMAC reads the source data. Bit [10] 0 = ARPROT[2] is LOW 1 = ARPROT[2] is HIGH. Bit [9] 0 = ARPROT[1] is LOW 1 = ARPROT[1] is HIGH. Bit [8] 0 = ARPROT[0] is LOW 1 = ARPROT[0] is HIGH.
7:4	RO	0x0	For each burst, these bits program the number of data transfers that the DMAC performs when it reads the source data: b0000 = 1 data transfer b0001 = 2 data transfers b0010 = 3 data transfers … b1111 = 16 data transfers. The total number of bytes that the DMAC reads into the MFIFO when it executes a DMALD instruction is the product of src_burst_len and src_burst_size

Bit	Attr	Reset Value	Description
3:1	RO	0x0	For each beat within a burst, it programs the number of bytes that the DMAC reads from the source: b000 = reads 1 byte per beat b001 = reads 2 bytes per beat b010 = reads 4 bytes per beat b011 = reads 8 bytes per beat b100 = reads 16 bytes per beat b101-b111 = reserved. The total number of bytes that the DMAC reads into the MFIFO when it executes a DMALD instruction is the product of src_burst_len and src_burst_size
0	RO	0x0	Programs the burst type that the DMAC performs when it reads the source data: 0 = Fixed-address burst. The DMAC signals ARBURST[0] LOW. 1 = Incrementing-address burst. The DMAC signals ARBURST[0] HIGH

DMAC_LC0_0~DMAC_LC0_7

Address: Operational Base+0x40c
Operational Base+0x42C
Operational Base+0x44C
Operational Base+0x46C
Operational Base+0x48C
Operational Base+0x4AC
Operational Base+0x4CC
Operational Base+0x4EC

Loop Counter 0 Registers

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RO	0x00	Loop counter 0 iterations

DMAC_LC1_0~DMAC_LC1_7

Address: Operational Base+0x410
Operational Base+0x430
Operational Base+0x450
Operational Base+0x470
Operational Base+0x490
Operational Base+0x4B0
Operational Base+0x4D0
Operational Base+0x4F0

Loop Counter 1 Registers

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RO	0x00	Loop counter 1 iterations

DMAC_DBGSTATUS

Address: Operational Base + offset (0x0d00)

Debug Status Register

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved

Bit	Attr	Reset Value	Description
1:0	RO	0x0	The debug encoding is as follows: b00 = execute the instruction that the DBGINST [1:0] Registers contain b01 = reserved b10 = reserved b11 = reserved.

DMAC_DBGCMD
Address: Operational Base + offset (0x0d04)
Debug Command Register

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1:0	WO	0x0	The debug encoding is as follows: b00 = execute the instruction that the DBGINST [1:0] Registers contain b01 = reserved b10 = reserved b11 = reserved

DMAC_DBGINST0
Address: Operational Base + offset (0x0d08)
Debug Instruction-0 Register

Bit	Attr	Reset Value	Description
31:24	WO	0x00	Instruction byte 1
23:16	WO	0x00	Instruction byte 0
15:11	RO	0x0	reserved
10:8	WO	0x0	DMA channel number: b000 = DMA channel 0 b001 = DMA channel 1 b010 = DMA channel 2 … b111 = DMA channel 7
7:1	RO	0x0	reserved
0	WO	0x0	The debug thread encoding is as follows: 0 = DMA manager thread 1 = DMA channel.

DMAC_DBGINST1
Address: Operational Base + offset (0x0d0c)
Debug Instruction-1 Register

Bit	Attr	Reset Value	Description
31:24	WO	0x00	Instruction byte 5
23:16	WO	0x00	Instruction byte 4
15:8	WO	0x00	Instruction byte 3
7:0	WO	0x00	Instruction byte 2

DMAC_CR0
Address: Operational Base + offset (0x0e00)
Configuration Register 0

Bit	Attr	Reset Value	Description
31:22	RO	0x0	reserved
21:17	RO	0x02	Number of interrupt outputs that the DMAC provides: b00000 = 1 interrupt output, irq[0] b00001 = 2 interrupt outputs, irq[1:0] b00010 = 3 interrupt outputs, irq[2:0] … b11111 = 32 interrupt outputs, irq[31:0].
16:12	RO	0x07	Number of peripheral request interfaces that the DMAC provides: b00000 = 1 peripheral request interface b00001 = 2 peripheral request interfaces b00010 = 3 peripheral request interfaces … b11111 = 32 peripheral request interfaces.
11:7	RO	0x0	reserved
6:4	RO	0x5	Number of DMA channels that the DMAC supports: b000 = 1 DMA channel b001 = 2 DMA channels b010 = 3 DMA channels … b111 = 8 DMA channels.
3	RO	0x0	reserved
2	RO	0x0	Indicates the status of the boot_manager_ns signal when the DMAC exited from reset: 0 = boot_manager_ns was LOW 1 = boot_manager_ns was HIGH.
1	RO	0x0	Indicates the status of the boot_from_pc signal when the DMAC exited from reset: 0 = boot_from_pc was LOW 1 = boot_from_pc was HIGH
0	RO	0x1	Supports peripheral requests: 0 = the DMAC does not provide a peripheral request interface 1 = the DMAC provides the number of peripheral request interfaces that the num_periph_req field specifies.

DMAC_CR1
Address: Operational Base + offset (0x0e04)
Configuration Register 1

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved

Bit	Attr	Reset Value	Description
7:4	RO	0x5	[7:4] num_i-cache_lines Number of i-cache lines: b0000 = 1 i-cache line b0001 = 2 i-cache lines b0010 = 3 i-cache lines … b1111 = 16 i-cache lines.
3	RO	0x0	reserved
2:0	RO	0x7	The length of an i-cache line: b000-b001 = reserved b010 = 4 bytes b011 = 8 bytes b100 = 16 bytes b101 = 32 bytes b110-b111 = reserved

DMAC_CR2
Address: Operational Base + offset (0x0e08)
Configuration Register 2

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	Provides the value of boot_addr[31:0] when the DMAC exited from reset

DMAC_CR3
Address: Operational Base + offset (0x0e0c)
Configuration Register 3

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	Provides the security state of an event-interrupt resource: Bit [N] = 0 Assigns event<N> or irq[N] to the Secure state. Bit [N] = 1 Assigns event<N> or irq[N] to the Non-secure state.

DMAC_CR4
Address: Operational Base + offset (0x0e10)
Configuration Register 4

Bit	Attr	Reset Value	Description
31:0	RO	0x00000006	Provides the security state of the peripheral request interfaces: Bit [N] = 0 Assigns peripheral request interface N to the Secure state. Bit [N] = 1 Assigns peripheral request interface N to the Non-secure state

DMAC_CRDn
Address: Operational Base + offset (0x0e14)
DMA Configuration Register

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved

Bit	Attr	Reset Value	Description
29:20	RO	0x020	The number of lines that the data buffer contains: b000000000 = 1 line b000000001 = 2 lines … b111111111 = 1024 lines
19:16	RO	0x9	The depth of the read queue: b0000 = 1 line b0001 = 2 lines … b1111 = 16 lines.
15	RO	0x0	reserved
14:12	RO	0x4	Read issuing capability that programs the number of outstanding read transactions: b000 = 1 b001 = 2 … b111 = 8
11:8	RO	0x7	The depth of the write queue: b0000 = 1 line b0001 = 2 lines … b1111 = 16 lines.
7	RO	0x0	reserved
6:4	RO	0x3	Write issuing capability that programs the number of outstanding write transactions: b000 = 1 b001 = 2 … b111 = 8
3	RO	0x0	reserved
2:0	RO	0x3	The data bus width of the AXI interface: b000 = reserved b001 = reserved b010 = 32-bit b011 = 64-bit b100 = 128-bit b101-b111 = reserved.

DMAC_WD

Address: Operational Base + offset (0x0e80)

DMA Watchdog Register

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved

Bit	Attr	Reset Value	Description
0	RW	0x0	Controls how the DMAC responds when it detects a lock-up condition: 0 = the DMAC aborts all of the contributing DMA channels and sets irq_abort HIGH 1 = the DMAC sets irq_abort HIGH.

13.5 Timing Diagram

Following picture shows the relationship between dma_req and dma_ack.

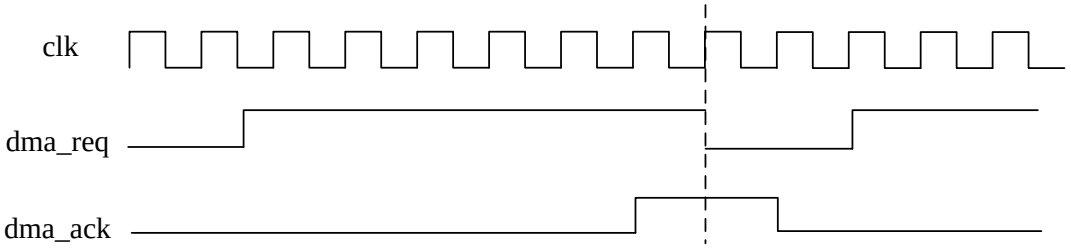


Fig. 13-3 DMAC request and acknowledge timing

13.6 Interface Description

DMAC has the following tie-off signals. It can be configured by SGRF register. (Please refer to the chapter to find how to configure)

Table 13-3 DMAC0 boot interface

Interface	Reset value	Control source
boot_addr	0x0	SGRF
boot_from_pc	0x0	SGRF
boot_manager_ns	0x1	SGRF
boot_irq_ns	0xfff	SGRF
boot_periph_ns	0xfff	SGRF

Table 13-4 DMAC1 boot interface

Interface	Reset value	Control source
boot_addr	0x0	SGRF
boot_from_pc	0x0	SGRF
boot_manager_ns	0x1	SGRF
boot_irq_ns	0xffff	SGRF
boot_periph_ns	0xfffff	SGRF

boot_addr
Configures the address location that contains the first instruction the DMAC executes, when it exits from reset.
boot_from_pc
Controls the location in which the DMAC executes its initial instruction, after it exits from reset:
0 = DMAC waits for an instruction from either APB interface
1 = DMA manager thread executes the instruction that is located at the address that
boot_manager_ns
When the DMAC exits from reset, this signal controls the security state of the DMA manager thread:
0 = assigns DMA manager to the Secure state
1 = assigns DMA manager to the Non-secure state.

boot_irq_ns
Controls the security state of an event-interrupt resource, when the DMAC exits from reset:
boot_irq_ns[x] is LOW
The DMAC assigns event<x> or irq[x] to the Secure state.
boot_irq_ns[x] is HIGH
The DMAC assigns event<x> or irq[x] to the Non-secure state.
boot_periph_ns
Controls the security state of a peripheral request interface, when the DMAC exits from reset:
boot_periph_ns[x] is LOW
The DMAC assigns peripheral request interface x to the Secure state.
boot_periph_ns[x] is HIGH
The DMAC assigns peripheral request interface x to the Non-secure state.

13.7 Application Notes

13.7.1 Using the APB slave interfaces

You must ensure that you use the appropriate APB interface, depending on the security state in which the boot_manager_ns initializes the DMAC to operate. For example, if the DMAC is in the secure state, you must issue the instruction using the secure APB interface, otherwise the DMAC ignores the instruction. You can use the secure APB interface, or the non-secure APB interface, to start or restart a DMA channel when the DMAC is in the Non-secure state.
The necessary steps to start a DMA channel thread using the debug instruction registers as following:
1. Create a program for the DMA channel.
2. Store the program in a region of system memory.
3. Poll the DBGSTATUS Register to ensure that debug is idle, that is, the dbgstatus bit is 0.
4. Write to the DBGINST0 Register and enter the:
- Instruction byte 0 encoding for DMAGO.
- Instruction byte 1 encoding for DMAGO.
- Debug thread bit to 0. This selects the DMA manager thread.

5. Write to the DBGINST1 Register with the DMAGO instruction byte [5:2] data, see Debug Instruction-1 Register o. You must set these four bytes to the address of the first instruction in the program, that was written to system memory in step 2.
6. Writing zero to the DBGCMD Register. The DMAC starts the DMA channel thread and sets the dbgstatus bit to 1.

13.7.2 Security usage

DMA manager thread is in the secure state
If the DNS bit is 0, the DMA manager thread operates in the secure state and it only performs secure instruction fetches. When a DMA manager thread in the secure state processes:
DMAGO
It uses the status of the ns bit, to set the security state of the DMA channel thread by writing to the CNS bit for that channel.
DMAWFE
It halts execution of the thread until the event occurs. When the event occurs, the DMAC continues execution of the thread, irrespective of the security state of the corresponding INS bit.
DMASEV
It sets the corresponding bit in the INT_EVENT_RIS Register, irrespective of the security state of the corresponding INS bit.
DMA manager thread is in the Non-secure state
If the DNS bit is 1, the DMA manager thread operates in the Non-secure state, and it only performs non-secure instruction fetches. When a DMA manager thread in the Non-secure state processes:
DMAGO
The DMAC uses the status of the ns bit, to control if it starts a DMA channel

thread. If:
ns = 0
The DMAC does not start a DMA channel thread and instead it:
1. Executes a NOP.
2. Sets the FSRD Register, see Fault Status DMA Manager
3. Sets the dmago_err bit in the FTRD Register, see Fault Type DMA Manager Register.
4. Moves the DMA manager to the Faulting state.
ns = 1
The DMAC starts a DMA channel thread in the Non-secure state and programs the CNS bit to be non-secure.

DMAWFE

The DMAC uses the status of the corresponding INS bit, in the CR3 Register, to control if it waits for the event. If:
INS = 0
The event is in the Secure state. The DMAC:
1. Executes a NOP.
2. Sets the FSRD Register, see Fault Status DMA Manager Register.
3. Sets the mgr_evnt_err bit in the FTRD Register, see Fault Type DMA Manager Register.
4. Moves the DMA manager to the Faulting state.
INS = 1
The event is in the Non-secure state. The DMAC halts execution of the thread and waits for the event to occur.

DMASEV

The DMAC uses the status of the corresponding INS bit, in the CR3Register, to control if it creates the event-interrupt. If:
INS = 0
The event-interrupt resource is in the secure state. The DMAC:
1. Executes a NOP.
2. Sets the FSRD Register, see Fault Status DMA Manager Register.
3. Sets the mgr_evnt_err bit in the FTRD Register, see Fault Type DMA Manager Register.
4. Moves the DMA manager to the Faulting state.
INS = 1
The event-interrupt resource is in the Non-secure state. The DMAC creates the event-interrupt.

DMA channel thread is in the secure state

When the CNS bit is 0, the DMA channel thread is programmed to operate in the Secure state and it only performs secure instruction fetches.
When a DMA channel thread in the secure state processes the following instructions:

DMAWFE

The DMAC halts execution of the thread until the event occurs. When the event occurs, the DMAC continues execution of the thread, irrespective of the security state of the corresponding INS bit, in the CR3 Register.

DMASEV

The DMAC creates the event-interrupt, irrespective of the security state of the corresponding INS bit, in the CR3 Register.

DMAWFP

The DMAC halts execution of the thread until the peripheral signals a DMA request. When this occurs, the DMAC continues execution of the thread, irrespective of the security state of the corresponding PNS bit, in the CR4 Register.

DMALDP, DMASTP

The DMAC sends a message to the peripheral to communicate that data transfer is complete, irrespective of the security state of the corresponding PNS bit, in the CR4 Register.

DMAFLUSHP

The DMAC clears the state of the peripheral and sends a message to the peripheral to resend its level status, irrespective of the security state of the corresponding PNS bit, in the CR4 Register.

When a DMA channel thread is in the Secure state, it enables the DMAC to perform secure and non-secure AXI accesses

DMA channel thread is in the Non-secure state

When the CNS bit is 1, the DMA channel thread is programmed to operate in the Non-secure state and it only performs non-secure instruction fetches.

When a DMA channel thread in the Non-secure state processes the following instructions:

DMAWFE

The DMAC uses the status of the corresponding INS bit, in the CR3 Register, to control if it waits for the event. If:

INS = 0

The event is in the Secure state. The DMAC:

1. Executes a NOP.
2. Sets the appropriate bit in the FSRC Register that corresponds to the DMA channel number. See Fault Status DMA Channel Register.
3. Sets the ch_evnt_err bit in the FTRn Register, see Fault Type DMA Channel Registers.
4. Moves the DMA channel to the Faulting completing state.

INS = 1

The event is in the Non-secure state. The DMAC halts execution of the thread and waits for the event to occur.

DMASEV

The DMAC uses the status of the corresponding INS bit, in the CR3 Register, to control if it creates the event. If:

INS = 0

The event-interrupt resource is in the Secure state. The DMAC:

1. Executes a NOP.
2. Sets the appropriate bit in the FSRC Register that corresponds to the DMA channel number. See Fault Status DMA Channel Register.
3. Sets the ch_evnt_err bit in the FTRn Register, see Fault Type DMA Channel Registers .
4. Moves the DMA channel to the Faulting completing state.

INS = 1

The event-interrupt resource is in the Non-secure state. The DMAC creates the event-interrupt.

DMAWFP

The DMAC uses the status of the corresponding PNS bit, in the CR4 Register, to control if it waits for the peripheral to signal a request. If:

PNS = 0

The peripheral is in the Secure state. The DMAC:

1. Executes a NOP.
2. Sets the appropriate bit in the FSRC Register that corresponds to the DMA channel number. See Fault Status DMA Channel Register.
3. Sets the ch_periph_err bit in the FTRn Register, see Fault Type DMA Channel Registers.
4. Moves the DMA channel to the Faulting completing state.

PNS = 1

The peripheral is in the Non-secure state. The DMAC halts execution of the thread and waits for the peripheral to signal a request.

DMALDP, DMASTP

The DMAC uses the status of the corresponding PNS bit, in the CR4 Register, to control if it sends an acknowledgement to the peripheral. If:

PNS = 0

The peripheral is in the secure state. The DMAC:

1. Executes a NOP.
2. Sets the appropriate bit in the FSRC Register that corresponds to the DMA channel number. See Fault Status DMA Channel Register.
3. Sets the ch_periph_err bit in the FTRn Register, see Fault Type DMA Channel Registers.
4. Moves the DMA channel to the Faulting completing state.

PNS = 1

The peripheral is in the Non-secure state. The DMAC sends a message to the peripheral to

communicate when the data transfer is complete.

DMAFLUSHP

The DMAC uses the status of the corresponding PNS bit, in the CR4 Register, to control if it sends a flush request to the peripheral. If:

PNS = 0

The peripheral is in the secure state. The DMAC:

1. Executes a NOP.
2. Sets the appropriate bit in the FSRC Register that corresponds to the DMA channel number. See Fault Status DMA Channel Register.
3. Sets the ch_periph_err bit in the FTRn Register, see Fault Type DMA Channel Registers.
4. Moves the DMA channel to the Faulting completing state.

PNS = 1

The peripheral is in the Non-secure state. The DMAC clears the state of the peripheral and sends a message to the peripheral to resend its level status.

When a DMA channel thread is in the Non-secure state, and a DMAMOV CCR instruction attempts to program the channel to perform a secure AXI transaction, the DMAC:

1. Executes a DMANOP.
2. Sets the appropriate bit in the FSRC Register that corresponds to the DMA channel number. See Fault Status DMA Channel Register.
3. Sets the ch_rdwr_err bit in the FTRn Register, see Fault Type DMA Channel Registers.
4. Moves the DMA channel thread to the Faulting completing state.

13.7.3 Programming restrictions

Fixed unaligned bursts

The DMAC does not support fixed unaligned bursts. If you program the following conditions, the DMAC treats this as a programming error:

Unaligned read

- src_inc field is 0 in the CCRn Register
- the SARn Register contains an address that is not aligned to the size of data that the src_burst_size field contain

Unaligned write

- dst_inc field is 0 in the CCRn Register
- the DARn Register contains an address that is not aligned to the size of data that the dst_burst_size field contains

Endian swap size restrictions

If you program the endian_swap_size field in the CCRn Register, to enable a DMA channel to perform an endian swap then you must set the corresponding SARn Register and the corresponding DARn Register to contain an address that is aligned to the value that the endian_swap_size field contains.

Updating DMA channel control registers during a DMA cycle restrictions

Prior to the DMAC executing a sequence of DMALD and DMAST instructions, the values you program in to the CCRn Register, SARn Register, and DARn Register control the data byte lane manipulation that the DMAC performs when it transfers the data from the source address to the destination address. You'd better not update these registers during a DMA cycle.

Resource sharing between DMA channels

DMA channel programs share the MFIFO data storage resource. You must not start a set of concurrently running DMA channel programs with a resource requirement that exceeds the configured size of the MFIFO. If you exceed this limit then the DMAC might lock up and generate a Watchdog abort.

13.7.4 Unaligned transfers may be corrupted

For a configuration with more than one channel, if any of channels 1 to 7 is performing transfers between certain types of misaligned source and destination addresses, then the output data may be corrupted by the action of channel 0.

Data corruption might occur if all of the following are true:

1. Two beats of AXI read data are received for one of channels 1 to 7.
2. Source and destination address alignments mean that each read data beat is splited across two lines in the data buffer (see Splitting data, below).

3. There is one idle cycle between the two read data beats.
4. Channel 0 performs an operation that updates channel control information during this idle cycle (see Updates to channel control information, below)

Splitting data

Depending upon the programmed values for the DMA transfer, one beat of read data from the AXI interface need to be splited across two lines in the internal data buffer. This occurs when the read data beat contains data bytes which will be written to addresses that wrap around at the AXI interface data width, so that these bytes could not be transferred by a single AXI write data beat of the full interface width.

Most applications of DMA-330 do not split data in this way, so are NOT vulnerable to data corruption from this defect.

The following cases are NOT vulnerable to data corruption because they do not split data:

- Byte lane offset between source and destination addresses is 0 when source and destination addresses have the same byte lane alignment, the offset is 0 and a wrap operation that splits data cannot occur.
- Byte lane offset between source and destination addresses is a multiple of source size

Table 13-5 Source size in CCRn

Source size in CCRn	Allowed offset between SARn and DARn
SS8	any offset allowed.
SS16	0,2,4,6,8,10,12,14
SS32	0,4,8,12
SS64	0,8

13.7.5 Interrupt shares between channel

As the DMAC does not record which channel (or list of channels) have asserted an interrupt. So it will depend on your program and whether any of the visible information for that program can be used to determine progress, and help identify the interrupt source. There are 4 likely information sources that can be used to determine the progress made by a program:

- Program counter (PC)
- Source address
- Destination address
- Loop counters (LC)

For example, a program might emit an interrupt each time that it iterates around a loop. In this case, the interrupt service routine (ISR) would need to store the loop value of each channel when it is called, and then compare against the new value when it is next called. A change in value would indicate that the program has progressed.

The ISR must be carefully written to ensure that no interrupts are lost. The sequence of operations is as follows:

1. Disable interrupts
2. Immediately clear the interrupt in DMA-330
3. Check the relevant registers for both channels to determine which must be serviced
4. Take appropriate action for the channels
5. Re-enable interrupts and exit ISR

13.7.6 Instruction sets

Table 13-6 DMAC Instruction sets

Mnemonic	Instruction	Thread usage
DMAADDH	Add Halfword	C
DMAEND	End	M/C
DMAFLUSHP	Flush and notify Peripheral	C
DMAGO	Go	M
DMAKILL	Kill	C
DMALD	Load	C
DMALDP	Load Peripheral	C
DMALP	Loop	C
DMALPEND	Loop End	C
DMALPFE	Loop Forever	C

Mnemonic	Instruction	Thread usage
DMAMOV	Move	C
DMANOP	No operation	M/C
DMARMB	Read Memory Barrier	C
DMASEV	Send Event	M/C
DMAST	Store	C
DMASTP	Store and notify Peripheral	C
DMASTZ	Store Zero	C
DMAWFE	Wait For Event M	M/C
DMAWFP	Wait For Peripheral	C
DMAWMB	Write Memory Barrier	C
DMAADNH	Add Negative Halfword	C

Notes: Thread usage: C=DMA channel, M=DMA manager

13.7.7 Assembler directives

In this document, only DMMADNH instruction is took as an example to show the way the instruction assembled.

DMAADNH

Add Negative Halfword adds an immediate negative 16-bit value to the SARn Register or DARn Register, for the DMA channel thread. This enables the DMAC to support 2D DMA operations, or reading or writing an area of memory in a different order to naturally incrementing addresses. See Source Address Registers and Destination Address Registers. The immediate unsigned 16-bit value is one-extended to 32 bits, to create a value that is the two's complement representation of a negative number between -65536 and -1, before the DMAC adds it to the address using 32-bit addition. The DMAC discards the carry bit so that addresses wrap from 0xFFFFFFFF to 0x00000000. The net effect is to subtract between 65536 and 1 from the current value in the Source or Destination Address Register.
Following table shows the instruction encoding.

Table 13-7 DMAC instruction encoding

Imm[15:8]	Imm[7:0]	0	1	0	1	1	1	ra	0

Assembler syntax

DMAADNH <address_register>, <16-bit immediate>
where:
<address_register>
Selects the address register to use. It must be either:
SAR
SARn Register and sets ra to 0.
DAR
DARn Register and sets ra to 1.
<16-bit immediate>
The immediate value to be added to the <address_register>.
You should specify the 16-bit immediate as the number that is to be represented in the instruction encoding. For example, DMAADNH DAR, 0xFFF0 causes the value 0xFFFFFFF0 to be added to the current value of the Destination Address Register, effectively subtracting 16 from the DAR.
You can only use this instruction in a DMA channel thread.

Chapter 14 Temperature Sensor ADC (TSADC)

14.1 Overview

TS-ADC Controller module supports user-defined mode and automatic mode.User-defined mode refers, TSADC all the control signals entirely by software writing to register for direct control.Automatic mode refers to the module automatically poll TSADC output, and the results were checked. If you find that the temperatureHigh in a period of time, an interrupt is generated to the processor down-measures taken; if the temperature over a period of timeHigh, the resulting TSHUT gave CRU module, let it reset the entire chip, or via GPIO give PMIC.

TS-ADC Controller supports the following features:

- Support User-Defined Mode and Automatic Mode
- In User-Defined Mode, start_of_conversion can be controlled completely by software, and also can be generated by hardware.
- In Automatic Mode, the temperature of alarm(high/low temperature) interrupt can be configurable
- In Automatic Mode, the temperature of system reset can be configurable
- Support to 2 channel TS-ADC, the temperature criteria of each channel can be configurable
- In Automatic Mode, the time interval of temperature detection can be configurable
- In Automatic Mode, when detecting a high temperature, the time interval of temperature detection can be configurable
- High temperature debounce can be configurable
- -40~125°C temperature range and 5°C temperature resolution
- 10-bit SARADC up to 50KS/s sampling rate

14.2 Block Diagram

TS-ADC controller comprises with:

- APB Interface
- TS-ADC control logic

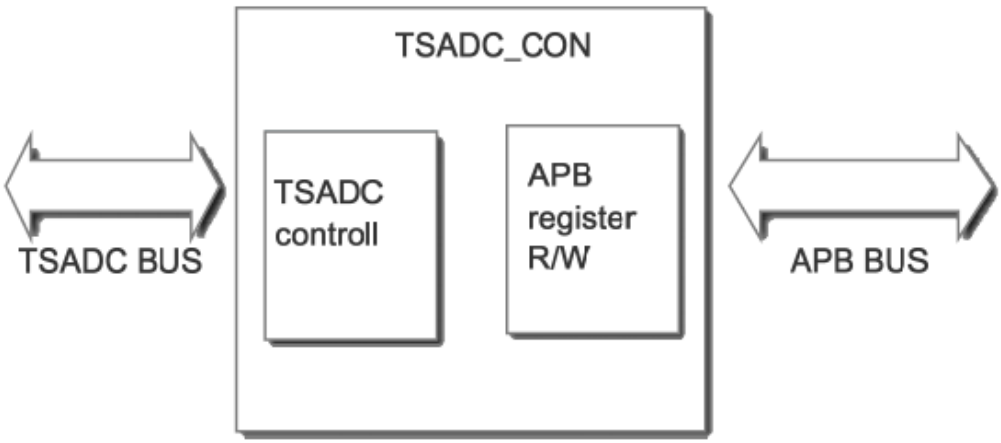


Fig. 14-1 TS-ADC Controller Block Diagram

14.3 Function Description

14.3.1 APB Interface

There is an APB Slave interface in TS-ADC Controller, which is used to configure the TS-ADC Controller registers and look up the temperature from the temperature sensor.

14.3.2 TS-ADC Controller

This block is exploited to realize binary search algorithm, storing the intermediate result and generate control signal for analog block.This block compares the analog input with the voltage

generated from D/A Converter, and output the comparison result to SAR and Control Logic Block for binary search. Three level amplifiers are employed in this comparator to provide enough gain.

14.4 Register description

14.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
TSADC_USER_CON	0x0000	W	0x00000200	The control register of A/D Converter.
TSADC_AUTO_CON	0x0004	W	0x00000000	TSADC auto mode control register
TSADC_INT_EN	0x0008	W	0x00000000	
TSADC_INT_PD	0x000c	W	0x00000000	
TSADC_DATA0	0x0020	W	0x00000000	This register contains the data after A/D Conversion.
TSADC_DATA1	0x0024	W	0x00000000	This register contains the data after A/D Conversion.
TSADC_COMP0_INT	0x0030	W	0x00000000	TSADC high temperature level for source 0
TSADC_COMP1_INT	0x0034	W	0x00000000	TSADC high temperature level for source 1
TSADC_COMP0_SHUT	0x0040	W	0x00000000	TSADC high temperature level for source 0
TSADC_COMP1_SHUT	0x0044	W	0x00000000	TSADC high temperature level for source 1
TSADC_HIGHT_INT_DEBOUNCE	0x0060	W	0x00000003	high temperature debounce
TSADC_HIGHT_TSHUT_DEBOUNCE	0x0064	W	0x00000003	high temperature debounce
TSADC_AUTO_PERIOD	0x0068	W	0x00010000	TSADC auto access period
TSADC_AUTO_PERIOD_HT	0x006c	W	0x00010000	TSADC auto access period when temperature is high
TSADC_COMP0_LOW_INT	0x0080	W	0x00000000	TSADC low temperature level for source 0
TSADC_COMP1_LOW_INT	0x0084	W	0x00000000	TSADC low temperature level for source 1

*Notes:Size:**B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access*

14.4.2 Detail Register Description

TSADC_USER_CON

Address: Operational Base + offset (0x0000)

The control register of A/D Converter.

Bit	Attr	Reset Value	Description
31:13	RO	0x0	reserved
12	RO	0x0	adc_status ADC status (EOC) 0: ADC stop 1: Conversion in progress

Bit	Attr	Reset Value	Description
11:6	RW	0x08	inter_pd_soc interleave between power down and start of conversion
5	RW	0x0	start When software write 1 to this bit , start_of_conversion will be assert. This bit will be cleared after TSADC access finishing. When TSADC_USER_CON[4] = 1'b1 take effect.
4	RW	0x0	start_mode start mode. 0: tsadc controller will asert start_of_conversion after "inter_pd_soc" cycles. 1: the start_of_conversion will be controlled by TSADC_USER_CON[5].
3	RW	0x0	adc_power_ctrl ADC power down control bit 0: ADC power down 1: ADC power up and reset
2:0	RW	0x0	adc_input_src_sel ADC input source selection(CH_SEL[2:0]). 000 : Input source 0 (SARADC_AIN[0]) 001 : Input source 1 (SARADC_AIN[1]) Others : Reserved

TSADC_AUTO_CON
Address: Operational Base + offset (0x0004)
TSADC auto mode control register

Bit	Attr	Reset Value	Description
31:26	RO	0x0	reserved
25	RW	0x0	last_tshut_2cru last_tshut_2cru for cru first/second reset TSHUT status. This bit will set to 1 when tshut is valid, and only be cleared when application write 1 to it. This bit will not be cleared by system reset.
24	RW	0x0	last_tshut_2gpio last_tshut_2gpio for hardware reset TSHUT status. This bit will set to 1 when tshut is valid, and only be cleared when application write 1 to it. This bit will not be cleared by system reset.
23:18	RO	0x0	reserved
17	RO	0x0	sample_dly_sel 0: AUTO_PERIOD is used 1: AUTO_PERIOD_HT is used

Bit	Attr	Reset Value	Description
16	RO	0x0	auto_status 0: auto mode stop; 1: auto mode in progress.
15:14	RO	0x0	reserved
13	RW	0x0	src1_lt_en 0: do not care low temperature of source 0 1: enable the low temperature monitor of source 0
12	RW	0x0	src0_lt_en 0: do not care low temperature of source 0 1: enable the low temperature monitor of source 0
11:9	RO	0x0	reserved
8	RW	0x0	tshut_prolarity 0: low active 1: high active
7:6	RO	0x0	reserved
5	RW	0x0	src1_en 0: do not care the temperature of source 1 1: if the temperature of source 0 is too high , TSHUT will be valid
4	RW	0x0	src0_en 0: do not care the temperature of source 0 1: if the temperature of source 0 is too high , TSHUT will be valid
3:2	RO	0x0	reserved
1	RW	0x0	tsadc_q_sel temperature coefficient 1'b0:use tsadc_q as output(positive temperature coefficient) 1'b1:use(1024 - tsadc_q) as output (negative temperature coefficient) RK3399Pro is negative temprature coefficient, so please set this bit as 1'b1
0	RW	0x0	auto_en 0: TSADC controller works at user-define mode 1: TSADC controller works at auto mode

TSADC_INT_EN
Address: Operational Base + offset (0x0008)

Bit	Attr	Reset Value	Description
31:17	RO	0x0	reserved
16	RW	0x0	eoc_int_en eoc_Interrupt enable. eoc_interrupt enable in user defined mode 0: disable 1: enable
15:14	RO	0x0	reserved

Bit	Attr	Reset Value	Description
13	RW	0x0	lt_inten_src1 low temperature interrupt enable for src1 0: disable 1: enable
12	RW	0x0	lt_inten_src0 low temperature interrupt enable for src0 0: disable 1: enable
11:10	RO	0x0	reserved
9	RW	0x0	tshut_2cru_en_src1 0: TSHUT output to cru disabled. TSHUT output will always keep low . 1: TSHUT output works.
8	RW	0x0	tshut_2cru_en_src0 0: TSHUT output to cru disabled. TSHUT output will always keep low . 1: TSHUT output works.
7:6	RO	0x0	reserved
5	RW	0x0	tshut_2gpio_en_src1 0: TSHUT output to gpio disabled. TSHUT output will always keep low . 1: TSHUT output works.
4	RW	0x0	tshut_2gpio_en_src0 0: TSHUT output to gpio disabled. TSHUT output will always keep low . 1: TSHUT output works.
3:2	RO	0x0	reserved
1	RW	0x0	ht_inten_src1 high temperature interrupt enable for src1 0: disable 1: enable
0	RW	0x0	ht_inten_src0 high temperature interrupt enable for src0 0: disable 1: enable

TSADC_INT_PD
Address: Operational Base + offset (0x000c)

Bit	Attr	Reset Value	Description
31:17	RO	0x0	reserved
16	RW	0x0	eoc_int_pd Interrupt status. This bit will be set to 1 when end-of-conversion. Set 0 to clear the interrupt.

Bit	Attr	Reset Value	Description
15:14	RO	0x0	reserved
13	RW	0x0	lt_irq_src1 When TSADC output is lower than COMP_INT_LOW, this bit will be valid, which means temperature is low, and the application should in charge of this. write 1 to it , this bit will be cleared.
12	RW	0x0	lt_irq_src0 When TSADC output is lower than COMP_INT_LOW, this bit will be valid, which means temperature is low, and the application should in charge of this. write 1 to it , this bit will be cleared.
11:6	RO	0x0	reserved
5	RW	0x0	tshut_o_src1 TSHUT output status When TSADC output is bigger than COMP_SHUT, this bit will be valid, which means temperature is VERY high, and the application should in charge of this. write 1 to it , this bit will be cleared.
4	RW	0x0	tshut_o_src0 TSHUT output status When TSADC output is bigger than COMP_SHUT, this bit will be valid, which means temperature is VERY high, and the application should in charge of this. write 1 to it , this bit will be cleared.
3:2	RO	0x0	reserved
1	RW	0x0	ht_irq_src1 When TSADC output is bigger than COMP_INT, this bit will be valid, which means temperature is high, and the application should in charge of this. write 1 to it , this bit will be cleared.
0	RW	0x0	ht_irq_src0 When TSADC output is bigger than COMP_INT, this bit will be valid, which means temperature is high, and the application should in charge of this. write 1 to it , this bit will be cleared.

TSADC_DATA0
Address: Operational Base + offset (0x0020)
This register contains the data after A/D Conversion.

Bit	Attr	Reset Value	Description
31:12	RO	0x0	reserved
11:0	RO	0x000	adc_data A/D value of the channel 0 last conversion (DOUT[9:0]).

TSADC_DATA1
Address: Operational Base + offset (0x0024)
This register contains the data after A/D Conversion.

Bit	Attr	Reset Value	Description
31:12	RO	0x0	reserved
11:0	RO	0x000	adc_data A/D value of the channel 0 last conversion (DOUT[9:0]).

TSADC_COMP0_INT
Address: Operational Base + offset (0x0030)
TSADC high temperature level for source 0

Bit	Attr	Reset Value	Description
31:12	RO	0x0	reserved
11:0	RW	0x000	tsadc_comp_src0 TSADC high temperature level. TSADC output is bigger than tsadc_comp, means the temperature is high. TSADC_INT will be valid.

TSADC_COMP1_INT
Address: Operational Base + offset (0x0034)
TSADC high temperature level for source 1

Bit	Attr	Reset Value	Description
31:12	RO	0x0	reserved
11:0	RW	0x000	tsadc_comp_src1 TSADC high temperature level. TSADC output is bigger than tsadc_comp, means the temperature is high. TSADC_INT will be valid.

TSADC_COMP0_SHUT
Address: Operational Base + offset (0x0040)
TSADC high temperature level for source 0

Bit	Attr	Reset Value	Description
31:12	RO	0x0	reserved
11:0	RW	0x000	tsadc_comp_src0 TSADC high temperature level. TSADC output is bigger than tsadc_comp, means the temperature is too high. TSHUT will be valid.

TSADC_COMP1_SHUT
Address: Operational Base + offset (0x0044)
TSADC high temperature level for source 1

Bit	Attr	Reset Value	Description
31:12	RO	0x0	reserved
11:0	RW	0x000	tsadc_comp_src1 TSADC high temperature level. TSADC output is bigger than tsadc_comp, means the temperature is too high. TSHUT will be valid.

TSADC_HIGHT_INT_DEBOUNCE
Address: Operational Base + offset (0x0060)
high temperature debounce

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RW	0x03	debounce TSADC controller will only generate interrupt or TSHUT when temperature is higher than COMP_INT for "debounce" times.

TSADC_HIGHT_TSHUT_DEBOUNCE
Address: Operational Base + offset (0x0064)
high temperature debounce

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RW	0x03	debounce TSADC controller will only generate interrupt or TSHUT when temperature is higher than COMP_SHUT for "debounce" times.

TSADC_AUTO_PERIOD
Address: Operational Base + offset (0x0068)
TSADC auto access period

Bit	Attr	Reset Value	Description
31:0	RW	0x00010000	auto_period when auto mode is enabled, this register controls the interleave between every two accessing of TSADC.

TSADC_AUTO_PERIOD_HT
Address: Operational Base + offset (0x006c)
TSADC auto access period when temperature is high

Bit	Attr	Reset Value	Description
31:0	RW	0x00010000	auto_period This register controls the interleave between every two accessing of TSADC after the temperature is higher than COMP_SHUT or COMP_INT

TSADC_COMP0_LOW_INT
Address: Operational Base + offset (0x0080)
TSADC low temperature level for source 0

Bit	Attr	Reset Value	Description
31:12	RO	0x0	reserved
11:0	RW	0x000	tsadc_comp_src0 TSADC low temperature level. TSADC output is lower than tsadc_comp, means the temperature is low. TSADC_LOW_INT will be valid.

TSADC_COMP1_LOW_INT
Address: Operational Base + offset (0x0084)
TSADC low temperature level for source 1

Bit	Attr	Reset Value	Description
31:12	RO	0x0	reserved
11:0	RW	0x000	tsadc_comp_src1 TSADC low temperature level. TSADC output is lower than tsadc_comp, means the temperature is low. TSADC_LOW_INT will be valid.

Application Notes

14.4.3 Channel Select

The system has two Temperature Sensors, channel 0 is for CPU, channel 1 is for GPU.

14.4.4 Single-sample conversion

- The start timing

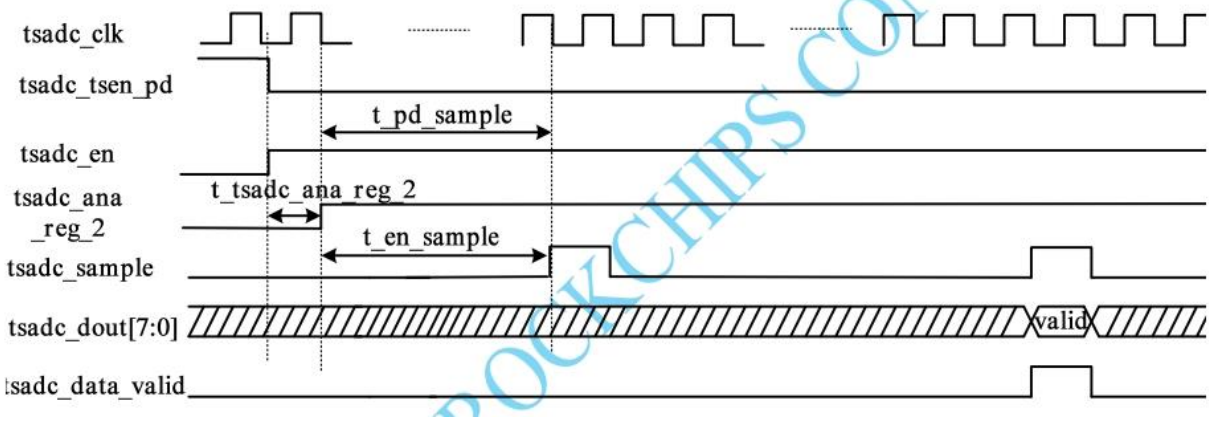


Fig. 14-2 the start flow to enable the sensor and adc

- Bypass mode(grf_sco_con1[1] = 1'b1)
  When the ADC bypass mode is enable(tsadc_dig_bypass = 1'b1), the ADC will cost 14 clock cycles to complete the conversion. The tsadc_dout will keep the valid data output unchanged untile the next clock cycles when the tsadc_sample is valid.

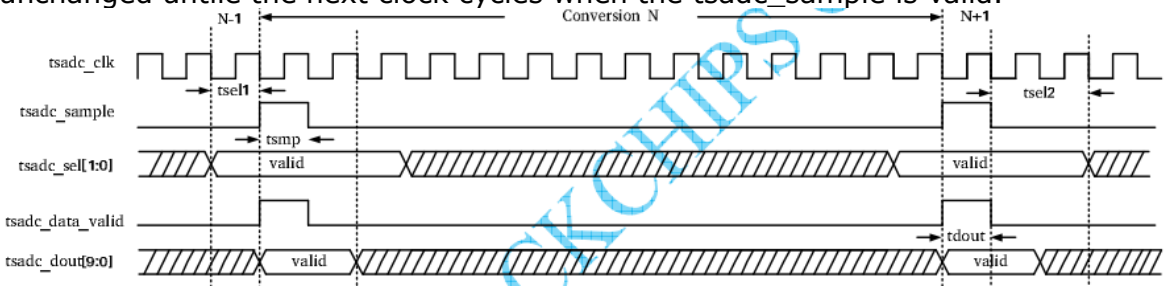


Fig. 14-3 tsadc timing diagram in bypass mode

- The Normal mode

- Tsadc_clk_sel = 1'b0(grf_sco_con1[0] = 1'b0)
  The ADC will cost 15 clock cycles to complete the conversion.The tsadc_dout will keep the valid data output unchanged untile the next two clock cycles when the tsadc_sample is valid.

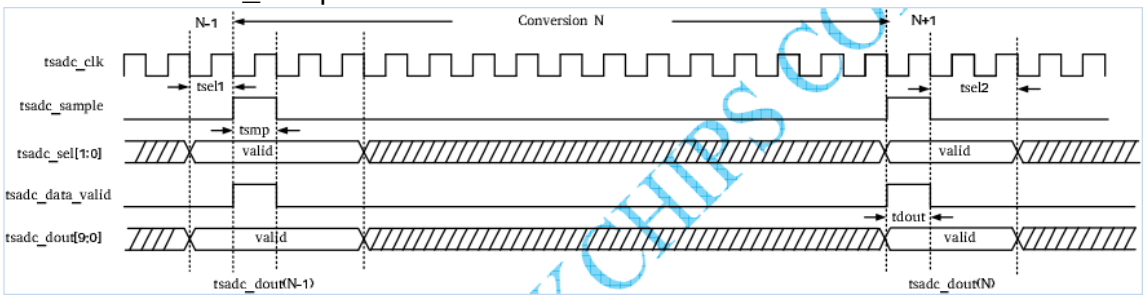


Fig. 14-4 tsadc timing diagram in normal mode with tsadc_clk_sel = 1'b0

- Tsadc_clk_sel = 1'b1(grf_sco_con1[0] = 1'b1)
  The ADC will cost 16 clock cycles to complete the conversion.The tsadc_dout will keep the valid data output unchanged untile the next three clock cycles when the tsadc_sample is valid.

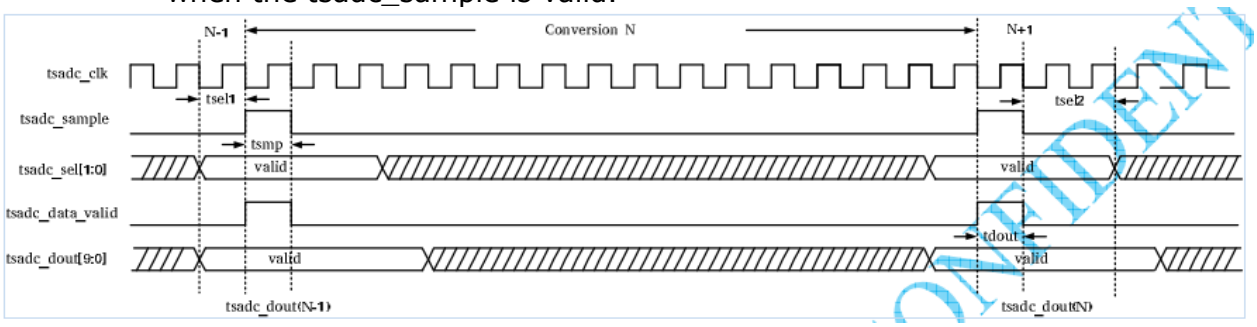


Fig. 14-5 tsadc timing diagram in normal mode with tsadc_clk_sel = 1'b1

14.4.5 Temperature-to-code mapping

Table 14-1 Temperature Code Mapping

Temperature/℃	ADC Output Data AUTO_CON[1] = 1'b0			ADC Output Data AUTO_CON[1] = 1'b1		
	Min	Typ	Max	Min	Typ	Max
-40	-	622	-	-	402	-
-35	-	614	-	-	410	-
-30	-	605	-	-	419	-
-25	-	597	-	-	427	-
-20	-	588	-	-	436	-
-15	-	580	-	-	444	-
-10	-	571	-	-	453	-
-5	-	563	-	-	461	-
0	-	554	-	-	470	-
5	-	546	-	-	478	-
10	-	537	-	-	487	-
15	-	528	-	-	496	-
20	-	520	-	-	504	-
25	-	511	-	-	513	-
30	-	503	-	-	521	-
35	-	494	-	-	530	-
40	-	486	-	-	538	-
45	-	477	-	-	547	-

50	–	469	–	–	555	–
55	–	460	–	–	564	–
60	–	451	–	–	573	–
65	–	443	–	–	581	–
70	–	434	–	–	590	–
75	–	425	–	–	599	–
80	–	417	–	–	607	–
85	–	408	–	–	616	–
90	–	400	–	–	624	–
95	–	391	–	–	633	–
100	–	382	–	–	642	–
105	–	374	–	–	650	–
110	–	365	–	–	659	–
115	–	356	–	–	668	–
120	–	347	–	–	677	–
125	–	339	–	–	685	–

Note:
Code to Temperature mapping of the Temperature sensor is a piece wise linear curve. Any temperature, code faling between to 2 give temperatures can be linearly interpolated.
Code to Temperature mapping should be updated based on sillcon results.

14.4.6 User-Define Mode

- In user-define mode, the PD_DVDD and CHSEL_DVDD are generate by setting register TSADC_USER_CON, bit[3] and bit[2:0]. In order to ensure timing between PD_DVDD and CHSEL_DVDD, the CHSEL_DVDD must be set before the PD_DVDD.
- In user-define mode, you can choose the method to control the START_OF_CONVERSION by setting bit[4] of TSADC_USER_CON. If set to 0, the start_of_conversion will be assert after "inter_pd_soc" cycles, which could be set by bit[11:6] of TSADC_USER_CON. And if start_mode was set 1, the start_of_conversion will be controlled by bit[5] of TSADC_USER_CON.
- Software can get the four channel temperature from TSADC_DATAn (n=0,1,2,3).

14.4.7 Automatic Mode

You can use the automatic mode with the following step:

- Set TSADC_AUTO_PERIOD，configure the interleave between every two accessing of TSADC in normal operation.
- Set TSADC_AUTO_PERIOD_HT. configure the interleave between every two accessing of TSADC after the temperature is higher than COMP_SHUT or COMP_INT.
- Set TSADC_COMPn_INT(n=0,1), configure the high temperature level, if tsadc output is smaller than the value, means the temperature is high, tsadc_int will be asserted.
- Set TSADC_COMPn_SHUT(n=0,1), configure the super high temperature level, if tsadc output is smaller than the value, means the temperature is too high, TSHUT will be asserted.
- Set TSADC_INT_EN, you can enable the high temperature interrupt for all channel; and you can also set TSHUT output to gpio to reset the whole chip; and you can set TSHUT output to cru to reset the whole chip.
- Set TSADC_HIGHT_INT_DEBOUNCE and TSADC_HIGHT_TSHUT_DEBOUNCE, if the temperature is higher than COMP_INT or COMP_SHUT for "debounce" times, TSADC controller will generate interrupt or TSHUT.
- Set TSADC_AUTO_CON, enable the TSADC controller.

Chapter 15 Debug

15.1 Overview

The RK3399Pro uses the Coresight-SOC Technology to support real-time debug access and trace for the multi-core. Software can access debug components and control of debug behavior through a DAP (Debug Access Port). A standard infrastructure is implemented for the capture and transmission of trace data, combination of multiple data streams by funneling together, and then output of data to a trace port. The cross-triggering components are also implemented in RK3399Pro for debug component broadcast events to each other.

15.1.1 Feature

- Access to debug features and on-chip AHB, APB, and JTAG buses through a JTAG or Single Wire Debug (SWD) interface.
- Merging of multiple trace sources into a single trace stream.
- Capture of trace streams on-chip or off-chip.
- Cross-triggering between different debug and trace components.
- Support Timestamp generation.

15.1.2 Debug components address map

The following table shows the debug components address in memory map, the DEBUG system base address is 0xfe400000:

Module	Offset	Size
DAP_ROM	0x000000	4KB
Trace Funnel	0x001000	4KB
CTI_TPIU	0x003000	4KB
Timestamp	0x004000	4KB
TPIU	0x005000	4KB
CLUSTERL_ROM	0x020000	4KB
CLUSTERL_DBG0	0x030000	4KB
CLUSTERL_PMU0	0x031000	4KB
CLUSTERL_DBG1	0x032000	4KB
CLUSTERL_PMU1	0x033000	4KB
CLUSTERL_DBG2	0x034000	4KB
CLUSTERL_PMU2	0x035000	4KB
CLUSTERL_DBG3	0x036000	4KB
CLUSTERL_PMU3	0x037000	4KB
CLUSTERL_CTI0	0x038000	4KB
CLUSTERL_CTI1	0x039000	4KB
CLUSTERL_CTI2	0x03A000	4KB
CLUSTERL_CTI3	0x03B000	4KB
CLUSTERL_ETM0	0x03C000	4KB
CLUSTERL_ETM1	0x03D000	4KB
CLUSTERL_ETM2	0x03E000	4KB
CLUSTERL_ETM3	0x03F000	4KB
CLUSTERB_ROM	0x200000	64KB
CLUSTERB_DBG0	0x210000	64KB
CLUSTERB_CTI0	0x220000	64KB
CLUSTERB_PMU0	0x230000	64KB
CLUSTERB_ETM0	0x240000	64KB
CLUSTERB_DEBUG1	0x310000	64KB
CLUSTERB_CTI1	0x320000	64KB
CLUSTERB_PMU1	0x330000	64KB
CLUSTERB_ETM1	0x340000	64KB

15.2 Block Diagram

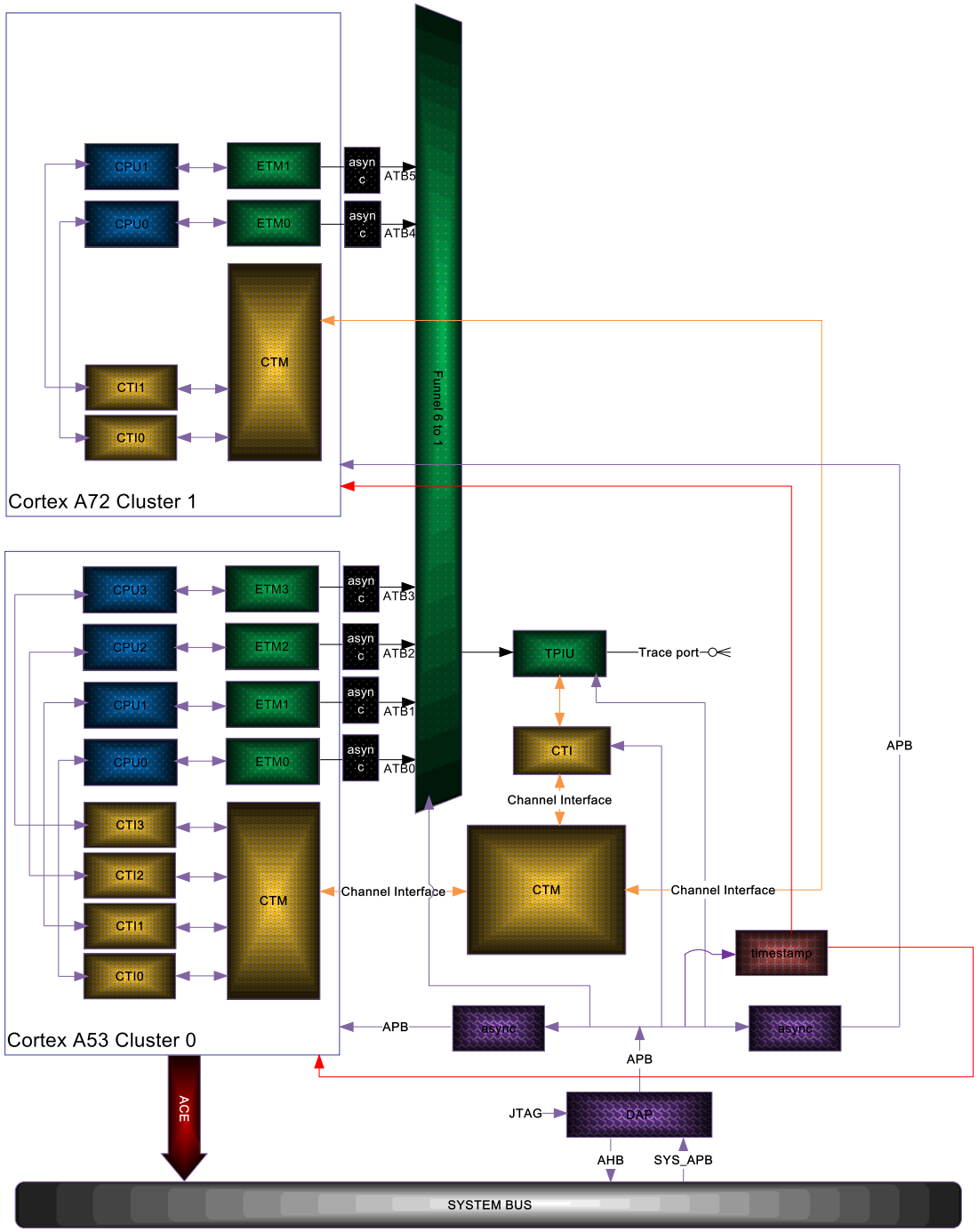


Fig. 15-1 RK3399Pro Debug system structure

The RK3399Pro DEBUG system is designed for a dual-cluster processor system, as shown in figure above. Cluster 0 is a four-core Cortex-A53, and cluster 1 is a dual-core Cortex-A72. The JTAG/SW controller can access all debug system components and system memory through DAP which is a package of APs(access port).

The data flow of trace can be linked together through a “trace funnel component” and be sent to IO by a TPIU component.

ECT (embedded cross trigger) is supported by CTM/CTIs in DEBUG system, CTM/CTIs in the Cortex-A53 cluster and CTM/CTIs in the Cortex-A72 cluster. And a timestamp generator in the DEBUG system can generate a 64bit binary count and be shared and synchronized to the two clusters by asynchronous bridges.

15.3 Function Description

15.3.1 DAP (Debug Access Port)

DAP in RK3399Pro is designed base on ARM coresight400 technology, which do not provide a "universal" DAP component as the previous version. So the real DAP in RK3399Pro is consist of a group of coresight400 components which are integrated by ourselves.
The RK3399Pro DAP architecture is illustrated in figure 6-2, and be described in the following sections.

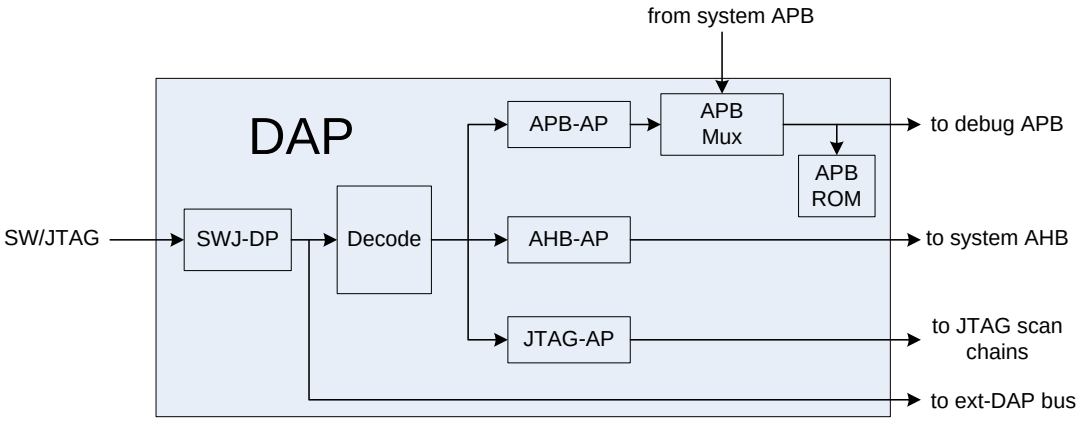


Fig. 15-2 RK3399Pro Debug system DAP structure

- **SWJ-DP:**

The SWJ-DP is a combined JTAG-DP and SW-DP that enables you to connect either an SWD or JTAG probe to a target. It is the standard CoreSight debug port, and enables access either to the JTAG-DP or SW-DP blocks. To make efficient use of package pins, serial wire shares, or overlays, the JTAG pins use an auto-detect mechanism that switches between JTAG-DP and SW-DP depending on which probe is connected. A special sequence on the swdiotms pin switches between JTAG-DP and SW-DP.
The SWJ-DP consists of a wrapper around the JTAG-DP and SW-DP. It selects JTAG or SWD as the connection mechanism and enables either JTAG-DP or SW-DP as the interface to the DAP.
Figure 6-3 shows the structure of the SWJ-DP.

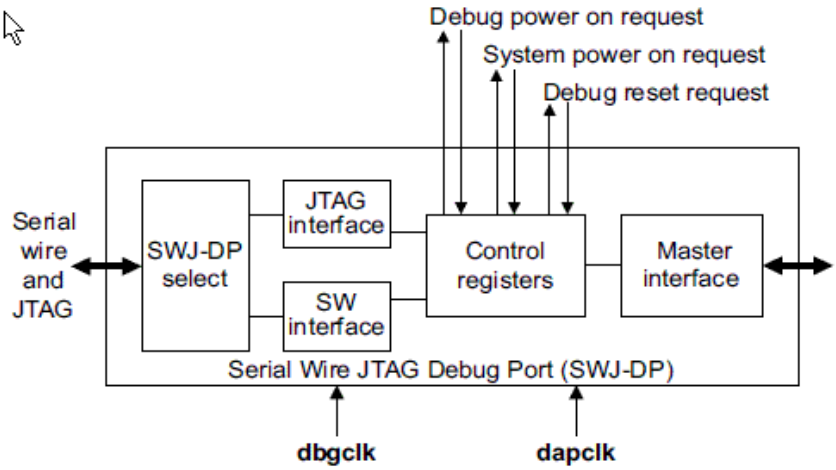


Fig. 15-3 RK3399Pro SWJ-DP structure

When operating as a JTAG-DP this follows the JTAG-DP as defined in the ARM Debug interface Architecture Specification, ADIv5.0 to ADIv5.2. It also contains an explanation of its programmer model, capabilities, and features.
The JTAG-DP contains a debug port state machine that controls the JTAG-DP mode operation, including controlling the scan chain interface that provides the external physical interface to the JTAG-DP. It is based closely on the JTAG TAP State Machine. See IEEE std 1149.1-2001.
When operating as an SW-DP Interface, this implementation is taken from the ARM debug Interface Architecture Specification, ADIv5.0 to ADIv5.2, and operates with a synchronous

serial interface. This uses a single bidirectional data signal and a clock signal.
The SW-DP provides a low pin count, bidirectional serial connection to the DAP with a reference clock signal from synchronous operation.
Communications with the SW-DP use a 3-phase protocol:

- A host-to-target packet request
- A target-to-host acknowledge response
- A data transfer phase, if required. This can be target-to-host or host-to-target, depending on the request made in the first phase.

- **DAPBUSIC:**

The DAPBUS interconnect is a combinational component for connecting the DP to the APs which are AHB-AP and APB-AP in RK3399Pro.
To address a particular AP, the DP uses the eight MSBs of its address bus, dapcaddrs[15:0]. The value driven on these address lines is determined by the APSEL[7:0] field in the AP Select register.

- **AHB-AP:**

The AHB-AP implements the MEM-AP architecture to directly connect to an AHB-based memory system. Connection to other memory systems is possible through suitable bridging.

- **APB-AP:**

The APB-AP implements the MEM-AP architecture to connect directly to an APB based system. This bus is normally dedicated to CoreSight and other debug components.

- **APBIC:**

The APB interconnect connects one or more APB bus masters, for example an APB-AP and an APB interface driven by an on-chip processor. APB interconnects can be cascaded. The RK3399Pro APBIC implements a ROM table at address 0x00000000, which identifies the locations of the CoreSight compoents accessed through it.

15.3.2 ETM (Embedded Trace Macro)

There are eight ETMs in the RK3399Pro along with Cortex-A53 processor and Cortex-A72 processor separately in little and big clusters. The ETM trace unit is a module that performs real-time instruction flow tracing based on the Embedded Trace Macrocell(ETM), architecture ETMv4. ETM is a CoreSight component, and is an integral part of the ARM Real-time Debug solution, DS-5 Development Studio.
The figure 6-4 shows the main function blocks of the ETM trace unit.

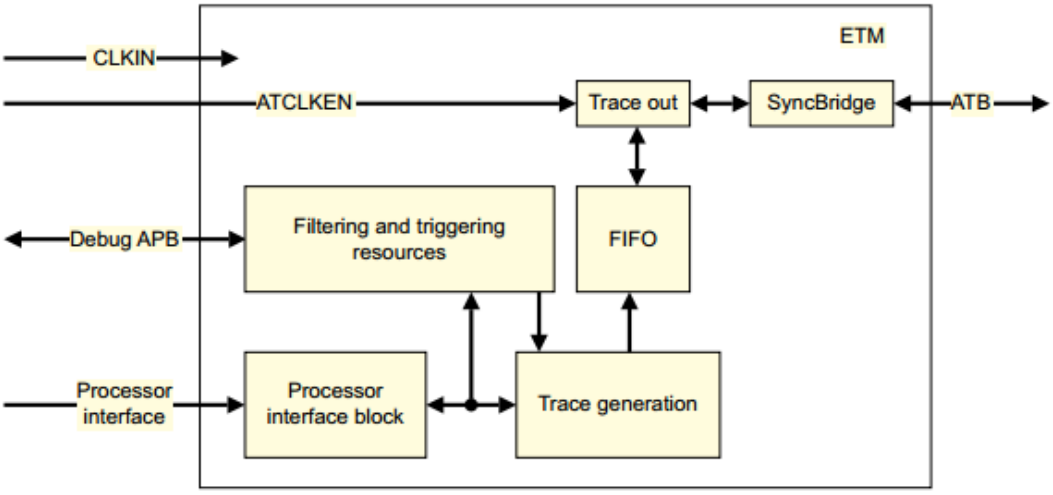


Fig. 15-4 RK3399Pro ETM structure

15.3.3 Trace funnel

The ATB funnel component merges multiple ATB buses into a single ATB buses into a single ATB bus. If the optional APB interface is implemented, a debugger can also control the arbitration scheme and selectively enable the ATB slave interfaces for tracing.
The following figure shows ATB funnel block architecture.

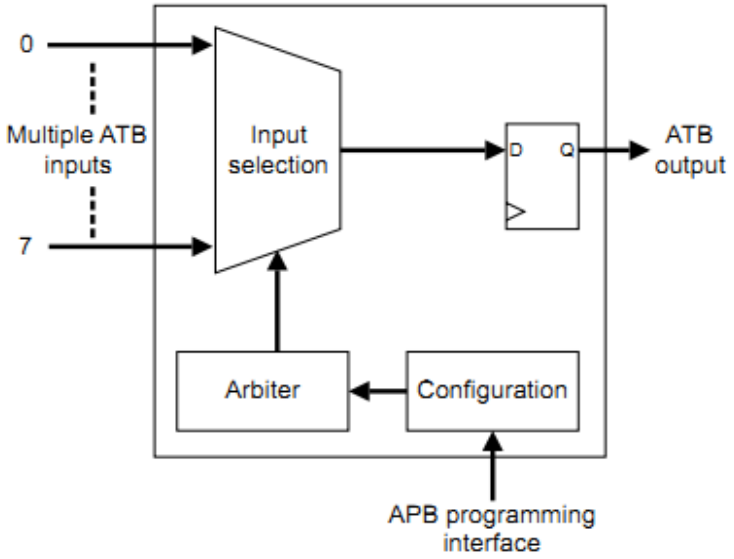


Fig. 15-5 Trace funnel architecture

15.3.4 TPIU

The TPIU acts as a bridge between the on-chip trace data, with separate IDs, to a data stream, encapsulating IDs where required, that is then captured by a Trace Port Analyzer (TPA). Fig. 6-6 shows the main blocks of the TPIU and the clock domains.

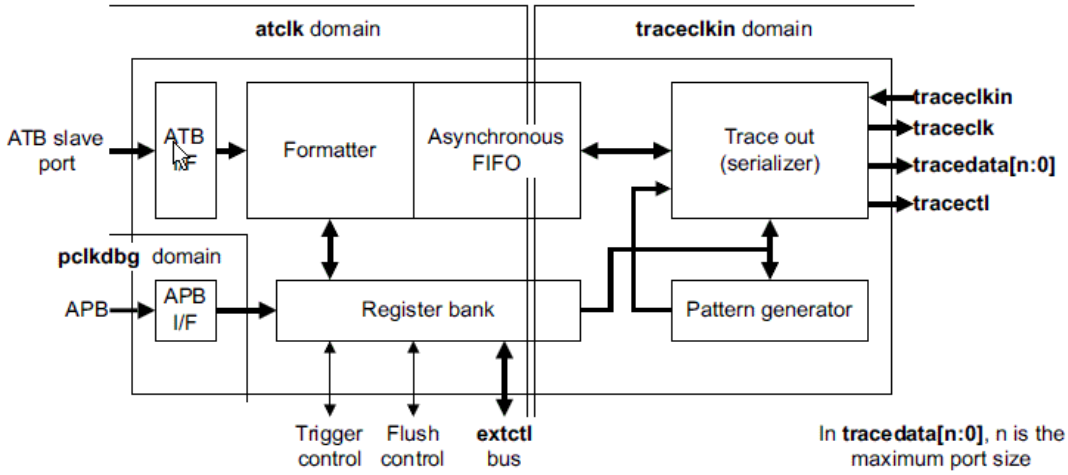


Fig. 15-6 RK3399Pro TPIU structure

The TPIU contains the following components:

- Formatter
  Inserts source ID signals into the data packet stream so that trace data can be re-associated with its trace source. See TPIU formatter and FIFO.
- Asynchronous FIFO
  Enables trace data to be driven out at a speed that is not dependent on the on-chip bus clock.
- Register bank
  Contains the management, control and status registers for triggers, flushing behavior and external control.
- Trace out
  The trace out block serializes formatted data before it goes off-chip.
- Pattern Generator
  The pattern generator unit provides a simple set of defined bit sequences or patterns that can be output over the Trace Port and be detected by the TPA or other associated Trace Capture Device (TCD). The TCD can use these patterns to indicate if it is possible to increase or to decrease the trace port clock speed.
- ATB interface
  The TPIU accepts trace data from a trace source, either direct from a trace source or

using a Trace Funnel.

- APB interface
  The APB interface is the programming interface for the TPIU.

Software must consider the following when programming the TPIU registers for trace capture:

- TPAs that are only capable of operation with tracectl must only use the formatter in either bypass or normal mode, not in continuous mode.
- ARM recommends that following a trigger event within a multi-trace source configuration, a flush is performed to ensure that all historical information related to the trigger is output.
- If flush on trigger event and stop on trigger event options are chosen then any data after the trigger is not captured by the TPA. When the TPIU is instructed to stop, it discards any subsequent trace data, including data returned by the flush. Select Stop on Flush completion instead.
- Although multiple flushes can be scheduled using flush on trigger event, flush on flushin, and manual flush, when one of these requests are made, it masks additional requests of the same type. This means repeated writing to the manual flush bit does not schedule multiple manual requests unless each is permitted to complete first.
- Unless multiple triggers are required, it is not advisable to set both trigger event and trigger on flush completion, if flush on trigger event is also enabled. In addition, if trigger on trigin is enabled with this configuration, it can also cause multiple trigger markers groom on trigger request.

15.3.5 ECT (CTI & CTM)

The ECT for CoreSight consists of a number of CTIs and CTMs connected together. This enables ARM/ETM subsystems to interact. That is cross trigger, with each other. The debug system enables debug support for multiple cores, together with cross triggering between the cores and their respective ETMs.

The main function of the ECT (CTI and CTM) is to pass debug events from one core to another. For example, the ECT can communicate debug state information from one core to another, so that program execution on both processors can be stopped at the same time if required.

- CTI (Cross Trigger Interface)
  The CTI combines and maps the trigger requests, and broadcasts them to all other interfaces on the ECT as channel events. When the CTI receives a channel event it maps this onto a trigger output. This enables subsystems to cross trigger with each other. The receiving and transmitting of triggers is performed through the trigger interface.
- CTM (Cross Trigger Matrix)
  This block controls the distribution of channel events. It provides Channel Interfaces (CIs) for connection to either CTIs or CTMs. This enables multiple CTIs to be linked together.

15.3.6 Timestamp

The timestamp components generate and distribute a consistent time value to multiple processors in RK3399Pro.

The timestamp system is shown in the following figure:

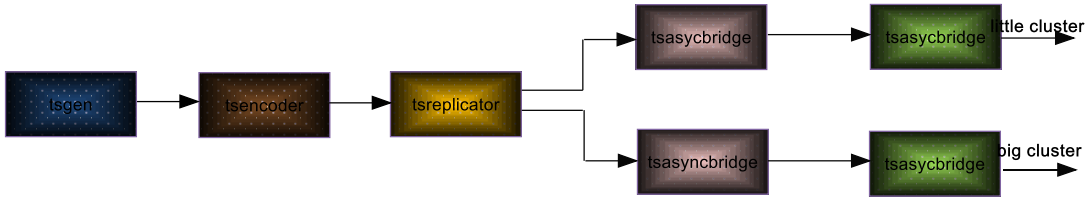


Fig. 15-7 RK3399Pro Timestamp structure

The timestamp interconnect provides a mechanism for efficiently distributing a timestamp value across a potentially large system in a way that is cost-effective to implement. It has the following features:

- Uses a master timing reference with a fixed frequency of typically 10-50 MHz

- Time always counts forward
- Time available as a natural binary number to software
- Writeable and readable count value
- Distributed synchronization of timestamp
- Time value presented as a 64-bit binary count.

The interconnect ensures that any components that uses the distributed timestamp are synchronized to the distributed count value with minimal skew while the timestamp interconnect is clocked.

The CoreSight timestamp generator can be used in one of the following ways:

- To generate the time reported by ARM processors that implements the Generic Timer specification. Software expects that this time does not count backwards, and so it is important that only secure software can change the timestamp value. The programmer's model of the timestamp generator has been designed to enable non-secure software to read the timestamp value while only permitting secure software to change the timestamp value.
- To generate the time used to align traces and other debug information in the CoreSight system. The timestamp generator is controlled by debug software and connected to the debug APB interconnects. The read-only interface is not used.

15.4 Register Description

For details of DEBUG system components (such as DAP, FUNNEL, TPIU, TIMESTAMP and CTI) registers, please reference ARM document "DDI0480F_soc_r3p1_trm" chapter 3.
For details of debug components inside Cortex-A53 cluster (such as ETM and CA53 CTI), please reference ARM document "DDI0500C_cortex_a53_r0p4_trm" chapter 11. And for Cortex-A72 cluster, please reference ARM document "ARM_Cortex-A72_MPCore_Technical_Reference_Manual_r0p1-00eac0" chapter 12, 13.

15.5 Interface description

15.5.1 DAP SWJ-DP interface

The following figure is the DAP SWJ-DP interface, the SWJ-DP is a combined JTAG-DP and SW-DP that enable you connect either a Serial Write Debug(SWJ) to JTAG probe to a target.

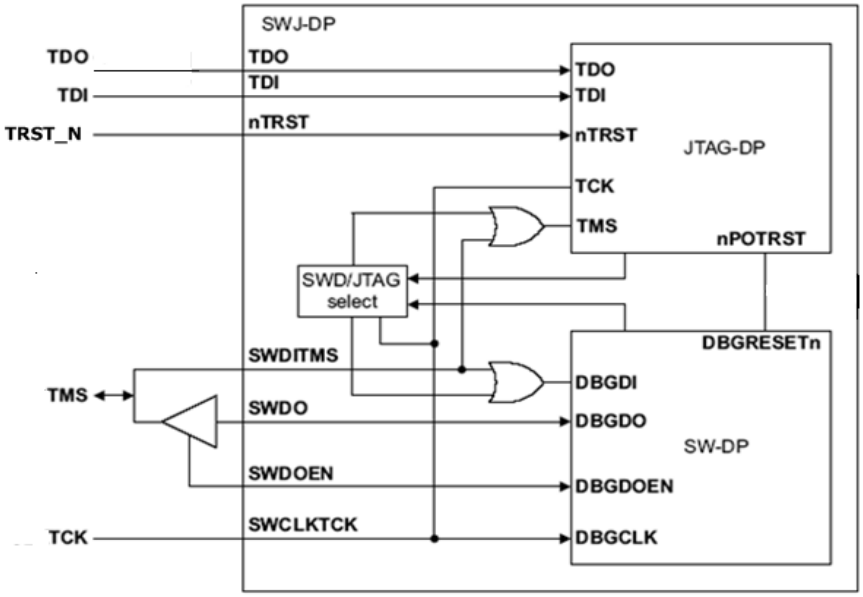


Fig. 15-8 DAP SWJ interface

Table 15-1 SWJ interface

Module Pin	Direction	PAD Name	IOMUX Setting
TCK	I	IO_SDMMCdata2_CXCSJTAGtck_ HDCPJTAGtdi_SDMMCgpio4b2	grf_gpio4b_sel[5:4]= 0x2
TMS	IO	IO_SDMMCdata3_CXCSJTAGtms_ HDCPJTAGtdo_SDMMCgpio4b3	grf_gpio4b_sel[7:6]= 0x2

15.5.2 TPIU trace port interface

Table 15-2 TPIU interface

Module Pin	Direction	PAD Name	IOMUX Setting
trace_data[0]	O	IO_I2S0sclk_TRACEdata0_A72CORE0wfi_AUDIOgpio3d0	grf_gpio3d_sel[1:0] = 0x2
trace_data[1]	O	IO_I2S0lrckrx_TRACEdata1_A72CORE1wfi_AUDIOgpio3d1	grf_gpio3d_sel[3:2] = 0x2
trace_data[2]	O	IO_I2S0lrcktx_TRACEdata2_A53CORE0wfi_AUDIOgpio3d2	grf_gpio3d_sel[5:4] = 0x2
trace_data[3]	O	IO_I2S0sdi0_TRACEdata3_A53CORE1wfi_AUDIOgpio3d3	grf_gpio3d_sel[7:6] = 0x2
trace_data[4]	O	IO_I2S0sdi1sdo3_TRACEdata4_A53CORE2wfi_AUDIOgpio3d4	grf_gpio3d_sel[9:8] = 0x2
trace_data[5]	O	IO_I2S0sdi2sdo2_TRACEdata5_A53CORE3wfi_AUDIOgpio3d5	grf_gpio3d_sel[11:10] = 0x2
trace_data[6]	O	IO_I2S0sdi3sdo1_TRACEdata6_A72L2wfi_AUDIOgpio3d6	grf_gpio3d_sel[13:12] = 0x2
trace_data[7]	O	IO_I2S0sdo0_TRACEdata7_A53L2wfi_AUDIOgpio3d7	grf_gpio3d_sel[15:14] = 0x2
trace_data[8]	O	IO_I2C1AUDIOCAMscl_TRACEdata8_AUDIOgpio4a2	grf_gpio4a_sel[5:4] = 0x2
trace_data[9]	O	IO_I2S1sclk_TRACEdata9_AUDIOgpio4a3	grf_gpio4a_sel[7:6] = 0x2
trace_data[10]	O	IO_I2S1lrckrx_TRACEdata10_AUDIOgpio4a4	grf_gpio4a_sel[9:8] = 0x2
trace_data[11]	O	IO_I2S1lrcktx_TRACEdata11_AUDIOgpio4a5	grf_gpio4a_sel[11:10] = 0x2
trace_data[12]	O	IO_MACtxd2_SPI4EXPrxd_TRACEdata12_GMACgpio3a0	grf_gpio3a_sel[1:0] = 0x3
trace_data[13]	O	IO_MACtxd3_SPI4EXPtxd_TRACEdata13_GMACgpio3a1	grf_gpio3a_sel[3:2] = 0x3
trace_data[14]	O	IO_MACrxd2_SPI4EXPclk_TRACEdata14_GMACgpio3a2	grf_gpio3a_sel[5:4] = 0x3
trace_data[15]	O	IO_MACrxd3_SPI4EXPcsn0_TRACEdata15_GMACgpio3a3	grf_gpio3a_sel[7:6] = 0x3
trace_clk	O	IO_I2C1AUDIOCAMsda_TRACEclk_AUDIOgpio4a1	grf_gpio4a_sel[3:2] = 0x2
trace_ctl	O	IO_I2Sclk_TRACEctl_LPM0wfi_AUDIOgpio4a0	grf_gpio4a_sel[1:0] = 0x2

Chapter 16 Mailbox

16.1 Overview

The Mailbox module is a simple APB peripheral that allows both the Cortex-A53/Cortex-A72 and Cortex-M0 system to communicate by writing operation to generate interrupt. The registers are accessible by both CPU via APB interface.

The Mailbox has the following main features:

- Support dual-core system: Cortex-A53/Cortex-A72 and Cortex-M0
- Support APB interface
- Support four mailbox elements, each element includes one data word, one command word register and one flag bit that can represent one interrupt
- Four interrupts to Cortex-A53/Cortex-A72
- Four interrupts to Cortex-M0
- Provide 32 lock registers for software to use to indicate whether mailbox is occupied

16.2 Block Diagram

The figure below shows Mailbox block diagram:

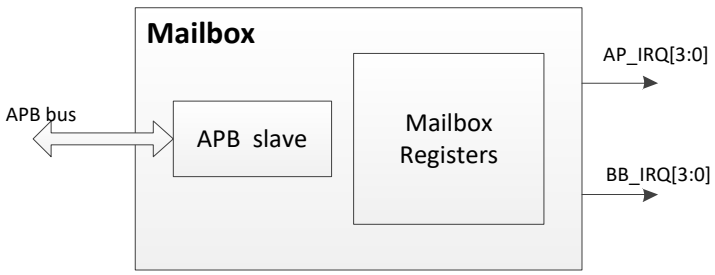


Fig. 16-1 Mailbox Block Diagram

16.3 Function Description

16.3.1 Interrupt to Cortex-A53/ Cortex-A72

The interrupt to Cortex-A53/ Cortex-A72 (CA53/72_IRQ[i] , i=0~3) is generated when B2A_INTEN[i] equals to 1 and there are writing operation to B2A_CMD_i and B2A_DATA_i orderly.(i=0~3)

The interrupt to Cortex-A53/ Cortex-A72 (CA53/72_IRQ[i], i=0~3) is cleared when writing 1 to B2A_STATUS[i]. (i=0~3)

16.3.2 Interrupt to Cortex-M0

The interrupt to Cortex-M0 (MCU_IRQ[i]) is generated when A2B_INTEN[i] equals to 1 and there are writing operation to A2B_CMD_i and A2B_DATA_i orderly.(i=0~3)

The interrupt to Cortex-M0 (MCU_IRQ[i],i=0~3) is cleared when writing 1 to A2B_STATUS[i]. (i=0~3).

There are two Cortex-M0s (perilp Cortex-M0 and PMU Cortex-M0), so there are two independent mail box(mailbox0 for PERILP and mailbox1 for PMU) for each.

16.4 Register Description

16.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
MAILBOX_A2B_INTEN	0x00000	W	0x00000000	Cortex-A53/Cortex-A72 to Cortex-M0 interrupt enable register

Name	Offset	Size	Reset Value	Description
MAILBOX_A2B_STATUS	0x00004	W	0x00000000	Cortex-A53/Cortex-A72 to Cortex-M0 interrupt status register
MAILBOX_A2B_CMD_0	0x00008	W	0x00000000	Cortex-A53/Cortex-A72 to Cortex-M0 command 0
MAILBOX_A2B_DAT_0	0x0000c	W	0x00000000	Cortex-A53/Cortex-A72 to Cortex-M0 data 0
MAILBOX_A2B_CMD_1	0x00010	W	0x00000000	Cortex-A53/Cortex-A72 to Cortex-M0 command 1
MAILBOX_A2B_DAT_1	0x00014	W	0x00000000	Cortex-A53/Cortex-A72 to Cortex-M0 data 1
MAILBOX_A2B_CMD_2	0x00018	W	0x00000000	Cortex-A53/Cortex-A72 to Cortex-M0 command 2
MAILBOX_A2B_DAT_2	0x0001c	W	0x00000000	Cortex-A53/Cortex-A72 to Cortex-M0 data 2
MAILBOX_A2B_CMD_3	0x00020	W	0x00000000	Cortex-A53/Cortex-A72 to Cortex-M0 command 3
MAILBOX_A2B_DAT_3	0x00024	W	0x00000000	Cortex-A53/Cortex-A72 to Cortex-M0 data 3
MAILBOX_B2A_INTEN	0x00028	W	0x00000000	Cortex-A53/Cortex-A72 to Cortex-M0 interrupt enable register
MAILBOX_B2A_STATUS	0x0002c	W	0x00000000	Cortex-M0 to Cortex-A53/Cortex-A72 interrupt status register
MAILBOX_B2A_CMD_0	0x00030	W	0x00000000	Cortex-M0 to Cortex-A53/Cortex-A72 command 0
MAILBOX_B2A_DAT_0	0x00034	W	0x00000000	Cortex-M0 to Cortex-A53/Cortex-A72 data 0
MAILBOX_B2A_CMD_1	0x00038	W	0x00000000	Cortex-M0 to Cortex-A53/Cortex-A72 command 1
MAILBOX_B2A_DAT_1	0x0003c	W	0x00000000	Cortex-M0 to Cortex-A53/Cortex-A72 data 1
MAILBOX_B2A_CMD_2	0x00040	W	0x00000000	Cortex-M0 to Cortex-A53/Cortex-A72 command 2
MAILBOX_B2A_DAT_2	0x00044	W	0x00000000	Cortex-M0 to Cortex-A53/Cortex-A72 data 2
MAILBOX_B2A_CMD_3	0x00048	W	0x00000000	Cortex-M0 to Cortex-A53/Cortex-A72 command 3
MAILBOX_B2A_DAT_3	0x0004c	W	0x00000000	Cortex-M0 to Cortex-A53/Cortex-A72 data 3
MAILBOX_ATOMIC_LOCK_00	0x00100	W	0x00000000	Lock flag register 00
MAILBOX_ATOMIC_LOCK_01	0x00104	W	0x00000000	Lock flag register 01
MAILBOX_ATOMIC_LOCK_02	0x00108	W	0x00000000	Lock flag register 02

Name	Offset	Size	Reset Value	Description
MAILBOX_ATOMIC_LOCK_03	0x0010c	W	0x00000000	Lock flag register 03
MAILBOX_ATOMIC_LOCK_04	0x00110	W	0x00000000	Lock flag register 04
MAILBOX_ATOMIC_LOCK_05	0x00114	W	0x00000000	Lock flag register 05
MAILBOX_ATOMIC_LOCK_06	0x00118	W	0x00000000	Lock flag register 06
MAILBOX_ATOMIC_LOCK_07	0x0011c	W	0x00000000	Lock flag register 07
MAILBOX_ATOMIC_LOCK_08	0x00120	W	0x00000000	Lock flag register 08
MAILBOX_ATOMIC_LOCK_09	0x00124	W	0x00000000	Lock flag register 09
MAILBOX_ATOMIC_LOCK_10	0x00128	W	0x00000000	Lock flag register 10
MAILBOX_ATOMIC_LOCK_11	0x0012c	W	0x00000000	Lock flag register 11
MAILBOX_ATOMIC_LOCK_12	0x00130	W	0x00000000	Lock flag register 12
MAILBOX_ATOMIC_LOCK_13	0x00134	W	0x00000000	Lock flag register 13
MAILBOX_ATOMIC_LOCK_14	0x00138	W	0x00000000	Lock flag register 14
MAILBOX_ATOMIC_LOCK_15	0x0013c	W	0x00000000	Lock flag register 15
MAILBOX_ATOMIC_LOCK_16	0x00140	W	0x00000000	Lock flag register 16
MAILBOX_ATOMIC_LOCK_17	0x00144	W	0x00000000	Lock flag register 17
MAILBOX_ATOMIC_LOCK_18	0x00148	W	0x00000000	Lock flag register 18
MAILBOX_ATOMIC_LOCK_19	0x0014c	W	0x00000000	Lock flag register 19
MAILBOX_ATOMIC_LOCK_20	0x00150	W	0x00000000	Lock flag register 20
MAILBOX_ATOMIC_LOCK_21	0x00154	W	0x00000000	Lock flag register 21
MAILBOX_ATOMIC_LOCK_22	0x00158	W	0x00000000	Lock flag register 22
MAILBOX_ATOMIC_LOCK_23	0x0015c	W	0x00000000	Lock flag register 23
MAILBOX_ATOMIC_LOCK_24	0x00160	W	0x00000000	Lock flag register 24

Name	Offset	Size	Reset Value	Description
MAILBOX_ATOMIC_LOCK_25	0x00164	W	0x00000000	Lock flag register 25
MAILBOX_ATOMIC_LOCK_26	0x00168	W	0x00000000	Lock flag register 26
MAILBOX_ATOMIC_LOCK_27	0x0016c	W	0x00000000	Lock flag register 27
MAILBOX_ATOMIC_LOCK_28	0x00170	W	0x00000000	Lock flag register 28
MAILBOX_ATOMIC_LOCK_29	0x00174	W	0x00000000	Lock flag register 29
MAILBOX_ATOMIC_LOCK_30	0x00178	W	0x00000000	Lock flag register 30
MAILBOX_ATOMIC_LOCK_31	0x0017c	W	0x00000000	Lock flag register 31

*Notes:Size:**B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access*

16.4.2 Detail Register Description

MAILBOX_A2B_INTEN

Address: Operational Base + offset (0x00000)

Cortex-A53/Cortex-A72 to MCU interrupt enable register

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3	RW	0x0	int3 interrupt enable for int3
2	RW	0x0	int2 interrupt enable for int2
1	RW	0x0	int1 interrupt enable for int1
0	RW	0x0	int0 interrupt enable for int0

MAILBOX_A2B_STATUS

Address: Operational Base + offset (0x00004)

Cortex-A53/Cortex-A72 to MCU interrupt status register

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3	RW	0x0	int3 interrupt status for int3. when writte 1, int is cleared.
2	RW	0x0	int2 interrupt status for int2. when writte 1, int is cleared.

Bit	Attr	Reset Value	Description
1	RW	0x0	int1 interrupt status for int1. when writte 1, int is cleared.
0	RW	0x0	int0 interrupt status for int0. when writte 1, int is cleared.

MAILBOX_A2B_CMD_0
Address: Operational Base + offset (0x00008)
Cortex-A53/Cortex-A72 to MCU command 0

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	command command of Cortex-A53/Cortex-A72 to Cortex-M0

MAILBOX_A2B_DAT_0
Address: Operational Base + offset (0x0000c)
Cortex-A53/Cortex-A72 to MCU data 0

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	data data of Cortex-A53/Cortex-A72 to Cortex-M0

MAILBOX_A2B_CMD_1
Address: Operational Base + offset (0x00010)
Cortex-A53/Cortex-A72 to MCU command 1

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	command command of Cortex-A53/Cortex-A72 to Cortex-M0

MAILBOX_A2B_DAT_1
Address: Operational Base + offset (0x00014)
Cortex-A53/Cortex-A72 to MCU data 1

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	data data of Cortex-A53/Cortex-A72 to Cortex-M0

MAILBOX_A2B_CMD_2
Address: Operational Base + offset (0x00018)
Cortex-A53/Cortex-A72 to MCU command 2

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	command command of Cortex-A53/Cortex-A72 to Cortex-M0

MAILBOX_A2B_DAT_2
Address: Operational Base + offset (0x0001c)
Cortex-A53/Cortex-A72 to MCU data 2

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	data data of Cortex-A53/Cortex-A72 to Cortex-M0

MAILBOX_A2B_CMD_3
Address: Operational Base + offset (0x00020)
Cortex-A53/Cortex-A72 to MCU command 3

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	command command of Cortex-A53/Cortex-A72 to Cortex-M0

MAILBOX_A2B_DAT_3
Address: Operational Base + offset (0x00024)
Cortex-A53/Cortex-A72 to MCU data 3

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	data data of Cortex-A53/Cortex-A72 to Cortex-M0

MAILBOX_B2A_INTEN
Address: Operational Base + offset (0x00028)
Cortex-A53/Cortex-A72 to MCU interrupt enable register

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3	RW	0x0	int3 interrupt enable for int3
2	RW	0x0	int2 interrupt enable for int2
1	RW	0x0	int1 interrupt enable for int1
0	RW	0x0	int0 interrupt enable for int0

MAILBOX_B2A_STATUS
Address: Operational Base + offset (0x0002c)
MCU to Cortex-A53/Cortex-A72 interrupt status register

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3	RW	0x0	int3 interrupt status for int3. when writte 1, int is cleared.

Bit	Attr	Reset Value	Description
2	RW	0x0	int2 interrupt status for int2. when writte 1, int is cleared.
1	RW	0x0	int1 interrupt status for int1. when writte 1, int is cleared.
0	RW	0x0	int0 interrupt status for int0. when writte 1, int is cleared.

MAILBOX_B2A_CMD_0
Address: Operational Base + offset (0x00030)
MCU to Cortex-A53/Cortex-A72 command 0

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	command command of MCU to Cortex-A53/Cortex-A72

MAILBOX_B2A_DAT_0
Address: Operational Base + offset (0x00034)
MCU to Cortex-A53/Cortex-A72 data 0

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	data data of MCU to Cortex-A53/Cortex-A72

MAILBOX_B2A_CMD_1
Address: Operational Base + offset (0x00038)
MCU to Cortex-A53/Cortex-A72 command 1

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	command command of MCU to Cortex-A53/Cortex-A72

MAILBOX_B2A_DAT_1
Address: Operational Base + offset (0x0003c)
MCU to Cortex-A53/Cortex-A72 data 1

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	data data of MCU to Cortex-A53/Cortex-A72

MAILBOX_B2A_CMD_2
Address: Operational Base + offset (0x00040)
MCU to Cortex-A53/Cortex-A72 command 2

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	command command of Cortex-M0 to Cortex-A53/Cortex-A72

MAILBOX_B2A_DAT_2
Address: Operational Base + offset (0x00044)
MCU to Cortex-A53/Cortex-A72 data 2

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	data data of Cortex-M0 to Cortex-A53/Cortex-A72

MAILBOX_B2A_CMD_3
Address: Operational Base + offset (0x00048)
MCU to Cortex-A53/Cortex-A72 command 3

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	command command of Cortex-M0 to Cortex-A53/Cortex-A72

MAILBOX_B2A_DAT_3
Address: Operational Base + offset (0x0004c)
MCU to Cortex-A53/Cortex-A72 data 3

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	data data of Cortex-M0 to Cortex-A53/Cortex-A72

MAILBOX_ATOMIC_LOCK_00
Address: Operational Base + offset (0x00100)
Lock flag register 00

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	atomic_lock_00 lock flag bit 00

MAILBOX_ATOMIC_LOCK_01
Address: Operational Base + offset (0x00104)
Lock flag register 01

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	atomic_lock_01 lock flag bit 01

MAILBOX_ATOMIC_LOCK_02
Address: Operational Base + offset (0x00108)
Lock flag register 02

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved

Bit	Attr	Reset Value	Description
0	RW	0x0	atomic_lock_02 lock flag bit 02

MAILBOX_ATOMIC_LOCK_03
Address: Operational Base + offset (0x0010c)
Lock flag register 03

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	atomic_lock_03 lock flag bit 03

MAILBOX_ATOMIC_LOCK_04
Address: Operational Base + offset (0x00110)
Lock flag register 04

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	atomic_lock_04 lock flag bit 04

MAILBOX_ATOMIC_LOCK_05
Address: Operational Base + offset (0x00114)
Lock flag register 05

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	atomic_lock_05 lock flag bit 05

MAILBOX_ATOMIC_LOCK_06
Address: Operational Base + offset (0x00118)
Lock flag register 06

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	atomic_lock_06 lock flag bit 06

MAILBOX_ATOMIC_LOCK_07
Address: Operational Base + offset (0x0011c)
Lock flag register 07

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	atomic_lock_07 lock flag bit 07

MAILBOX_ATOMIC_LOCK_08

Address: Operational Base + offset (0x00120)
Lock flag register 08

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	atomic_lock_08 lock flag bit 08

MAILBOX_ATOMIC_LOCK_09
Address: Operational Base + offset (0x00124)
Lock flag register 09

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	atomic_lock_09 lock flag bit 09

MAILBOX_ATOMIC_LOCK_10
Address: Operational Base + offset (0x00128)
Lock flag register 10

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	atomic_lock_10 lock flag bit 10

MAILBOX_ATOMIC_LOCK_11
Address: Operational Base + offset (0x0012c)
Lock flag register 11

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	atomic_lock_11 lock flag bit 11

MAILBOX_ATOMIC_LOCK_12
Address: Operational Base + offset (0x00130)
Lock flag register 12

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	atomic_lock_12 lock flag bit 12

MAILBOX_ATOMIC_LOCK_13
Address: Operational Base + offset (0x00134)
Lock flag register 13

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	atomic_lock_13 lock flag bit 13

MAILBOX_ATOMIC_LOCK_14
Address: Operational Base + offset (0x00138)
Lock flag register 14

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	atomic_lock_14 lock flag bit 14

MAILBOX_ATOMIC_LOCK_15
Address: Operational Base + offset (0x0013c)
Lock flag register 15

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	atomic_lock_15 lock flag bit 15

MAILBOX_ATOMIC_LOCK_16
Address: Operational Base + offset (0x00140)
Lock flag register 16

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	atomic_lock_16 lock flag bit 16

MAILBOX_ATOMIC_LOCK_17
Address: Operational Base + offset (0x00144)
Lock flag register 17

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	atomic_lock_17 lock flag bit 17

MAILBOX_ATOMIC_LOCK_18
Address: Operational Base + offset (0x00148)
Lock flag register 18

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	atomic_lock_18 lock flag bit 18

MAILBOX_ATOMIC_LOCK_19
Address: Operational Base + offset (0x0014c)
Lock flag register 19

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	atomic_lock_19 lock flag bit 19

MAILBOX_ATOMIC_LOCK_20
Address: Operational Base + offset (0x00150)
Lock flag register 20

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	atomic_lock_20 lock flag bit 20

MAILBOX_ATOMIC_LOCK_21
Address: Operational Base + offset (0x00154)
Lock flag register 21

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	atomic_lock_21 lock flag bit 21

MAILBOX_ATOMIC_LOCK_22
Address: Operational Base + offset (0x00158)
Lock flag register 22

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	atomic_lock_22 lock flag bit 22

MAILBOX_ATOMIC_LOCK_23
Address: Operational Base + offset (0x0015c)
Lock flag register 23

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	atomic_lock_23 lock flag bit 23

MAILBOX_ATOMIC_LOCK_24
Address: Operational Base + offset (0x00160)
Lock flag register 24

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	atomic_lock_24 lock flag bit 24

MAILBOX_ATOMIC_LOCK_25
Address: Operational Base + offset (0x00164)
Lock flag register 25

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	atomic_lock_25 lock flag bit 25

MAILBOX_ATOMIC_LOCK_26
Address: Operational Base + offset (0x00168)
Lock flag register 26

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	atomic_lock_26 lock flag bit 26

MAILBOX_ATOMIC_LOCK_27
Address: Operational Base + offset (0x0016c)
Lock flag register 27

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	atomic_lock_27 lock flag bit 27

MAILBOX_ATOMIC_LOCK_28
Address: Operational Base + offset (0x00170)
Lock flag register 28

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	atomic_lock_28 lock flag bit 28

MAILBOX_ATOMIC_LOCK_29
Address: Operational Base + offset (0x00174)
Lock flag register 29

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	atomic_lock_29 lock flag bit 29

MAILBOX_ATOMIC_LOCK_30
Address: Operational Base + offset (0x00178)
Lock flag register 30

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved

Bit	Attr	Reset Value	Description
0	RW	0x0	atomic_lock_30 lock flag bit 30

MAILBOX_ATOMIC_LOCK_31
Address: Operational Base + offset (0x0017c)
Lock flag register 31

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	atomic_lock_31 lock flag bit 31

16.5 Application Notes

- The order for writing MAILBOX_B2A_CMD_i/ MAILBOX_A2B_CMD_i and MAILBOX_B2A_DATA_i/ MAILBOX_A2B_DATA_i registers is limited: MAILBOX_B2A_CMD_i/ MAILBOX_A2B_CMD_i firstly, then MAILBOX_B2A_DATA_i/ MAILBOX_A2B_DATA_i. If wrong order is used, then the interrupt cannot be generated successfully.
- If you want to clear the interrupt, you can read out the STATUS register and writing 1 to corresponding bit.
- When using mailbox, software should read MAILBOX_ATOMIC_LOCK_i first. That the reading value is 0 means that it is available, and 1 means it has been automatically locked. Writing MAILBOX_ATOMIC_LOCK_i will clear this bit.

Chapter 17 eFuse

17.1 Overview

This device supports two eFuse. Both are organized as 32bits by 32 one-time programmable electrical fuses. eFuse0 is non-secure efuse(NSeFuse), and eFuse1 is secure efuse(SeFuse).
eFuse0 can be accessed by APB bus at secure mode and non-secure mode. eFuse1 can only be accessed by APB bus at secure mode. It is a type of non-volatile memory fabricated in standard CMOS logic process. The main features are as follows:

- Working condition : VDD = 0.9±10%
- Programming condition : VQPS = 1.8V~1.98V
- Program time : 12us±1us
- Read condition : VQPS = 0V
- Embedded four redundancy bits
- Provide power-down and standby mode

17.2 Block Diagram

In the following diagram, all the signals except power supply VDD and VQPS are controlled by registers. For detailed description, please refer to detailed register descriptions.

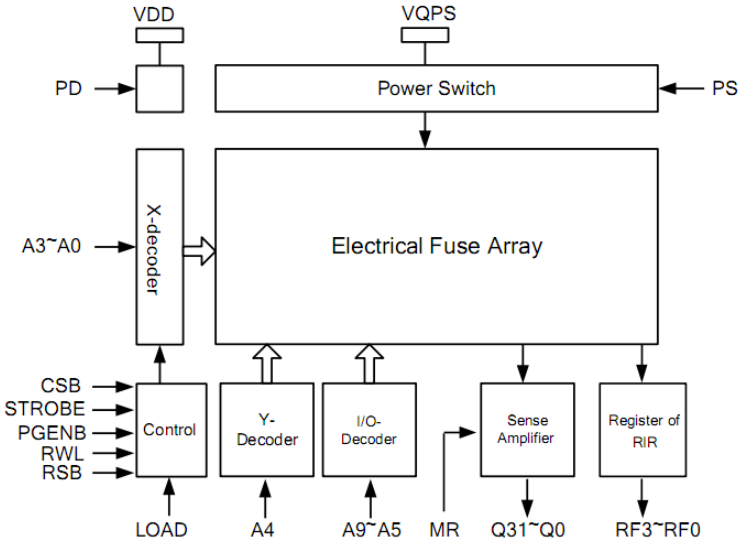


Fig. 17-1 eFuse block diagram

17.3 Function Description

This electrical fuse macro has 8 operation modes: array program(A_PGM), redundancy program (R_PGM), array read (A_READ), redundancy read (R_READ), power-down, standby mode and 2 test modes: margin A_READ1 mode and margin R_READ1 mode.

Array Program Mode (A_PGM)

In this mode, this macro is ready for electrical fuse programming. Any bit in this macro can be programmed in any order by raising STROBE high with a proper address selected. The selected address needs to satisfy setup and hold time with respect to STROBE to be valid. Only one bit is programmed at a time.

Array Read Mode (A_READ); Margin A_READ1 Mode

In these 2 modes, this macro is ready to read data from fuse cells. 32 bits Q31~Q0 can be read out by raising STROBE high with a proper address selected. During this read operation, address signals A9~A5 are "don"t care". The read trip point of Array Read Mode is lower than that of Margin A_READ1 Mode. If RSB is at "L" and enable redundancy function, please MUST enter the redundancy read mode (R_READ or Margin R_READ1) and read the RIR data once prior to the array read mode (A_READ or Margin A_READ1) after power-up even if repairing is not needed. Redundancy read requires two strobe cycles to

read out completed repairing information.The data will be stored in registers and remain there until power-down or power-off. In subsequent array read, when read access the failure bit in the main array, the corresponding output data will be corrected automatically.

Redundancy Program Mode (R_PGM)

In this mode, this macro is ready for electrical fuse programming on redundancy bits. Any bit within the redundancy array can be programmed in any order by raising STROBE high with a proper address selected. The selected address needs to satisfy setup and hold time with respect to STROBE to be valid. Only one bit is programmed at a time.

Redundancy Read Mode (R_READ); Margin R_READ1 Mode

In these 2 modes, this macro is ready to read data from redundancy information row to register, Q31~Q0 and RF3~RF0 by two cycle STROBE high. During redundancy read operation, address signals A9~A0 are "don"t care" except A4. The read trip point of Redundancy READ Mode is lower than that of Margin R_READ1 Mode.

Power-down Mode

In this mode, the macro is at power-down mode. In the power-down mode, the power-leakage has best performance and consumes the least current of VDD.

Standby Mode

When PD=L, PS=L, CSB=H, STROBE, PGENB and LOAD, MR, RWL, RSB are "don"t care", the macro is at standby mode.

17.4 Register Description

This section describes the control/status registers of the design.

17.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
EFUSE_CTRL	0x0000	W	0x00000000	efuse control register
EFUSE_DOUT	0x0004	W	0x00000000	efuse data out register
EFUSE_RF	0x0008	W	0x00000000	efuse redundancy bit used indicator register
EFUSE_JTAG_PASS	0x0010	W	0x0cf7680a	Jtag password
EFUSE_STROBE_FINISH_CTRL	0x0014	W	0x00009003	efuse strobe finish control register

*Notes: Size: **B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**- WORD (32 bits) access*

17.4.2 Detail Register Description

EFUSE_CTRL

Address: Operational Base + offset (0x0000)

eFuse control register

Bit	Attr	Reset Value	Description
31:26	RO	0x00	reserved
25:16	RW	0x00	efuse_addr efuse address pins :A[9:0]
15:10	RO	0x00	reserved
9	RW	0x0	efuse_strobe_sft_sel efuse strobe control software mode select, active high
8	RW	0x0	efuse_rwl efuse redundancy information row select (active high) : RWL
7	RW	0x1	efuse_rsb efuse redundancy enable(active low) : RSB
6	RW	0x0	efuse_mr efuse read trip point setting, MR = L for normal read mode; MR = H for margin read1 mode : MR

Bit	Attr	Reset Value	Description
5	RW	0x1	efuse_pd efuse power down enable (active high) : PD
4	RW	0x0	efuse_ps efuse pass 1.8V program voltage to internal for program(active high) : PS
3	RW	0x1	efuse_pgenb efuse program enable (active low) : PGENB
2	RW	0x0	efuse_load efuse turn on sense amplifier and load data into latch (active high) : LOAD
1	RW	0x0	efuse_strobe efuse turn on the array for read or program access (active high) : STROBE
0	RW	0x1	efuse_csb efuse chip select enable signal, active low : CSB

EFUSE_DOUT
Address: Operational Base + offset (0x0004)
eFuse data out register

Bit	Attr	Reset Value	Description
31:0	RO	0x00	efuse_dout eFuse data output

EFUSE_RF
Address: Operational Base + offset (0x0008)
efuse redundancy bit used indicator register

Bit	Attr	Reset Value	Description
31:4	RO	0x00	reserved
3:0	RO	0x0	efuse_rf_r efuse redundancy bit used indicator register for RF3~RF0, Output high once the redundancy bit has been used.

EFUSE_JTAG_PASSWD
Address: Operational Base + offset (0x0010)
eFuse jtag passwd register

Bit	Attr	Reset Value	Description
31:0	RW	0xcf7680a	Jtag_passwd Jtag password for jtag monitor

EFUSE_STROBE_FINISH_CON
Address: Operational Base + offset (0x0014)
eFuse jtag passwd register

Bit	Attr	Reset Value	Description
31:16	RO	0x00	reserved

Bit	Attr	Reset Value	Description
15:8	RW	0x90	efuse_strobe_finish_prg efuse read strobe finish control in hardware mode.
7:0	RW	0x03	efuse_strobe_finish_read efuse program strobe finish control in hardware mode.

17.5 Timing Diagram

- **When eFuse is in A_PGM mode**

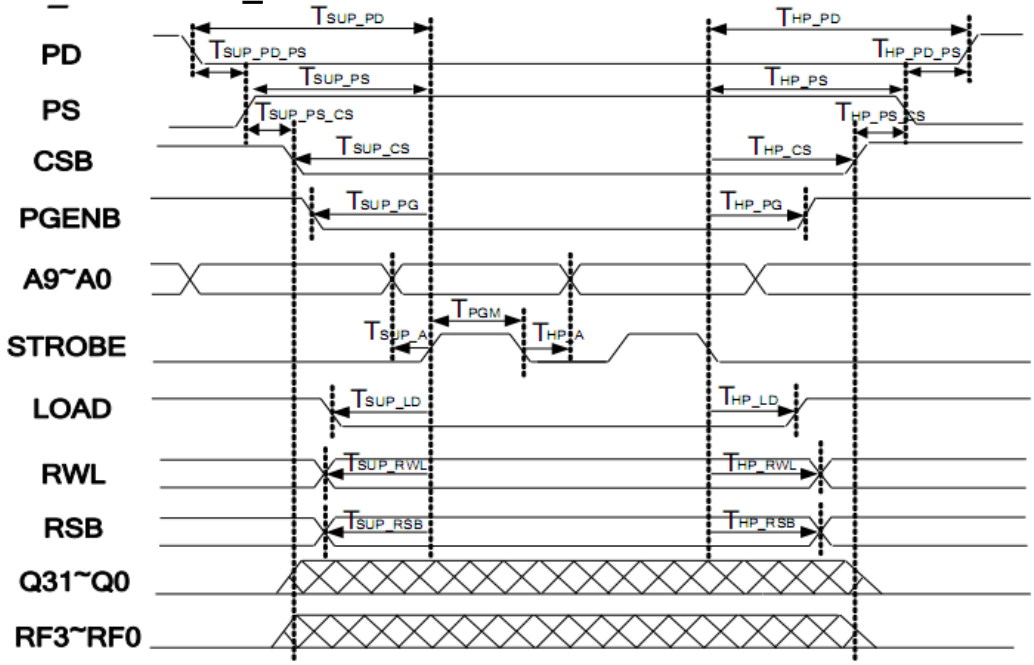


Fig. 17-2 eFuse timing diagram A_PGM mode

- **When eFuse is in R_PGM mode**

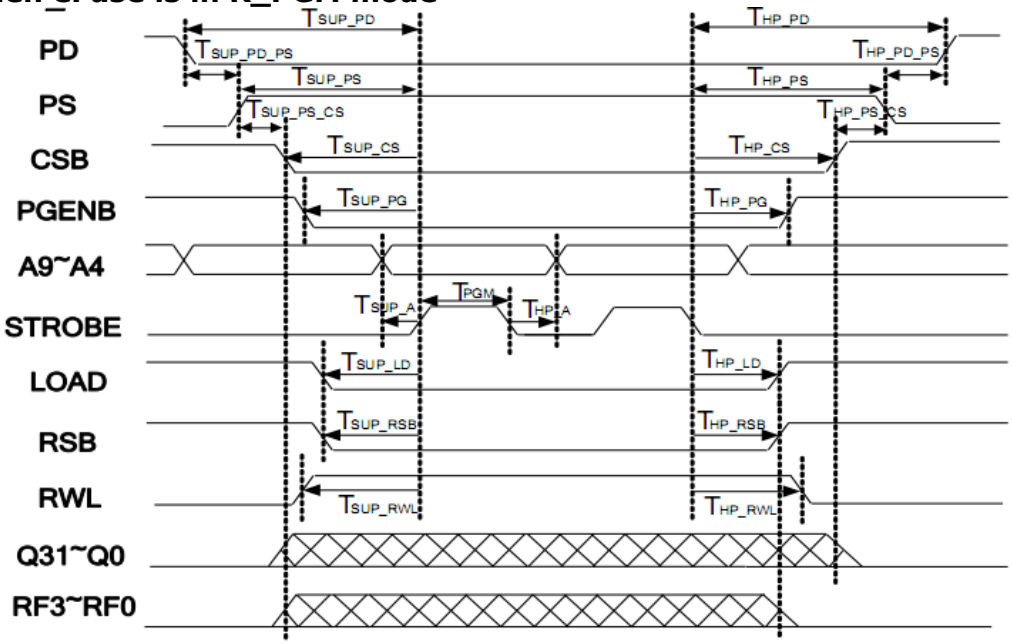


Fig. 17-3 eFuse timing diagram R_PGM mode

Table 17-1 Timing Requirements for Program Mode

Parameter	Symbol	Min.	Typ.	Max.	Unit
PD to PS setup time	$T_{SUP\_PD\_PS}$		796.549		ns
PD to PS hold time	$T_{HP\_PD\_PS}$		8.17		ns

Parameter	Symbol	Min.	Typ.	Max.	Unit
PS to CSB setup time	$T_{SUP\_PS\_CS}$		45.549		ns
PS to CSB hold time	$T_{HP\_PS\_CS}$		45.17		ns
PD to STROBE setup time	T_{SUP_PD}		846.368		ns
PD to STROBE hold time	T_{HP_PD}		57.394		ns
PS to STROBE setup	T_{SUP_PS}		49.819		ns
PS to STROBE hold time	T_{HP_PS}		49.224		ns
RWL to STROBE setup time	T_{SUP_RWL}		12.397		ns
RWL to STROBE hold time	T_{HP_RWL}		11.91		ns
RSB to STROBE setup time	T_{SUP_RSB}		12.897		ns
RSB to STROBE hold time	T_{HP_RSB}		12.41		ns
CSB to STROBE setup time	T_{SUP_CS}		4.29		ns
CSB to STROBE hold time	T_{HP_CS}		4.056		ns
PGENB to STROBE setup time	T_{SUP_PG}		3.846		ns
PGENB to STROBE hold time	T_{HP_PG}		3.872		ns
Typical program strobe pulse width	T_{PGM}		12		us
A7~A0 to STROBE setup time	T_{SUP_A}		11.797		ns
A7~A0 to STROBE hold time	T_{HP_A}		11.31		ns
LOAD to STROBE setup time	T_{SUP_LD}		3.929		ns
LOAD to STROBE hold time	T_{HP_LD}		3.787		ns

- **When efuse is in A_READ mode; Margin A_READ1 Mode.**

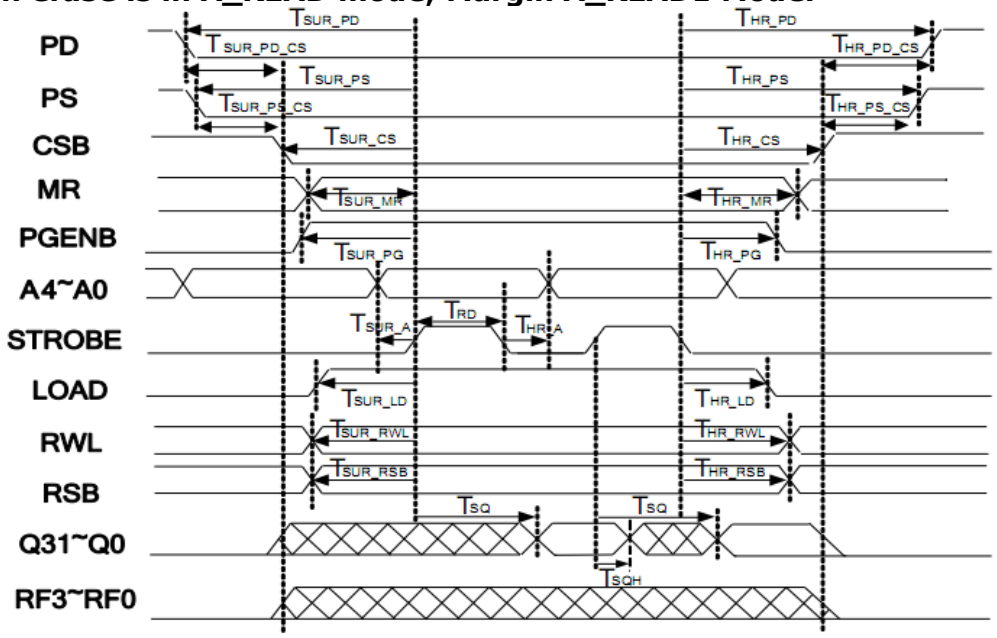


Fig. 17-4 eFuse timing diagram in A_READ mode and Margin A_READ1 Mode

- **When efuse is in R_READ mode; Margin R_READ1 Mode.**

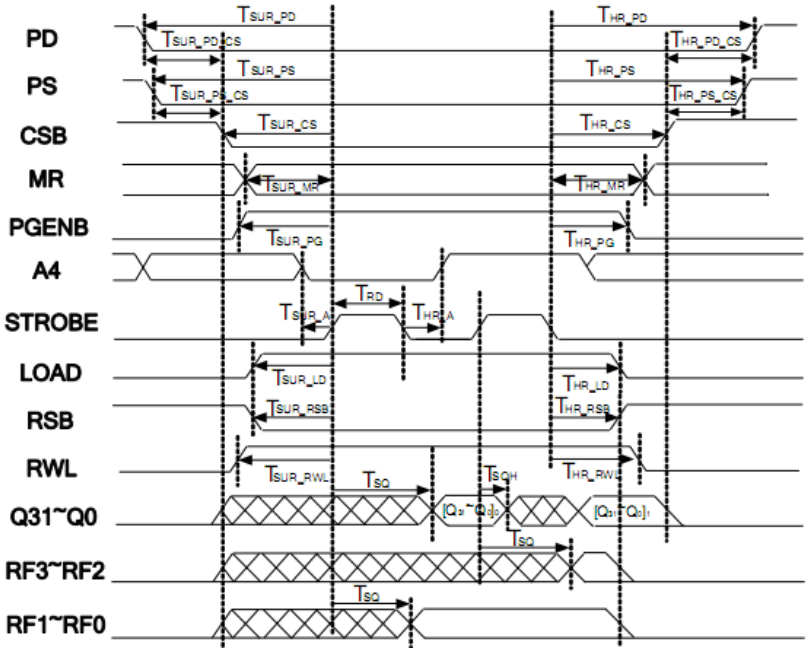


Fig. 17-5 eFuse timing diagram in R_READ mode and Margin R_READ1 Mode

Table 17-2 Timing Requirements for Read Mode

Parameter	Symbol	Min.	Typ.	Max.	Unit
PD to CSB setup time	$T_{SUR\_PD\_CS}$		796.531		ns
PD to CSB hold time	$T_{HR\_PD\_CS}$		8.168		ns
PS to CSB setup time	$T_{SUR\_PS\_CS}$		45.531		ns
PS to CSB hold time in	$T_{HR\_PS\_CS}$		47.168		ns
PD to STROBE setup time	T_{SUR_PD}		800.781		ns
PD to STROBE hold time	T_{HR_PD}		12.22		ns
PS to STROBE setup time	T_{SUR_PS}		49.781		ns
PS to STROBE hold time	T_{HR_PS}		51.22		ns
RWL to STROBE setup time	T_{SUR_RWL}		12.397		ns
RWL to STROBE hold time	T_{HR_RWL}		11.91		ns
RSB to STROBE setup time	T_{SUR_RSB}		12.897		ns
RSB to STROBE hold time	T_{HR_RSB}		12.41		ns
CSB to STROBE setup time	T_{SUR_CS}		4.25		ns
CSB to STROBE hold time	T_{HR_CS}		4.052		ns
PGENB to STROBE setup time	T_{SUR_PG}		3.844		ns
PGENB to STROBE hold time	T_{HR_PG}		3.872		ns
Read strobe pulse width	T_{RD}		110		ns
A2~A0 to STROBE setup time	T_{SUR_A}		11.797		ns
A2~A0 to STROBE hold time	T_{HR_A}		11.31		ns
LOAD to STROBE setup time	T_{SUR_LD}		3.918		ns
LOAD to STROBE hold time	T_{HR_LD}		3.782		ns
MR to STROBE setup time	T_{SUR_MR}		10.097		ns
MR to STROBE hold time	T_{HR_MR}		9.61		ns
Q31~Q0 access time from STROBE high	T_{SQ}		109.2		ns
Q31~Q0 hold time to the next STROBE	T_{SQH}		0		ns

17.6 Application Notes

During usage of efuse, customers must pay more attention to the following items:
1. In condition of program(PGM) mode, VQPS= 1.8V~1.98V.
2. Q0~Q7/Q31 will be reset to "0" once CSB at high.
3. No data access allowed at the rising edge of CSB.
4. All the program timing for each signal must be more than the value defined in the timing table.
5. It must use Margin A_READ1 Mode when read out the eFuse data to determine whether program successfully.
6.If enable redundancy function, please MUST enter the redundancy read mode (R_READ or Margin R_READ1) and read the RIR data once prior to the array read mode (A_READ or Margin A_READ1) after power-up even if repairing is not needed. Redundancy read requires two strobe cycles to read out the complete repairing information. The data will be stored in registers and will remain there until power-down or power-off. In subsequent array read, when read access the failure bit in the main array, the corresponding output data will be corrected automatically.

17.6.1 eFuse Macro Operating Modes

Follow tapble is eFuse macro operation mode truth table

Table 17-3 eFuse macro operation mode truth table

Mode	CSB	STROBE	LOAD	PGENB	PS	PD	MR	RSB	RWL	VQPS
A_READ mode	L	H	H	H	L	L	L	L	L	0V
	L	H	H	H	L	L	L	H	X	0V
A_PGM mode	L	H	L	L	H	L	X	L	L	1.8V~1.98V
	L	H	L	L	H	L	X	H	X	1.8V~1.98V
R_READ mode	L	H	H	H	L	L	L	L	H	0V
R_PGM mode	L	H	L	L	H	L	X	L	H	1.8V~1.98V
Standby mode	H	X	X	X	L	L	X	X	X	0V
Power-down mode	H	X	X	X	L	H	X	X	X	0V
Margin A_READ1	L	H	H	H	L	L	H	L	L	0V
	L	H	H	H	L	L	H	H	X	0V
Margin R_READ1 mode	L	H	H	H	L	L	H	L	H	0V

- Maximum accumulative time when VQPS=1.8V~1.98V, PS=H should be less than 0.2 sec.
- VQPS MUST NOT exceed 1.98V (1.8V+10%) when PS=H for device reliability concern.
- Please always keep PS at "L" except program mode.
- Max accumulative read access time should be less than 2 seconds per each bit when CSB=L, LOAD=H, PGENB=H, STROBE=H.
- Max accumulative numbers of read MUST be less than 2 million reads per each bit when CSB=L, LOAD=H, PGENB=H, STROBE=H.
- PS=H, PD=H state is not allowed to avoid unintended program.
- RSB signal (active low) is used to disable/enable redundancy feature (repair function).

17.6.2 eFuse STOBOE control

eFuse STROBE has hardware and software control mode, the default mode is hardware control.

- In hardware control mode, the STROBE is asserted at the rising edge of efuse_strobe and de-asserted by the internal circuit accordding to the timing requirements. You can configure the efuse_strobe_finish_read and efuse_strobe_finish_prg to adjust the active time of STROBE. The default active time of STROBE is 250ns in read mode and 12us in program mode. At default setting, internal circuit require efuse_strobe to remain valid longer than 500 ns in read mode and longer than 12.5 us in program mode. In addition, due the exist of internal circuit control, the Tsq (efuse_strobe -> Q) increase 250 ns.
- You can configure the efuse_strobe_sft_sel to select software control mode, the STROBE is directly controlled by efuse_strobe. You must configure the STROBE according to the timing requirements.

17.6.3 Read Mode

For read mode address signals A[9]~A[5] are "invalid"

Table 17-4 eFuse Dout Format

A[4]-~A[0]	D[0]	D[1]	…	…	D[30]	D[31]
00000	Fuse[0]	Fuse[32]	…	…	Fuse[960]	Fuse[992]
00001	Fuse[1]	Fuse[33]	…	…	Fuse[961]	Fuse[993]
…	…	…	…	…	…	…
…	…	…	…	…	…	…
11110	Fuse[30]	Fuse[62]	…	…	Fuse[990]	Fuse[1022]
11111	Fuse[31]	Fuse[63]	…	…	Fuse[991]	Fuse[1023]

17.6.4 Read & Program Masked For eFuse1

The eFuse1 was masked by the program masked bits in program mode and read masked bits in read mode.The masked bits comes from Secure GRF registers SGRF_EFUSE_PRG_MASK and SGRF_EFUSE_READ_MASK. The registers only can write one time. Only when the masked bit is set to "1", the program and read is valid.

The Masked bit only 32bits, so one bit controlled 32 eFuse bit, showed as the following:

READ_MASK /PGR_MASK	A[4]-~A[0]	D[0]	D[1]	…	…	D[30]	D[31]
[0]	00000	Fuse[0]	Fuse[32]	…	…	Fuse[960]	Fuse[992]
[1]	00001	Fuse[1]	Fuse[33]	…	…	Fuse[961]	Fuse[993]
…	…	…	…	…	…	…	…
…	…	…	…	…	…	…	…
[30]	11110	Fuse[30]	Fuse[62]	…	…	Fuse[990]	Fuse[1022]
[31]	11111	Fuse[31]	Fuse[63]	…	…	Fuse[991]	Fuse[1023]

Chapter 18 Watchdog (WDT)

18.1 Overview

Watchdog Timer (WDT) is an APB slave peripheral that can be used to prevent system lockup that caused by conflicting parts or programs in a SOC. The WDT would generate interrupt or reset signal when its counter reaches zero, then a reset controller would reset the system.

WDT supports the following features:

- 32 bits APB bus width
- WDT counter's clock is pclk
- 32 bits WDT counter width
- Counter counts down from a preset value to 0 to indicate the occurrence of a timeout
- WDT can perform two types of operations when timeout occurs:
  - Generate a system reset
  - First generate an interrupt and if this is not cleared by the service routine by the time a second timeout occurs then generate a system reset
- Programmable reset pulse length
- Total 16 defined-ranges of main timeout period
- There are two watchdogs in ALIVE named WDT0 and WDT1, and one watchdog in PMU named WDT2. WDT0 can drive CRU to generate global software reset.

18.2 Block Diagram

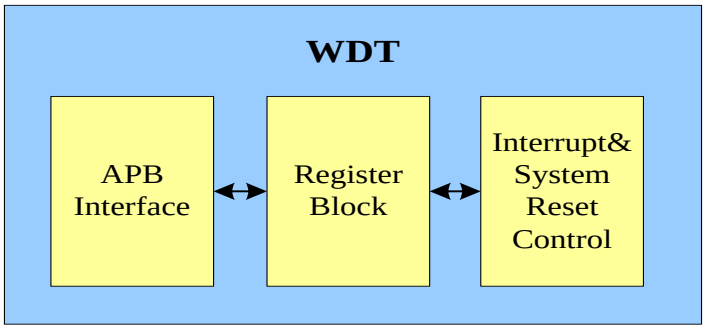


Fig. 18-1 WDT block diagram

Block Descriptions:

- APB Interface

The APB Interface implements the APB slave operation. Its data bus width is 32 bits.

- Register Block

A register block that read coherence for the current count register.

- Interrupt & system reset control

An interrupt/system reset generation block is comprised of a decrementing counter and control logic.

18.3 Function Description

18.3.1 Operation

Counter

The WDT counts from a preset (timeout) value in descending order to zero. When the counter reaches zero, depending on the output response mode selected, either a system reset or an interrupt occurs. When the counter reaches zero, it wraps to the selected timeout value and continues decrementing. The user can restart the counter to its initial value. This is programmed by writing to the restart register at any time. The process of restarting the watchdog counter is sometimes referred as kicking the dog. As a safety feature to prevent accidental restarts, the value 0x76 must be written to the Current Counter Value Register (WDT_CRR).

Interrupts

The WDT can be programmed to generate an interrupt (and then a system reset) when a timeout occurs. When a 1 is written to the response mode field (RMOD, bit 1) of the Watchdog Timer Control Register (WDT_CR), the WDT generates an interrupt. If it is not cleared by the time a second timeout occurs, then it generates a system reset. If a restart occurs at the same time the watchdog counter reaches zero, an interrupt is not generated.

System Resets

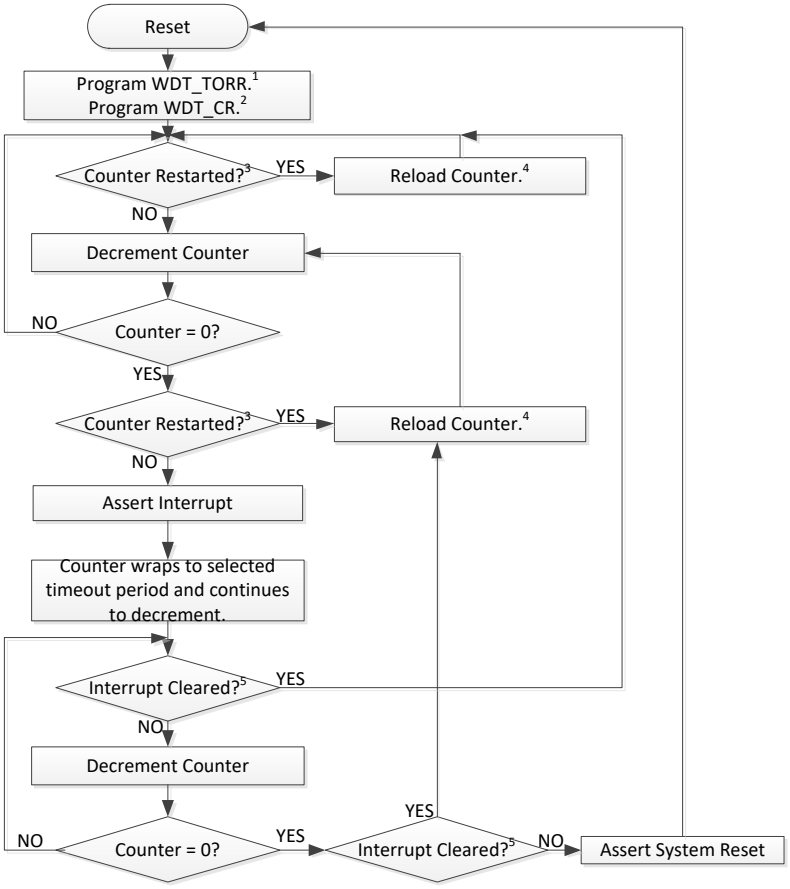
When a 0 is written to the output response mode field (RMOD, bit 1) of the Watchdog Timer Control Register (WDT_CR), the WDT generates a system reset when a timeout occurs.

Reset Pulse Length

The reset pulse length is the number of pclk cycles for which a system reset is asserted. When a system reset is generated, it remains asserted for the number of cycles specified by the reset pulse length or until the system is reset. A counter restart has no effect on the system reset once it has been asserted.

18.3.2 Programming sequence

Operation Flow Chart (Response mode=1)



1. Select required timeout period.
2. Set reset pulse length, response mode, and enable WDT.
3. Write 0x76 to WDT_CRR.
4. Starts back to selected timeout period.
5. Can clear by reading WDT_EOI or restarting (kicking) the counter by writing 0x76 to WDT_CRR.

Fig. 18-2 WDT Operation Flow

18.4 Register Description

This section describes the control/status registers of the design.

18.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
WDT_CR	0x0000	W	0x0000000a	Control Register

Name	Offset	Size	Reset Value	Description
WDT_TORR	0x0004	W	0x00000000	Timeout range Register
WDT_CCVR	0x0008	W	0x00000000	Current counter value Register
WDT_CRR	0x000c	W	0x00000000	Counter restart Register
WDT_STAT	0x0010	W	0x00000000	Interrupt status Register
WDT_EOI	0x0014	W	0x00000000	Interrupt clear Register

*Notes:Size:**B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access*

18.4.2 Detail Register Description

WDT_CR

Address: Operational Base + offset (0x0000)
Control Register

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4:2	RW	0x2	rst_pluse_lenth Reset pulse length. This is used to select the number of pclk cycles for which the system reset stays asserted. 000: 2 pclk cycles 001: 4 pclk cycles 010: 8 pclk cycles 011: 16 pclk cycles 100: 32 pclk cycles 101: 64 pclk cycles 110: 128 pclk cycles 111: 256 pclk cycles
1	RW	0x1	resp_mode Response mode. Selects the output response generated to a timeout. 0: Generate a system reset. 1: First generate an interrupt and if it is not cleared by the time a second timeout occurs then generate a system reset.
0	RW	0x0	wdt_en Writable when the configuration parameter WDT_ALWAYS_EN=0, otherwise, it is readable. This bit is used to enable and disable the watchdog. When disabled, the counter dose not decrement .Thus, no interrupt or system reset is generated. Once this bit has been enabled, it can be cleared only by a system reset. 0: WDT disabled; 1: WDT enabled.

WDT_TORR

Address: Operational Base + offset (0x0004)
Timeout range Register

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3:0	RW	0x0	timeout_period Timeout period. This field is used to select the timeout period from which the watchdog counter restarts. A change of the timeout period takes effect only after the next counter restart (kick). The range of values available for a 32-bit watchdog counter are: 0000: 0x0000ffff 0001: 0x0001ffff 0010: 0x0003ffff 0011: 0x0007ffff 0100: 0x000fffff 0101: 0x001fffff 0110: 0x003fffff 0111: 0x007fffff 1000: 0x00ffffff 1001: 0x01ffffff 1010: 0x03ffffff 1011: 0x07ffffff 1100: 0x0fffffff 1101: 0x1fffffff 1110: 0x3fffffff 1111: 0x7fffffff

WDT_CCVR
Address: Operational Base + offset (0x0008)
Current counter value Register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	cur_cnt Current counter value This register, when read, is the current value of the internal counter. This value is read coherently whenever it is read

WDT_CRR
Address: Operational Base + offset (0x000c)
Counter restart Register

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	W1C	0x00	cnt_restart Counter restart This register is used to restart the WDT counter. As a safety feature to prevent accidental restarts, the value 0x76 must be written. A restart also clears the WDT interrupt. Reading this register returns zero.

WDT_STAT
Address: Operational Base + offset (0x0010)
Interrupt status Register

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RO	0x0	wdt_status This register shows the interrupt status of the WDT. 1: Interrupt is active regardless of polarity; 0: Interrupt is inactive.

WDT_EOI
Address: Operational Base + offset (0x0014)
Interrupt clear Register

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RC	0x0	wdt_int_clr Clears the watchdog interrupt. This can be used to clear the interrupt without restarting the watchdog counter.

Chapter 19 Pulse Width Modulation (PWM)

19

19.1 Overview

The pulse-width modulator (PWM) feature is very common in embedded systems. It provides a way to generate a pulse periodic waveform for motor control or can act as a digital-to-analog converter with some external components.
The PWM Module supports the following features:

- 4-built-in PWM channels
- Configurable to operate in capture mode
  - Measures the high/low polarity effective cycles of this input waveform
  - Generates a single interrupt at the transition of input waveform polarity
  - 32-bit high polarity capture register
  - 32-bit low polarity capture register
  - 32-bit current value register
  - The capture result of channel 3 can be stored in a FIFO. The depth of FIFO is 8, and the data in FIFO can be read through DMA. It also supports timeout interrupt when the data in FIFO has not been read in a time threshold.
- Configurable to operate in continuous mode or one-shot mode
  - 32-bit period counter
  - 32-bit duty register
  - 32-bit current value register
  - Configurable PWM output polarity in inactive state and duty period pulse polarity
  - Period and duty cycle are shadow buffered. Change takes effect when the end of the effective period is reached or when the channel is disabled
  - Programmable center or left aligned outputs, and change takes effect when the end of the effective period is reached or when the channel is disabled
  - 8-bit repeat counter for one-shot operation. One-shot operation will produce N + 1 periods of the waveform, where N is the repeat counter value, and generates a single interrupt at the end of operation
  - Continuous mode generates the waveform continuously, and does not generates any interrupts
- pre-scaled operation to bus clock and then further scaled
- Available low-power mode to reduce power consumption when the channel is inactive.

19.2 Block Diagram

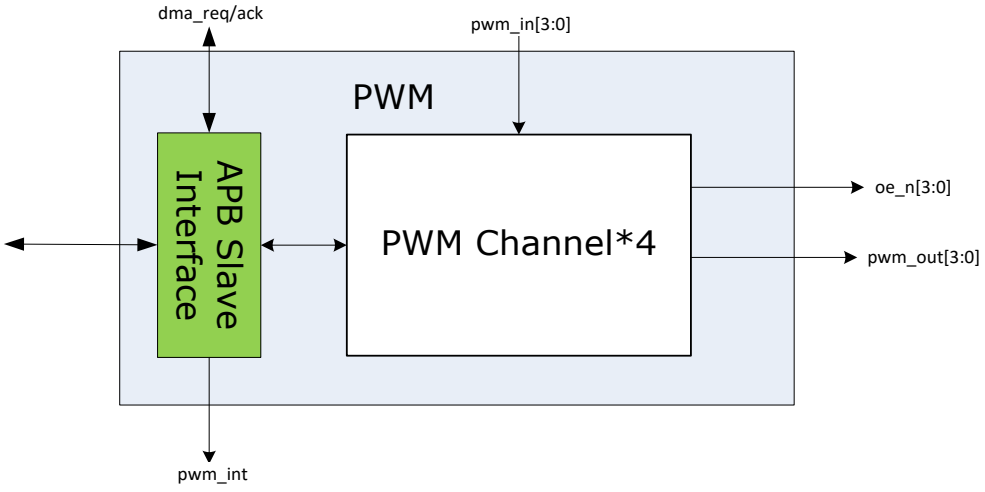


Fig. 19-1 PWM Block Diagram

The host processor gets access to PWM Register Block through the APB slave interface with 32-bit bus width, and asserts the active-high level interrupt. PWM only supports one interrupt output, please refer to interrupt register to know the raw interrupt status when an interrupt is asserted.

PWM Channel is the control logic of PWM module, and controls the operation of PWM module according to the configured working mode.

19.3 Function Description

The PWM supports three operation modes: capture mode, one-shot mode and continuous mode. For the one-shot mode and the continuous mode, the PWM output can be configured as the left-aligned mode or the center-aligned mode.

19.3.1 Capture mode

The capture mode is used to measure the PWM channel input waveform high/low effective cycles with the PWM channel clock, and asserts an interrupt when the polarity of the input waveform changes. The number of the high effective cycles is recorded in the PWMx_PERIOD_HPC register, while the number of the low effective cycles is recorded in the PWMx_DUTY_LPC register.

Notes: the PWM input waveform is doubled buffered when the PWM channel is working in order to filter unexpected shot-time polarity transition, and therefore the interrupt is asserted several cycles after the input waveform polarity changes, and so does the change of the values of PWMx_PERIOD_HPC and PWMx_DUTY_LPC.

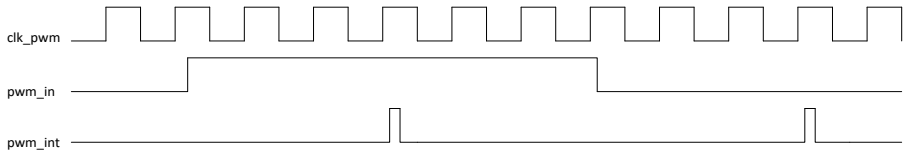


Fig. 19-2 PWM Capture Mode

19.3.2 Continuous mode

The PWM channel generates a series of the pulses continuously as expected once the channel is enabled with continuous mode.

In the continuous mode, the PWM output waveforms can be in one form of the two output mode: left-aligned mode or center-aligned mode.

For the left-aligned output mode, the PWM channel firstly starts the duty cycle with the configured duty polarity (PWMx_CTRL.duty_pol). Once duty cycle number (PWMx_DUTY_LPC) is reached, the output is switched to the opposite polarity. After the period number (PWMx_PERIOD_HPC) is reached, the output is again switched to the opposite polarity to start another period of desired pulse.

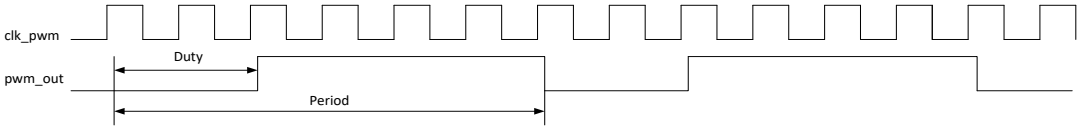


Fig. 19-3 PWM Continuous Left-aligned Output Mode

For the center-aligned output mode, the PWM channel firstly starts the duty cycle with the configured duty polarity (PWMx_CTRL.duty_pol). Once one half of duty cycle number (PWMx_DUTY_LPC) is reached, the output is switched to the opposite polarity. Then if there is one half of duty cycle left for the whole period, the output is again switched to the opposite polarity. Finally after the period number (PWMx_PERIOD_HPC) is reached, the output starts another period of desired pulse.

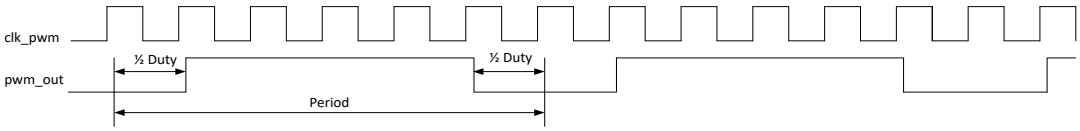


Fig. 19-4 PWM Continuous Center-aligned Output Mode

Once disable the PWM channel, the channel stops generating the output waveforms and output polarity is fixed as the configured inactive polarity (PWMx_CTRL.inactive_pol).

19.3.3 One-shot mode

Unlike the continuous mode, the PWM channel generates the output waveforms within the configured periods (PWM_CTRL.rpt + 1), and then stops. At the same times, an interrupt is asserted to inform that the operation has been finished.

There are also two output modes for the one-shot mode: the left-aligned mode and the center-aligned mode.

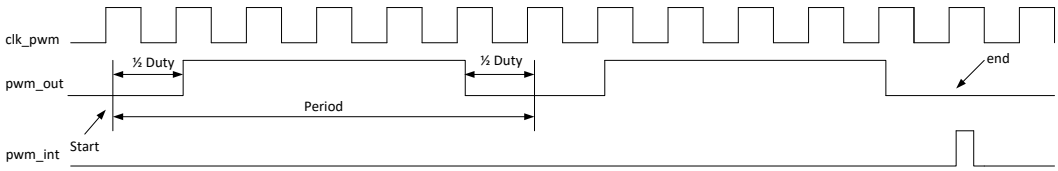


Fig. 19-5 PWM One-shot Center-aligned Output Mode

19.4 Register Description

19.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
PWM_PWM0_CNT	0x0000	W	0x00000000	PWM Channel 0 Counter Register
PWM_PWM0_PERIOD_HPR	0x0004	W	0x00000000	PWM Channel 0 Period Register/High Polarity Capture Register
PWM_PWM0_DUTY_LPR	0x0008	W	0x00000000	PWM Channel 0 Duty Register/Low Polarity Capture Register
PWM_PWM0_CTRL	0x000c	W	0x00000000	PWM Channel 0 Control Register
PWM_PWM1_CNT	0x0010	W	0x00000000	PWM Channel 1 Counter Register
PWM_PWM1_PERIOD_HPR	0x0014	W	0x00000000	PWM Channel 1 Period Register/High Polarity Capture Register
PWM_PWM1_DUTY_LPR	0x0018	W	0x00000000	PWM Channel 1 Duty Register/Low Polarity Capture Register
PWM_PWM1_CTRL	0x001c	W	0x00000000	PWM Channel 1 Control Register
PWM_PWM2_CNT	0x0020	W	0x00000000	PWM Channel 2 Counter Register
PWM_PWM2_PERIOD_HPR	0x0024	W	0x00000000	PWM Channel 2 Period Register/High Polarity Capture Register
PWM_PWM2_DUTY_LPR	0x0028	W	0x00000000	PWM Channel 2 Duty Register/Low Polarity Capture Register
PWM_PWM2_CTRL	0x002c	W	0x00000000	PWM Channel 2 Control Register
PWM_PWM3_CNT	0x0030	W	0x00000000	PWM Channel 3 Counter Register
PWM_PWM3_PERIOD_HPR	0x0034	W	0x00000000	PWM Channel 3 Period Register/High Polarity Capture Register
PWM_PWM3_DUTY_LPR	0x0038	W	0x00000000	PWM Channel 3 Duty Register/Low Polarity Capture Register
PWM_PWM3_CTRL	0x003c	W	0x00000000	PWM Channel 3 Control Register
PWM_INTSTS	0x0040	W	0x00000000	Interrupt Status Register
PWM_INT_EN	0x0044	W	0x00000000	Interrupt Enable Register
PWM_PWM_FIFO_CTRL	0x0050	W	0x00000000	PWM Channel 3 FIFO Mode Control Register

Name	Offset	Size	Reset Value	Description
PWM_PWM_FIFO_INTSTS	0x0054	W	0x00000000	FIFO Interrupts Status Register
PWM_PWM_FIFO_TOUTTHR	0x0058	W	0x00000000	FIFO Timeout Threshold Register
PWM_PWM_FIFO	0x0060 ~0x007C	W	0x00000000	FIFO Register

*Notes:Size:**B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access*

19.4.2 Detail Register Description

PWM_PWM0_CNT

Address: Operational Base + offset (0x0000)

PWM Channel 0 Counter Register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	CNT Timer Counter The 32-bit indicates current value of PWM Channel 0 counter. The counter runs at the rate of PWM clock. The value ranges from 0 to (2^32-1).

PWM_PWM0_PERIOD_HPR

Address: Operational Base + offset (0x0004)

PWM Channel 0 Period Register/High Polarity Capture Register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	PERIOD_HPR Output Waveform Period/Input Waveform High Polarity Cycle If PWM is operated at the continuous mode or one-shot mode, this value defines the period of the output waveform. Note that, if the PWM is operated at the center-aligned mode, the period should be an even one, and therefore only the bit [31:1] is taken into account and bit [0] always considered as 0. If PWM is operated at the capture mode, this value indicates the effective high polarity cycles of input waveform. This value is based on the PWM clock. The value ranges from 0 to (2^32-1).

PWM_PWM0_DUTY_LPR

Address: Operational Base + offset (0x0008)

PWM Channel 0 Duty Register/Low Polarity Capture Register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	DUTY_LPR Output Waveform Duty Cycle/Input Waveform Low Polarity Cycle If PWM is operated at the continuous mode or one-shot mode, this value defines the duty cycle of the output waveform. The PWM starts the output waveform with duty cycle. Note that, if the PWM is operated at the center-aligned mode, the period should be an even one, and therefore only the [31:1] is taken into account. If PWM is operated at the capture mode, this value indicates the effective low polarity cycles of input waveform. This value is based on the PWM clock. The value ranges from 0 to (2^32-1).

PWM_PWM0_CTRL
Address: Operational Base + offset (0x000c)
PWM Channel 0 Control Register

Bit	Attr	Reset Value	Description
31:24	RW	0x00	rpt Repeat Counter This field defines the repeated effective periods of output waveform in one-shot mode. The value N means N+1 repeated effective periods.
23:16	RW	0x00	scale Scale Factor This field defines the scale factor applied to prescaled clock. The value N means the clock is divided by 2*N. If N is 0, it means that the clock is divided by 512(2*256).
15	RO	0x0	reserved
14:12	RW	0x0	prescale Prescale Factor This field defines the prescale factor applied to input clock. The value N means that the input clock is divided by 2^N.
11:10	RO	0x0	reserved
9	RW	0x0	clk_sel Clock Source Select 0: non-scaled clock is selected as PWM clock source. It means that the prescale clock is directly used as the PWM clock source 1: scaled clock is selected as PWM clock source
8	RW	0x0	lp_en Low Power Mode Enable 0: disabled 1: enabled When PWM channel is inactive state and Low Power Mode is enabled, the path to PWM Clock prescale module is blocked to reduce power consumption.

Bit	Attr	Reset Value	Description
7:6	RO	0x0	reserved
5	RW	0x0	output_mode PWM Output mode 0: left aligned mode 1: center aligned mode
4	RW	0x0	inactive_pol Inactive State Output Polarity This defines the output waveform polarity when PWM channel is in inactive state. The inactive state means that PWM finishes the complete waveform in one-shot mode or PWM channel is disabled. 0: negative 1: positive
3	RW	0x0	duty_pol Duty Cycle Output Polarity This defines the polarity for duty cycle. PWM starts the output waveform with duty cycle. 0: negative 1: positive
2:1	RW	0x0	pwm_mode PWM Operation Mode 00: One shot mode. PWM produces the waveform within the repeated times defined by PWMx_CTRL_rpt. 01: Continuous mode. PWM produces the waveform continuously 10: Capture mode. PWM measures the cycles of high/low polarity of input waveform. 11: reserved
0	RW	0x0	pwm_en PWM channel enable 0: disabled 1: enabled. If the PWM is worked in the one-shot mode, this bit will be cleared at the end of operation

PWM_PWM1_CNT
Address: Operational Base + offset (0x0010)
PWM Channel 1 Counter Register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	CNT Timer Counter The 32-bit indicates current value of PWM Channel 1 counter. The counter runs at the rate of PWM clock. The value ranges from 0 to (2^32-1).

PWM_PWM1_PERIOD_HPR
Address: Operational Base + offset (0x0014)

PWM Channel 1 Period Register/High Polarity Capture Register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	PERIOD_HPR Output Waveform Period/Input Waveform High Polarity Cycle If PWM is operated at the continuous mode or one-shot mode, this value defines the period of the output waveform. Note that, if the PWM is operated at the center-aligned mode, the period should be an even one, and therefore only the bit [31:1] is taken into account and bit [0] always considered as 0. If PWM is operated at the capture mode, this value indicates the effective high polarity cycles of input waveform. This value is based on the PWM clock. The value ranges from 0 to (2^32-1).

PWM_PWM1_DUTY_LPR

Address: Operational Base + offset (0x0018)

PWM Channel 1 Duty Register/Low Polarity Capture Register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	DUTY_LPR Output Waveform Duty Cycle/Input Waveform Low Polarity Cycle If PWM is operated at the continuous mode or one-shot mode, this value defines the duty cycle of the output waveform. The PWM starts the output waveform with duty cycle. Note that, if the PWM is operated at the center-aligned mode, the period should be an even one, and therefore only the [31:1] is taken into account. If PWM is operated at the capture mode, this value indicates the effective low polarity cycles of input waveform. This value is based on the PWM clock. The value ranges from 0 to (2^32-1).

PWM_PWM1_CTRL

Address: Operational Base + offset (0x001c)

PWM Channel 1 Control Register

Bit	Attr	Reset Value	Description
31:24	RW	0x00	rpt Repeat Counter This field defines the repeated effective periods of output waveform in one-shot mode. The value N means N+1 repeated effective periods.
23:16	RW	0x00	scale Scale Factor This field defines the scale factor applied to prescaled clock. The value N means the clock is divided by 2*N. If N is 0, it means that the clock is divided by 512(2*256).
15	RO	0x0	reserved

Bit	Attr	Reset Value	Description
14:12	RW	0x0	prescale Prescale Factor This field defines the prescale factor applied to input clock. The value N means that the input clock is divided by 2^N.
11:10	RO	0x0	reserved
9	RW	0x0	clk_sel Clock Source Select 0: non-scaled clock is selected as PWM clock source. It means that the prescale clock is directly used as the PWM clock source 1: scaled clock is selected as PWM clock source
8	RW	0x0	lp_en Low Power Mode Enable 0: disabled 1: enabled When PWM channel is inactive state and Low Power Mode is enabled, the path to PWM Clock prescale module is blocked to reduce power consumption.
7:6	RO	0x0	reserved
5	RW	0x0	output_mode PWM Output mode 0: left aligned mode 1: center aligned mode
4	RW	0x0	inactive_pol Inactive State Output Polarity This defines the output waveform polarity when PWM channel is in inactive state. The inactive state means that PWM finishes the complete waveform in one-shot mode or PWM channel is disabled. 0: negative 1: positive
3	RW	0x0	duty_pol Duty Cycle Output Polarity This defines the polarity for duty cycle. PWM starts the output waveform with duty cycle. 0: negative 1: positive
2:1	RW	0x0	pwm_mode PWM Operation Mode 00: One shot mode. PWM produces the waveform within the repeated times defined by PWMx_CTRL_rpt 01: Continuous mode. PWM produces the waveform continuously 10: Capture mode. PWM measures the cycles of high/low polarity of input waveform. 11: reserved

Bit	Attr	Reset Value	Description
0	RW	0x0	pwm_en PWM channel enable 0: disabled 1: enabled. If the PWM is worked in the one-shot mode, this bit will be cleared at the end of operation

PWM_PWM2_CNT
Address: Operational Base + offset (0x0020)
PWM Channel 2 Counter Register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	CNT Timer Counter The 32-bit indicates current value of PWM Channel 2 counter. The counter runs at the rate of PWM clock. The value ranges from 0 to (2^32-1).

PWM_PWM2_PERIOD_HPR
Address: Operational Base + offset (0x0024)
PWM Channel 2 Period Register/High Polarity Capture Register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	PERIOD_HPR Output Waveform Period/Input Waveform High Polarity Cycle If PWM is operated at the continuous mode or one-shot mode, this value defines the period of the output waveform. Note that, if the PWM is operated at the center-aligned mode, the period should be an even one, and therefore only the bit [31:1] is taken into account and bit [0] always considered as 0. If PWM is operated at the capture mode, this value indicates the effective high polarity cycles of input waveform. This value is based on the PWM clock. The value ranges from 0 to (2^32-1).

PWM_PWM2_DUTY_LPR
Address: Operational Base + offset (0x0028)
PWM Channel 2 Duty Register/Low Polarity Capture Register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	DUTY_LPR Output Waveform Duty Cycle/Input Waveform Low Polarity Cycle If PWM is operated at the continuous mode or one-shot mode, this value defines the duty cycle of the output waveform. The PWM starts the output waveform with duty cycle. Note that, if the PWM is operated at the center-aligned mode, the period should be an even one, and therefore only the [31:1] is taken into account. If PWM is operated at the capture mode, this value indicates the effective low polarity cycles of input waveform. This value is based on the PWM clock. The value ranges from 0 to (2^32-1).

PWM_PWM2_CTRL
Address: Operational Base + offset (0x002c)
PWM Channel 2 Control Register

Bit	Attr	Reset Value	Description
31:24	RW	0x00	rpt Repeat Counter This field defines the repeated effective periods of output waveform in one-shot mode. The value N means N+1 repeated effective periods.
23:16	RW	0x00	scale Scale Factor This field defines the scale factor applied to prescaled clock. The value N means the clock is divided by 2*N. If N is 0, it means that the clock is divided by 512(2*256).
15	RO	0x0	reserved
14:12	RW	0x0	prescale Prescale Factor This field defines the prescale factor applied to input clock. The value N means that the input clock is divided by 2^N.
11:10	RO	0x0	reserved
9	RW	0x0	clk_sel Clock Source Select 0: non-scaled clock is selected as PWM clock source. It means that the prescale clock is directly used as the PWM clock source 1: scaled clock is selected as PWM clock source

Bit	Attr	Reset Value	Description
8	RW	0x0	lp_en Low Power Mode Enable 0: disabled 1: enabled When PWM channel is inactive state and Low Power Mode is enabled, the path to PWM Clock prescale module is blocked to reduce power consumption.
7:6	RO	0x0	reserved
5	RW	0x0	output_mode PWM Output mode 0: left aligned mode 1: center aligned mode
4	RW	0x0	inactive_pol Inactive State Output Polarity This defines the output waveform polarity when PWM channel is in inactive state. The inactive state means that PWM finishes the complete waveform in one-shot mode or PWM channel is disabled. 0: negative 1: positive
3	RW	0x0	duty_pol Duty Cycle Output Polarity This defines the polarity for duty cycle. PWM starts the output waveform with duty cycle. 0: negative 1: positive
2:1	RW	0x0	pwm_mode PWM Operation Mode 00: One shot mode. PWM produces the waveform within the repeated times defined by PWMx_CTRL_rpt. 01: Continuous mode. PWM produces the waveform continuously 10: Capture mode. PWM measures the cycles of high/low polarity of input waveform. 11: reserved
0	RW	0x0	pwm_en PWM channel enable 0: disabled 1: enabled. If the PWM is worked in the one-shot mode, this bit will be cleared at the end of operation

PWM_PWM3_CNT
Address: Operational Base + offset (0x0030)
PWM Channel 3 Counter Register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	CNT Timer Counter The 32-bit indicates current value of PWM Channel 3 counter. The counter runs at the rate of PWM clock. The value ranges from 0 to (2^32-1).

PWM_PWM3_PERIOD_HPR
Address: Operational Base + offset (0x0034)
PWM Channel 3 Period Register/High Polarity Capture Register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	PERIOD_HPR Output Waveform Period/Input Waveform High Polarity Cycle If PWM is operated at the continuous mode or one-shot mode, this value defines the period of the output waveform. Note that, if the PWM is operated at the center-aligned mode, the period should be an even one, and therefore only the bit [31:1] is taken into account and bit [0] always considered as 0. If PWM is operated at the capture mode, this value indicates the effective high polarity cycles of input waveform. This value is based on the PWM clock. The value ranges from 0 to (2^32-1).

PWM_PWM3_DUTY_LPR
Address: Operational Base + offset (0x0038)
PWM Channel 3 Duty Register/Low Polarity Capture Register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	DUTY_LPR Output Waveform Duty Cycle/Input Waveform Low Polarity Cycle If PWM is operated at the continuous mode or one-shot mode, this value defines the duty cycle of the output waveform. The PWM starts the output waveform with duty cycle. Note that, if the PWM is operated at the center-aligned mode, the period should be an even one, and therefore only the [31:1] is taken into account. If PWM is operated at the capture mode, this value indicates the effective low polarity cycles of input waveform. This value is based on the PWM clock. The value ranges from 0 to (2^32-1).

PWM_PWM3_CTRL
Address: Operational Base + offset (0x003c)
PWM Channel 3 Control Register

Bit	Attr	Reset Value	Description
31:24	RW	0x00	rpt Repeat Counter This field defines the repeated effective periods of output waveform in one-shot mode. The value N means N+1 repeated effective periods.
23:16	RW	0x00	scale Scale Factor This field defines the scale factor applied to prescaled clock. The value N means the clock is divided by 2*N. If N is 0, it means that the clock is divided by 512(2*256).
15	RO	0x0	reserved
14:12	RW	0x0	prescale Prescale Factor This field defines the prescale factor applied to input clock. The value N means that the input clock is divided by 2^N.
11:10	RO	0x0	reserved
9	RW	0x0	clk_sel Clock Source Select 0: non-scaled clock is selected as PWM clock source. It means that the prescale clock is directly used as the PWM clock source 1: scaled clock is selected as PWM clock source
8	RW	0x0	lp_en Low Power Mode Enable 0: disabled 1: enabled When PWM channel is inactive state and Low Power Mode is enabled, the path to PWM Clock prescale module is blocked to reduce power consumption.
7:6	RO	0x0	reserved
5	RW	0x0	output_mode PWM Output mode 0: left aligned mode 1: center aligned mode
4	RW	0x0	inactive_pol Inactive State Output Polarity This defines the output waveform polarity when PWM channel is in inactive state. The inactive state means that PWM finishes the complete waveform in one-shot mode or PWM channel is disabled. 0: negative 1: positive

Bit	Attr	Reset Value	Description
3	RW	0x0	duty_pol Duty Cycle Output Polarity This defines the polarity for duty cycle. PWM starts the output waveform with duty cycle. 0: negative 1: positive
2:1	RW	0x0	pwm_mode PWM Operation Mode 00: One shot mode. PWM produces the waveform within the repeated times defined by PWMx_CTRL_rpt 01: Continuous mode. PWM produces the waveform continuously 10: Capture mode. PWM measures the cycles of high/low polarity of input waveform. 11: reserved
0	RW	0x0	pwm_en PWM channel enable 0: disabled 1: enabled. If the PWM is worked in the one-shot mode, this bit will be cleared at the end of operation

PWM_INTSTS
Address: Operational Base + offset (0x0040)
Interrupt Status Register

Bit	Attr	Reset Value	Description
31:12	RO	0x0	reserved
11	RO	0x0	CH3_Pol Channel 3 Interrupt Polarity Flag This bit is used in capture mode in order to identify the transition of the input waveform when interrupt is generated. When bit is 1, please refer to PWM3_PERIOD_HPR to know the effective high cycle of Channel 3 input waveform. Otherwise, please refer to PWM3_PERIOD_LPR to know the effective low cycle of Channel 3 input waveform. Write 1 to CH3_IntSts will clear this bit.
10	RO	0x0	CH2_Pol Channel 2 Interrupt Polarity Flag This bit is used in capture mode in order to identify the transition of the input waveform when interrupt is generated. When bit is 1, please refer to PWM2_PERIOD_HPR to know the effective high cycle of Channel 2 input waveform. Otherwise, please refer to PWM2_PERIOD_LPR to know the effective low cycle of Channel 2 input waveform. Write 1 to CH2_IntSts will clear this bit.

Bit	Attr	Reset Value	Description
9	RO	0x0	CH1_Pol Channel 1 Interrupt Polarity Flag This bit is used in capture mode in order to identify the transition of the input waveform when interrupt is generated. When bit is 1, please refer to PWM1_PERIOD_HPR to know the effective high cycle of Channel 1 input waveform. Otherwise, please refer to PWM1_PERIOD_LPR to know the effective low cycle of Channel 1 input waveform. Write 1 to CH1_IntSts will clear this bit.
8	RO	0x0	CH0_Pol Channel 0 Interrupt Polarity Flag This bit is used in capture mode in order to identify the transition of the input waveform when interrupt is generated. When bit is 1, please refer to PWM0_PERIOD_HPR to know the effective high cycle of Channel 0 input waveform. Otherwise, please refer to PWM0_PERIOD_LPR to know the effective low cycle of Channel 0 input waveform. Write 1 to CH0_IntSts will clear this bit.
7:4	RO	0x0	reserved
3	RW	0x0	CH3_IntSts Channel 3 Interrupt Status 0: Channel 3 Interrupt not generated 1: Channel 3 Interrupt generated
2	RW	0x0	CH2_IntSts Channel 2 Interrupt Status 0: Channel 2 Interrupt not generated 1: Channel 2 Interrupt generated
1	RW	0x0	CH1_IntSts Channel 1 Interrupt Status 0: Channel 1 Interrupt not generated 1: Channel 1 Interrupt generated
0	RW	0x0	CH0_IntSts Channel 0 Raw Interrupt Status 0: Channel 0 Interrupt not generated 1: Channel 0 Interrupt generated

PWM_INT_EN
Address: Operational Base + offset (0x0044)
Interrupt Enable Register

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3	RW	0x0	CH3_Int_en Channel 3 Interrupt Enable 0: Channel 3 Interrupt disabled 1: Channel 3 Interrupt enabled

Bit	Attr	Reset Value	Description
2	RW	0x0	CH2_Int_en Channel 2 Interrupt Enable 0: Channel 2 Interrupt disabled 1: Channel 2 Interrupt enabled
1	RW	0x0	CH1_Int_en Channel 1 Interrupt Enable 0: Channel 1 Interrupt disabled 1: Channel 1 Interrupt enabled
0	RW	0x0	CH0_Int_en Channel 0 Interrupt Enable 0: Channel 0 Interrupt disabled 1: Channel 0 Interrupt enabled

PWM_PWM_FIFO_CTRL
Address: Operational Base + offset (0x0050)
PWM Channel 3 FIFO Mode Control Register

Bit	Attr	Reset Value	Description
31:10	RO	0x0	reserved
9	RW	0x0	timeout_en FIFO Timeout Enable
8	RW	0x0	dma_mode_en DMA Mode Enable 1'b1: enable 1'b0: disable
7	RO	0x0	reserved
6:4	RW	0x0	almost_full_watermark Almost Full Watermark Level
3	RW	0x0	watermark_int_en Watermark Full Interrupt
2	RW	0x0	overflow_int_en FIFO Overflow Interrupt Enable When high, an interrupt asserts when the channel 3 FIFO is overflow.
1	RW	0x0	full_int_en FIFO Full Interrupt Enable When high, an interrupt asserts when the channel 3 FIFO is full.
0	RW	0x0	fifo_mode_sel FIFO MODE Sel When high, PWM FIFO mode is activated

PWM_PWM_FIFO_INTSTS
Address: Operational Base + offset (0x0054)
FIFO Interrupts Status Register

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved

Bit	Attr	Reset Value	Description
4	RO	0x0	fifo_empty_status FIFO Empty Status This bit indicates the FIFO is empty
3	W1C	0x0	timeout_intsts Timeout Interrupt
2	W1C	0x0	fifo_watermark_full_intsts FIFO Watermark Full Interrupt Status This bit indicates the FIFO is Watermark Full
1	W1C	0x0	fifo_overflow_intsts FIFO Overflow Interrupt Status This bit indicates the FIFO is overflow
0	W1C	0x0	fifo_full_intsts FIFO Full Interrupt Status This bit indicates the FIFO is full

PWM_PWM_FIFO_TOUTTHR
Address: Operational Base + offset (0x0058)
FIFO Timeout Threshold Register

Bit	Attr	Reset Value	Description
31:20	RO	0x0	reserved
19:0	RO	0x00000	timeout_threshold FIFO Timeout Value(unit pwmclk)

PWM_PWM_FIFO
Address: Operational Base + offset (0x0060~0x007C)
FIFO Register

Bit	Attr	Reset Value	Description
31	RW	0x0	pol Polarity This bit indicates the polarity of the lower 31-bit counter. 0: Low 1: High
30:0	RO	0x00000000	cycle_cnt High/Low Cycle Counter This 31-bit counter indicates the effective cycles of high/low waveform.

19.5 Interface Description

Table 19-1 PWM Interface Description

Module Pin	Direction	Pad Name	IOMUX Setting
PWM0	I/O	IO_PWM0_VOP0pwm_VOP1pwm_GPIO1830gpio4c2	GRF_GPIO4C_IOMUX[5:4]=2'b01
PWM1	I/O	IO_PWM1_GPIO1830gpio4c6	GRF_GPIO4C_IOMUX[13:12]=2'b01

PWM2	I/O	IO_PWM2_PMU1830gpio1c3	GRF_GPIO1C_IOMUX[7:6] =2'b01
PWM3	I/O	IO_PWMA3_PMUdebug4_PMU18gpio0a6	GRF_GPIO0A_IOMUX[13:12] =2'b01 PMUGRF_SOC_CON0[5]=1'b0
		IO_PWMB3_PMU1830gpio1b6	GRF_GPIO1B_IOMUX[13:12] =2'b01 PMUGRF_SOC_CON0[5]=1'b1

Notes: I=input, O=output, I/O=input/output.

19.6 Application Notes

19.6.1 PWM Capture Mode Standard Usage Flow

1. Set PWMx_CTRL.pwm_en to '0' to disable the PWM channel.
2. Choose the prescale factor and the scale factor for pclk by programming PWMx_CTRL.prescale and PWMx_CTRL.scale, and select the clock needed by setting PWMx_CTRL.clk_sel.
3. Configure the channel to work in the capture mode.
4. Enable the INT_EN.chx_int_en to enable the interrupt generation.
5. Enable the channel by writing '1' to PWMx_CTRL.pwm_en bit to start the channel.
6. When an interrupt is asserted, refer to INTSTS register to know the raw interrupt status. If the corresponding polarity flag is set, turn to PWMx_PERIOD_HPC register to know the effective high cycles of input waveforms, otherwise turn to PWMx_DUTY_LPC register to know the effective low cycles.
7. Write '0' to PWMx_CTRL.pwm_en to disable the channel.

19.6.2 PWM Capture DMA Mode Standard Usage Flow

1. Set PWMx_CTRL.pwm_en to '0' to disable the PWM channel.
2. Choose the prescale factor and the scale factor for pclk by programming PWMx_CTRL.prescale and PWMx_CTRL.scale, and select the clock needed by setting PWMx_CTRL.clk_sel.
3. Configure the channel 3 to work in the capture mode.
4. Configure the PWM_FIFO_CTRL.dma_mode_en and PWM_FIFO_CTRL.fifo_mode_sel to enable the DMA mode. Configure PWM_FIFO_CTRL.almost_full_watermark at appropriate value.
5. Configure DMAC_BUS to tansfer data from PWM to DDR.
6. Enable the channel by writing '1' to PWMx_CTRL.pwm_en bit to start the channel.
7. When an dma_req is asserted, DMAC_BUS transfer the data of effective high cycles and low cycles of input waveforms to DDR.
8. Write '0' to PWMx_CTRL.pwm_en to disable the channel.

19.6.3 PWM One-shot Mode/Continuous Standard Usage Flow

1. Set PWMx_CTRL.pwm_en to '0' to disable the PWM channel.
2. Choose the prescale factor and the scale factor for pclk by programming PWMx_CTRL.prescale and PWMx_CTRL.scale, and select the clock needed by setting PWMx_CTRL.clk_sel.
3. Choose the output mode by setting PWMx_CTRL.output_mode, and set the duty polarity and inactive polarity by programming PWMx_CTRL.duty_pol and PWMx_CTRL.inactive_pol.
4. Set the PWMx_CTRL.rpt if the channel is desired to work in the one-shot mode.
5. Configure the channel to work in the one-shot mode or the continuous mode.
6. Enable the INT_EN.chx_int_en to enable the interrupt generation if if the channel is desired to work in the one-shot mode.
7. If the channel is working in the one-shot mode, an interrupt is asserted after the end of operation, and the PWMx_CTRL.pwm_en is automatically cleared. Whatever mode the channel is working in, write '0' to PWMx_CTRL.pwm_en bit to disable the PWM channel.

19.6.4 Low-power mode

Setting PWMx_CTRL.lp_en to '1' makes the channel enter the low-power mode. When the PWM channel is inactive, the APB bus clock to the clock prescale module is gated in order

to reduce the power consumption. It is recommended to disable the channel before entering the low-power mode, and quit the low-power mode before enabling the channel.

19.6.5 Other notes

When the channel is active to produce waveforms, it is free to program the PWMx_PERIOD_HPC and PWMx_DUTY_LPC register. The change will not take effect immediately until the current period ends.

An active channel can be changed to another operation mode without disable the PWM channel. However, during the transition of the operation mode there may be some irregular output waveforms. So does changing the clock division factor when the channel is active.

Chapter 20 UART Interface

20

20.1 Overview

The Universal Asynchronous Receiver/Transmitter (UART) is used for serial communication with a peripheral, modem (data carrier equipment, DCE) or data set. Data is written from a master (CPU) over the APB bus to the UART and it is converted to serial form and transmitted to the destination device. Serial data is also received by the UART and stored for the master (CPU) to read back.
UART Controller supports the following features:

- Support 5 independent UART controller: UART0, UART1, UART2, UART3, UART4
- All contain two 64Bytes FIFOs for data receive and transmit
- UART0/UART3 support auto flow-control
- Support bit rates 115.2Kbps, 460.8Kbps, 921.6Kbps, 1.5Mbps, 3Mbps, 4Mbps
- Support programmable baud rates, even with non-integer clock divider
- Standard asynchronous communication bits (start, stop and parity)
- Support interrupt-based or DMA-based mode
- Support 5-8 bits width transfer

20.2 Block Diagram

This section provides a description about the functions and behavior under various conditions. The UART Controller comprises with:

- AMBA APB interface
- FIFO controllers
- Register block
- Modem synchronization block and baud clock generation block
- Serial receiver and serial transmitter

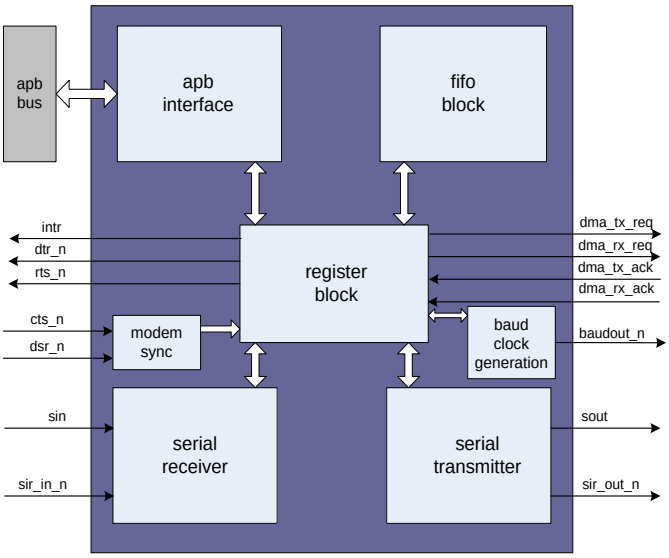


Fig. 20-1 UART Architecture

APB INTERFACE
The host processor accesses data, control, and status information on the UART through the APB interface. The UART supports APB data bus widths of 8, 16, and 32 bits.
Register block
Be responsible for the main UART functionality including control, status and interrupt generation.
Modem Synchronization block
Synchronizes the modem input signal.
FIFO block
Be responsible for FIFO control and storage (when using internal RAM) or signaling to control external RAM (when used).

Baud Clock Generator
Generates the transmitter and receiver baud clock along with the output reference clock signal (baudout_n).
Serial Transmitter
Converts the parallel data, written to the UART, into serial form and adds all additional bits, as specified by the control register, for transmission. This makeup of serial data, referred to as a character can exit the block in two forms, either serial UART format or IrDA 1.0 SIR format.
Serial Receiver
Converts the serial data character (as specified by the control register) received in either the UART or IrDA 1.0 SIR format to parallel form. Parity error detection, framing error detection and line break detection is carried out in this block.

20.3 Function Description

UART (RS232) Serial Protocol
Because the serial communication is asynchronous, additional bits (start and stop) are added to the serial data to indicate the beginning and end. An additional parity bit may be added to the serial character. This bit appears after the last data bit and before the stop bit(s) in the character structure to perform simple error checking on the received data, as shown in Figure.

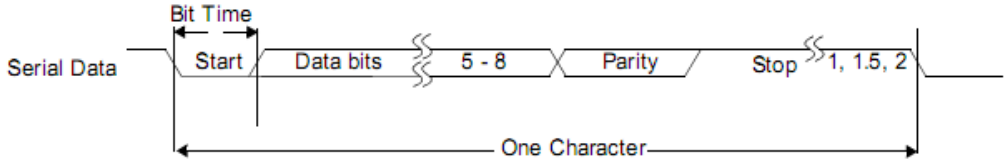


Fig. 20-2 UART Serial protocol

IrDA 1.0 SIR Protocol
The Infrared Data Association (IrDA) 1.0 Serial Infrared (SIR) mode supports bi-directional data communications with remote devices using infrared radiation as the transmission medium. IrDA 1.0 SIR mode specifies a maximum baud rate of 115.2 Kbaud.
Transmitting a single infrared pulse signals a logic zero, while a logic one is represented by not sending a pulse. The width of each pulse is 3/16ths of a normal serial bit time. Data transfers can only occur in half-duplex fashion when IrDA SIR mode is enabled.

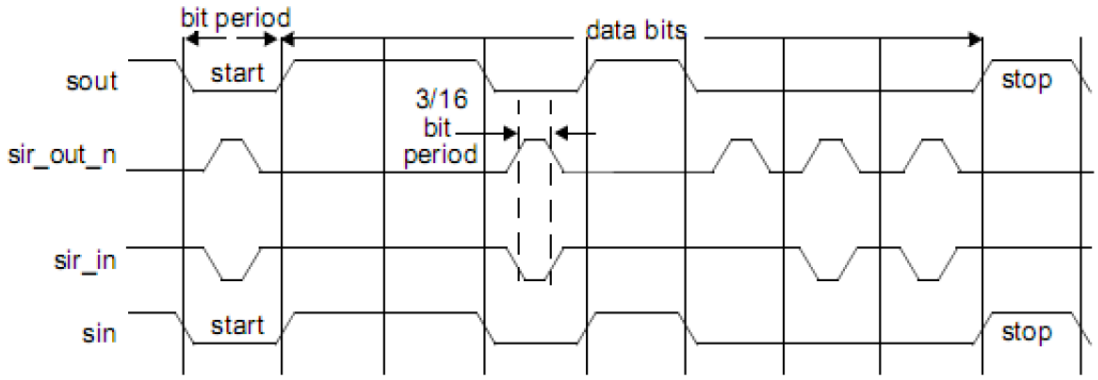


Fig. 20-3 IrDA 1.0

Baud Clock
The baud rate is controlled by the serial clock (sclk or pclk in a single clock implementation) and the Divisor Latch Register (DLH and DLL). As the exact number of baud clocks that each bit was transmitted for is known, calculating the mid-point for sampling is not difficult, that is every 16 baud clocks after the mid-point sample of the start bit.

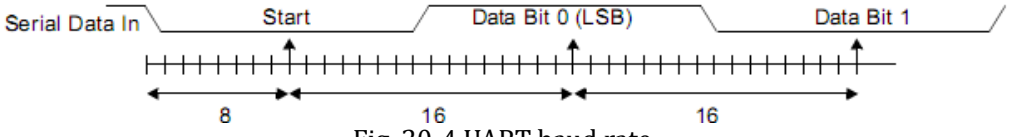


Fig. 20-4 UART baud rate

FIFO Support

1. NONE FIFO MODE

If FIFO support is not selected, then no FIFOs are implemented and only a single receive data byte and transmit data byte can be stored at a time in the RBR and THR.

2. FIFO MODE

The FIFO depth of UART0/UART1/UART2 is 64bytes. The FIFO mode of all the UART is enabled by register FCR[0].

Interrupts

The following interrupt types can be enabled with the IER register.

- Receiver Error
- Receiver Data Available
- Character Timeout (in FIFO mode only)
- Transmitter Holding Register Empty at/below threshold (in Programmable THRE Interrupt mode)
- Modem Status

DMA Support

The UART supports DMA signaling with the use of two output signals (dma_tx_req_n and dma_rx_req_n) to indicate when data is ready to be read or when the transmit FIFO is empty.

The dma_tx_req_n signal is asserted under the following conditions:

- When the Transmitter Holding Register is empty in non-FIFO mode.
- When the transmitter FIFO is empty in FIFO mode with Programmable THRE interrupt mode disabled.
- When the transmitter FIFO is at, or below the programmed threshold with Programmable THRE interrupt mode enabled.

The dma_rx_req_n signal is asserted under the following conditions:

- When there is a single character available in the Receive Buffer Register in non-FIFO mode.
- When the Receiver FIFO is at or above the programmed trigger level in FIFO mode.

Auto Flow Control

The UART can be configured to have a 16750-compatible Auto RTS and Auto CTS serial data flow control mode available. If FIFOs are not implemented, then this mode cannot be selected. When Auto Flow Control mode has been selected, it can be enabled with the Modem Control Register (MCR[5]). Following figure shows a block diagram of the Auto Flow Control functionality.

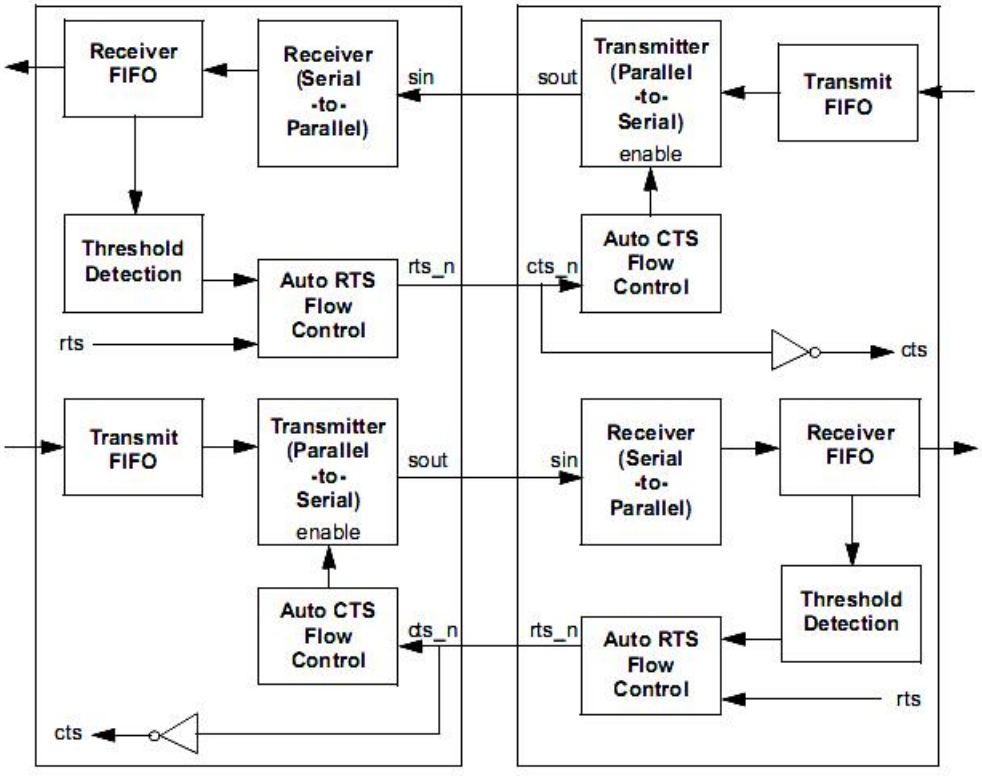


Fig. 20-5 UART Auto flow control block diagram

Auto RTS – Becomes active when the following occurs:
- Auto Flow Control is selected during configuration
- FIFOs are implemented
- RTS (MCR[1] bit and MCR[5]bit are both set)
- FIFOs are enabled (FCR[0]) bit is set)
- SIR mode is disabled (MCR[6] bit is not set)

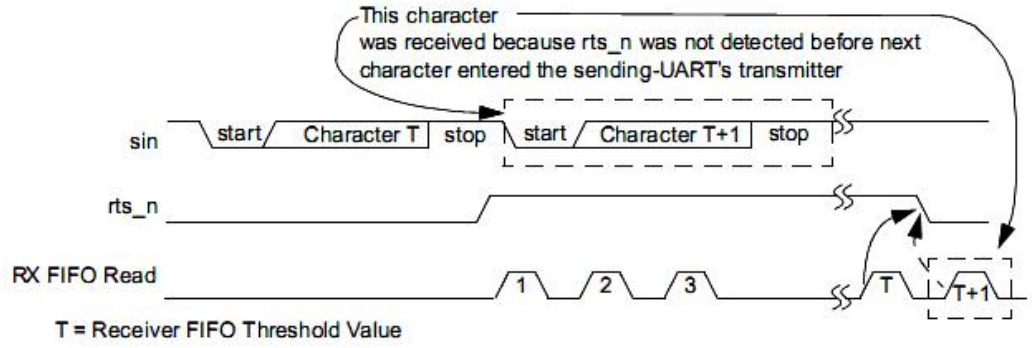


Fig. 20-6 UART AUTO RTS TIMING

Auto CTS – becomes active when the following occurs:
- Auto Flow Control is selected during configuration
- FIFOs are implemented
- AFCE (MCR[5] bit is set)
- FIFOs are enabled through FIFO Control Register FCR[0] bit
- SIR mode is disabled (MCR[6] bit is not set)

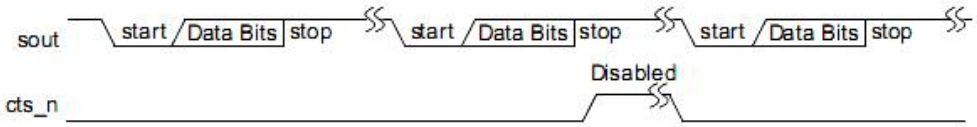


Fig. 20-7 UART AUTO CTS TIMING

20.4 Register Description

This section describes the control/status registers of the design. There are 3 UARTs in RK3228, and each one has its own base address.

20.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
UART_RBR	0x0000	W	0x00000000	Receive Buffer Register
UART_THR	0x0000	W	0x00000000	Transmit Holding Register
UART_DLL	0x0000	W	0x00000000	Divisor Latch (Low)
UART_DLH	0x0004	W	0x00000000	Divisor Latch (High)
UART_IER	0x0004	W	0x00000000	Interrupt Enable Register
UART_IIR	0x0008	W	0x00000000	Interrupt Identification Register
UART_FCR	0x0008	W	0x00000000	FIFO Control Register
UART_LCR	0x000c	W	0x00000000	Line Control Register
UART_MCR	0x0010	W	0x00000000	Modem Control Register
UART_LSR	0x0014	W	0x00000000	Line Status Register
UART_MSR	0x0018	W	0x00000000	Modem Status Register
UART_SCR	0x001c	W	0x00000000	Scratchpad Register
UART_SRBR	0x0030	W	0x00000000	Shadow Receive Buffer Register
UART_STHR	0x006c	W	0x00000000	Shadow Transmit Holding Register

Name	Offset	Size	Reset Value	Description
UART_FAR	0x0070	W	0x00000000	FIFO Access Register
UART_TFR	0x0074	W	0x00000000	Transmit FIFO Read
UART_RFW	0x0078	W	0x00000000	Receive FIFO Write
UART_USR	0x007c	W	0x00000000	UART Status Register
UART_TFL	0x0080	W	0x00000000	Transmit FIFO Level
UART_RFL	0x0084	W	0x00000000	Receive FIFO Level
UART_SRR	0x0088	W	0x00000000	Software Reset Register
UART_SRTS	0x008c	W	0x00000000	Shadow Request to Send
UART_SBCR	0x0090	W	0x00000000	Shadow Break Control Register
UART_SDMAM	0x0094	W	0x00000000	Shadow DMA Mode
UART_SFE	0x0098	W	0x00000000	Shadow FIFO Enable
UART_SRT	0x009c	W	0x00000000	Shadow RCVR Trigger
UART_STET	0x00a0	W	0x00000000	Shadow TX Empty Trigger
UART_HTX	0x00a4	W	0x00000000	Halt TX
UART_DMASA	0x00a8	W	0x00000000	DMA Software Acknowledge
UART_CPR	0x00f4	W	0x00000000	Component Parameter Register
UART_UCV	0x00f8	W	0x0330372a	UART Component Version
UART_CTR	0x00fc	W	0x44570110	Component Type Register

*Notes:Size:**B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access*

20.4.2 Detail Register Description

UART_RBR

Address: Operational Base + offset (0x0000)

Receive Buffer Register

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RW	0x00	data_input Data byte received on the serial input port (sin) in UART mode, or the serial infrared input (sir_in) in infrared mode. The data in this register is valid only if the Data Ready (DR) bit in the Line Status Register (LCR) is set. If in non-FIFO mode (FIFO_MODE == NONE) or FIFOs are disabled (FCR[0] set to zero), the data in the RBR must be read before the next data arrives, otherwise it is overwritten, resulting in an over-run error. If in FIFO mode (FIFO_MODE != NONE) and FIFOs are enabled (FCR[0] set to one), this register accesses the head of the receive FIFO. If the receive FIFO is full and this register is not read before the next data character arrives, then the data already in the FIFO is preserved, but any incoming data are lost and an over-run error occurs.

UART_THR

Address: Operational Base + offset (0x0000)
Transmit Holding Register

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RW	0x00	data_output Data to be transmitted on the serial output port (sout) in UART mode or the serial infrared output (sir_out_n) in infrared mode. Data should only be written to the THR when the THR Empty (THRE) bit (LSR[5]) is set. If in non-FIFO mode or FIFOs are disabled (FCR[0] = 0) and THRE is set, writing a single character to the THR clears the THRE. Any additional writes to the THR before the THRE is set again causes the THR data to be overwritten. If in FIFO mode and FIFOs are enabled (FCR[0] = 1) and THRE is set, x number of characters of data may be written to the THR before the FIFO is full. The number x (default=16) is determined by the value of FIFO Depth that you set during configuration. Any attempt to write data when the FIFO is full results in the write data being lost.

UART_DLL
Address: Operational Base + offset (0x0000)
Divisor Latch (Low)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RW	0x00	baud_rate_divisor_L Lower 8-bits of a 16-bit, read/write, Divisor Latch register that contains the baud rate divisor for the UART. This register may only be accessed when the DLAB bit (LCR[7]) is set and the UART is not busy (USR[0] is zero). The output baud rate is equal to the serial clock (sclk) frequency divided by sixteen times the value of the baud rate divisor, as follows: baud rate = (serial clock freq) / (16 * divisor). Note that with the Divisor Latch Registers (DLL and DLH) set to zero, the baud clock is disabled and no serial communications occur. Also, once the DLH is set, at least 8 clock cycles of the slowest UART clock should be allowed to pass before transmitting or receiving data.

UART_DLH
Address: Operational Base + offset (0x0004)
Divisor Latch (High)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RW	0x00	baud_rate_divisor_H Upper 8 bits of a 16-bit, read/write, Divisor Latch register that contains the baud rate divisor for the UART.

UART_IER
Address: Operational Base + offset (0x0004)
Interrupt Enable Register

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7	RW	0x0	prog_thre_int_en Programmable THRE Interrupt Mode Enable This is used to enable/disable the generation of THRE Interrupt. 0 = disabled 1 = enabled
6:4	RO	0x0	reserved
3	RW	0x0	modem_status_int_en Enable Modem Status Interrupt. This is used to enable/disable the generation of Modem Status Interrupt. This is the fourth highest priority interrupt. 0 = disabled 1 = enabled
2	RW	0x0	receive_line_status_int_en Enable Receiver Line Status Interrupt. This is used to enable/disable the generation of Receiver Line Status Interrupt. This is the highest priority interrupt. 0 = disabled 1 = enabled
1	RW	0x0	trans_hold_empty_int_en Enable Transmit Holding Register Empty Interrupt.
0	RW	0x0	receive_data_available_int_en Enable Received Data Available Interrupt. This is used to enable/disable the generation of Received Data Available Interrupt and the Character Timeout Interrupt (if in FIFO mode and FIFOs enabled). These are the second highest priority interrupts. 0 = disabled 1 = enabled

UART_IIR
Address: Operational Base + offset (0x0008)
Interrupt Identification Register

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:6	RO	0x0	fifos_en FIFOs Enabled. This is used to indicate whether the FIFOs are enabled or disabled. 00 = disabled 11 = enabled

Bit	Attr	Reset Value	Description
5:4	RO	0x0	reserved
3:0	RO	0x0	int_id Interrupt ID This indicates the highest priority pending interrupt which can be one of the following types: 0000 = modem status 0001 = no interrupt pending 0010 = THR empty 0100 = received data available 0110 = receiver line status 0111 = busy detect 1100 = character timeout

UART_FCR
Address: Operational Base + offset (0x0008)
FIFO Control Register

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:6	WO	0x0	rcvr_trigger RCVR Trigger. This is used to select the trigger level in the receiver FIFO at which the Received Data Available Interrupt is generated. In auto flow control mode it is used to determine when the rts_n signal is de-asserted. It also determines when the dma_rx_req_n signal is asserted in certain modes of operation. The following trigger levels are supported: 00 = 1 character in the FIFO 01 = FIFO 1/4 full 10 = FIFO 1/2 full 11 = FIFO 2 less than ful
5:4	WO	0x0	tx_empty_trigger TX Empty Trigger. This is used to select the empty threshold level at which the THRE Interrupts are generated when the mode is active. It also determines when the dma_tx_req_n signal is asserted when in certain modes of operation. The following trigger levels are supported: 00 = FIFO empty 01 = 2 characters in the FIFO 10 = FIFO 1/4 full 11 = FIFO 1/2 full

Bit	Attr	Reset Value	Description
3	WO	0x0	dma_mode DMA Mode This determines the DMA signalling mode used for the dma_tx_req_n and dma_rx_req_n output signals when additional DMA handshaking signals are not selected . 0 = mode 0 1 = mode 11100 = character timeout.
2	WO	0x0	xmit_fifo_reset XMIT FIFO Reset. This resets the control portion of the transmit FIFO and treats the FIFO as empty. This also de-asserts the DMA TX request and single signals when additional DMA handshaking signals are selected . Note that this bit is 'self-clearing'. It is not necessary to clear this bit.
1	WO	0x0	rcvr_fifo_reset RCVR FIFO Reset. This resets the control portion of the receive FIFO and treats the FIFO as empty. This also de-asserts the DMA RX request and single signals when additional DMA handshaking signals are selected. Note that this bit is 'self-clearing'. It is not necessary to clear this bit.
0	WO	0x0	fifo_en FIFO Enable. FIFO Enable. This enables/disables the transmit (XMIT) and receive (RCVR) FIFOs. Whenever the value of this bit is changed both the XMIT and RCVR controller portion of FIFOs is reset.

UART_LCR

Address: Operational Base + offset (0x000c)

Line Control Register

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7	RW	0x0	div_lat_access Divisor Latch Access Bit. Writeable only when UART is not busy (USR[0] is zero), always readable. This bit is used to enable reading and writing of the Divisor Latch register (DLL and DLH) to set the baud rate of the UART. This bit must be cleared after initial baud rate setup in order to access other registers.

Bit	Attr	Reset Value	Description
6	RW	0x0	break_ctrl Break Control Bit. This is used to cause a break condition to be transmitted to the receiving device. If set to one the serial output is forced to the spacing (logic 0) state. When not in Loopback Mode, as determined by MCR[4], the sout line is forced low until the Break bit is cleared. If MCR[6] set to one, the sir_out_n line is continuously pulsed. When in Loopback Mode, the break condition is internally looped back to the receiver and the sir_out_n line is forced low.
5	RO	0x0	reserved
4	RW	0x0	even_parity_sel Even Parity Select. Writeable only when UART is not busy (USR[0] is zero), always readable. This is used to select between even and odd parity, when parity is enabled (PEN set to one). If set to one, an even number of logic 1s is transmitted or checked. If set to zero, an odd number of logic 1s is transmitted or checked.
3	RW	0x0	parity_en Parity Enable. Writeable only when UART is not busy (USR[0] is zero), always readable. This bit is used to enable and disable parity generation and detection in transmitted and received serial character respectively. 0 = parity disabled 1 = parity enabled
2	RW	0x0	stop_bits_num Number of stop bits. Writeable only when UART is not busy (USR[0] is zero), always readable. This is used to select the number of stop bits per character that the peripheral transmits and receives. If set to zero, one stop bit is transmitted in the serial data. If set to one and the data bits are set to 5 (LCR[1:0] set to zero) one and a half stop bits is transmitted. Otherwise, two stop bits are transmitted. Note that regardless of the number of stop bits selected, the receiver checks only the first stop bit. 0 = 1 stop bit 1 = 1.5 stop bits when DLS (LCR[1:0]) is zero, else 2 stop bit.

Bit	Attr	Reset Value	Description
1:0	RW	0x0	data_length_sel Data Length Select. Writeable only when UART is not busy (USR[0] is zero), always readable. This is used to select the number of data bits per character that the peripheral transmits and receives. The number of bit that may be selected areas follows: 00 = 5 bits 01 = 6 bits 10 = 7 bits 11 = 8 bits

UART_MCR
Address: Operational Base + offset (0x0010)
Modem Control Register

Bit	Attr	Reset Value	Description
31:7	RO	0x0	reserved
6	RW	0x0	sir_mode_en SIR Mode Enable. SIR Mode Enable. This is used to enable/disable the IrDA SIR Mode . 0 = IrDA SIR Mode disabled 1 = IrDA SIR Mode enabled
5	RW	0x0	auto_flow_ctrl_en Auto Flow Control Enable. 0 = Auto Flow Control Mode disabled 1 = Auto Flow Control Mode enabled
4	RW	0x0	loopback LoopBack Bit. This is used to put the UART into a diagnostic mode for test purposes.
3	RW	0x0	out2 OUT2. This is used to directly control the user-designated Output2 (out2_n) output. The value written to this location is inverted and driven out on out2_n, that is: 0 = out2_n de-asserted (logic 1) 1 = out2_n asserted (logic 0)
2	RW	0x0	out1 OUT1 This is used to directly control the user-designated Output2 (out2_n) output. The value written to this location is inverted and driven out on out2_n, that is: 1'b0: out2_n de-asserted (logic 1) 1'b1: out2_n asserted (logic 0)

Bit	Attr	Reset Value	Description
1	RW	0x0	req_to_send Request to Send. This is used to directly control the Request to Send (rts_n) output. The Request To Send (rts_n) output is used to inform the modem or data set that the UART is ready to exchange data.
0	RW	0x0	data_terminal_ready Data Terminal Ready. This is used to directly control the Data Terminal Ready (dtr_n) output. The value written to this location is inverted and driven out on dtr_n, that is: 0 = dtr_n de-asserted (logic 1) 1 = dtr_n asserted (logic 0)

UART_LSR
Address: Operational Base + offset (0x0014)
Line Status Register

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7	RO	0x0	receiver_fifo_error Receiver FIFO Error bit. This bit is relevant FIFOs are enabled (FCR[0] set to one). This is used to indicate if there is at least one parity error, framing error, or break indication in the FIFO. 0 = no error in RX FIFO 1 = error in RX FIFO
6	RO	0x0	trans_empty Transmitter Empty bit. Transmitter Empty bit. If FIFOs enabled (FCR[0] set to one), this bit is set whenever the Transmitter Shift Register and the FIFO are both empty. If FIFOs are disabled, this bit is set whenever the Transmitter Holding Register and the Transmitter Shift Register are both empty.
5	RO	0x0	trans_hold_reg_empty Transmit Holding Register Empty bit. If THRE mode is disabled (IER[7] set to zero) and regardless of FIFO's being implemented/enabled or not, this bit indicates that the THR or TX FIFO is empty. This bit is set whenever data is transferred from the THR or TX FIFO to the transmitter shift register and no new data has been written to the THR or TX FIFO. This also causes a THRE Interrupt to occur, if the THRE Interrupt is enabled. If IER[7] set to one and FCR[0] set to one respectively, the functionality is switched to indicate the transmitter FIFO is full, and no longer controls THRE interrupts, which are then controlled by the FCR[5:4] threshold setting.

Bit	Attr	Reset Value	Description
4	RO	0x0	break_int Break Interrupt bit. This is used to indicate the detection of a break sequence on the serial input data.
3	RO	0x0	framing_error Framing Error bit. This is used to indicate the occurrence of a framing error in the receiver. A framing error occurs when the receiver does not detect a valid STOP bit in the received data.
2	RO	0x0	parity_eror Parity Error bit. This is used to indicate the occurrence of a parity error in the receiver if the Parity Enable (PEN) bit (LCR[3]) is set.
1	RO	0x0	overrun_error Overrun error bit. This is used to indicate the occurrence of an overrun error. This occurs if a new data character was received before the previous data was read.
0	RO	0x0	data_ready Data Ready bit. This is used to indicate that the receiver contains at least one character in the RBR or the receiver FIFO. 0 = no data ready 1 = data ready

UART_MSR
Address: Operational Base + offset (0x0018)
Modem Status Register

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7	RO	0x0	data_carrior_detect Data Carrier Detect. This is used to indicate the current state of the modem control line dcd_n.
6	RO	0x0	ring_indicator Ring Indicator. This is used to indicate the current state of the modem control line ri_n.
5	RO	0x0	data_set_ready Data Set Ready. This is used to indicate the current state of the modem control line dsr_n.

Bit	Attr	Reset Value	Description
4	RO	0x0	clear_to_send Clear to Send. This is used to indicate the current state of the modem control line cts_n.
3	RO	0x0	delta_data_carrier_detect Delta Data Carrier Detect. This is used to indicate that the modem control line dcd_n has changed since the last time the MSR was read.
2	RO	0x0	trailing_edge_ring_indicator Trailing Edge of Ring Indicator. Trailing Edge of Ring Indicator. This is used to indicate that a change on the input ri_n (from an active-low to an inactive-high state) has occurred since the last time the MSR was read.
1	RO	0x0	delta_data_set_ready Delta Data Set Ready. This is used to indicate that the modem control line dsr_n has changed since the last time the MSR was read.
0	RO	0x0	delta_clear_to_send Delta Clear to Send. This is used to indicate that the modem control line cts_n has changed since the last time the MSR was read.

UART_SCR
Address: Operational Base + offset (0x001c)
Scratchpad Register

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RW	0x00	temp_store_space This register is for programmers to use as a temporary storage space.

UART_SRBR
Address: Operational Base + offset (0x0030)
Shadow Receive Buffer Register

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved

Bit	Attr	Reset Value	Description
7:0	RO	0x00	shadow_rbr This is a shadow register for the RBR and has been allocated sixteen 32-bit locations so as to accommodate burst accesses from the master. This register contains the data byte received on the serial input port (sin) in UART mode or the serial infrared input (sir_in) in infrared mode. The data in this register is valid only if the Data Ready (DR) bit in the Line status Register (LSR) is set. If FIFOs are disabled (FCR[0] set to zero), the data in the RBR must be read before the next data arrives, otherwise it is overwritten, resulting in an overrun error. If FIFOs are enabled (FCR[0] set to one), this register accesses the head of the receive FIFO. If the receive FIFO is full and this register is not read before the next data character arrives, then the data already in the FIFO are preserved, but any incoming data is lost. An overrun error also occurs.

UART_STHR
Address: Operational Base + offset (0x006c)
Shadow Transmit Holding Register

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RO	0x00	shadow_thr This is a shadow register for the THR.

UART_FAR
Address: Operational Base + offset (0x0070)
FIFO Access Register

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	fifo_access_test_en This register is use to enable a FIFO access mode for testing, so that the receive FIFO can be written by the master and the transmit FIFO can be read by the master when FIFOs are implemented and enabled. When FIFOs are not enabled it allows the RBR to be written by the master and the THR to be read by the master. 0 = FIFO access mode disabled 1 = FIFO access mode enabled

UART_TFR
Address: Operational Base + offset (0x0074)
Transmit FIFO Read

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved

Bit	Attr	Reset Value	Description
7:0	RO	0x00	trans_fifo_read Transmit FIFO Read. These bits are only valid when FIFO access mode is enabled (FAR[0] is set to one).When FIFOs are implemented and enabled, reading this register gives the data at the top of the transmit FIFO. Each consecutive read pops the transmit FIFO and gives the next data value that is currently at the top of the FIFO.

UART_RFW
Address: Operational Base + offset (0x0078)
Receive FIFO Write

Bit	Attr	Reset Value	Description
31:10	RO	0x0	reserved
9	WO	0x0	receive_fifo_framing_error Receive FIFO Framing Error. These bits are only valid when FIFO access mode is enabled (FAR[0] is set to one).
8	WO	0x0	receive_fifo_parity_error Receive FIFO Parity Error. These bits are only valid when FIFO access mode is enabled (FAR[0] is set to one).
7:0	WO	0x00	receive_fifo_write Receive FIFO Write Data. These bits are only valid when FIFO access mode is enabled (FAR[0] is set to one). When FIFOs are enabled, the data that is written to the RFWD is pushed into the receive FIFO. Each consecutive write pushes the new data to the next write location in the receive FIFO. When FIFOs not enabled, the data that is written to the RFWD is pushed into the RBR.

UART_USR
Address: Operational Base + offset (0x007c)
UART Status Register

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4	RO	0x0	receive_fifo_full Receive FIFO Full. This is used to indicate that the receive FIFO is completely full. 0 = Receive FIFO not full 1 = Receive FIFO Full This bit is cleared when the RX FIFO is no longer full.

Bit	Attr	Reset Value	Description
3	RO	0x0	receive_fifo_not_empty Receive FIFO Not Empty. This is used to indicate that the receive FIFO contains one or more entries. 0 = Receive FIFO is empty 1 = Receive FIFO is not empty This bit is cleared when the RX FIFO is empty.
2	RO	0x0	trasn_fifo_empty Transmit FIFO Empty. This is used to indicate that the transmit FIFO is completely empty. 0 = Transmit FIFO is not empty 1 = Transmit FIFO is empty This bit is cleared when the TX FIFO is no longer empty
1	RO	0x0	trans_fifo_not_full Transmit FIFO Not Full. This is used to indicate that the transmit FIFO in not full. 0 = Transmit FIFO is full 1 = Transmit FIFO is not full This bit is cleared when the TX FIFO is full.
0	RO	0x0	uart_busy UART Busy. UART Busy. This is indicates that a serial transfer is in progress, when cleared indicates that the UART is idle or inactive. 0 = UART is idle or inactive 1 = UART is busy (actively transferring data)

UART_TFL
Address: Operational Base + offset (0x0080)
Transmit FIFO Level

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4:0	RW	0x00	trans_fifo_level Transmit FIFO Level. This is indicates the number of data entries in the transmit FIFO.

UART_RFL
Address: Operational Base + offset (0x0084)
Receive FIFO Level

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4:0	RO	0x00	receive_fifo_level Receive FIFO Level. This is indicates the number of data entries in the receive FIFO.

UART_SRR
Address: Operational Base + offset (0x0088)
Software Reset Register

Bit	Attr	Reset Value	Description
31:3	RO	0x0	reserved
2	WO	0x0	xmit_fifo_reset XMIT FIFO Reset. This is a shadow register for the XMIT FIFO Reset bit (FCR[2]).
1	WO	0x0	rcvr_fifo_reset RCVR FIFO Reset. This is a shadow register for the RCVR FIFO Reset bit (FCR[1]).
0	WO	0x0	uart_reset UART Reset. This asynchronously resets the UART and synchronously removes the reset assertion. For a two clock implementation both pclk and sclk domains are reset.

UART_SRTS
Address: Operational Base + offset (0x008c)
Shadow Request to Send

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	shadow_req_to_send Shadow Request to Send. This is a shadow register for the RTS bit (MCR[1]), this can be used to remove the burden of having to performing a read-modify-write on the MCR.

UART_SBCR
Address: Operational Base + offset (0x0090)
Shadow Break Control Register

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	shadow_break_ctrl Shadow Break Control Bit. This is a shadow register for the Break bit (LCR[6]), this can be used to remove the burden of having to performing a read modify write on the LCR.

UART_SDMAM
Address: Operational Base + offset (0x0094)
Shadow DMA Mode

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved

Bit	Attr	Reset Value	Description
0	RW	0x0	shadow_dma_mode Shadow DMA Mode. This is a shadow register for the DMA mode bit (FCR[3]).

UART_SFE
Address: Operational Base + offset (0x0098)
Shadow FIFO Enable

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	shadow_fifo_en Shadow FIFO Enable. Shadow FIFO Enable. This is a shadow register for the FIFO enable bit (FCR[0]).

UART_SRT
Address: Operational Base + offset (0x009c)
Shadow RCVR Trigger

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	shadow_rcvr_trigger Shadow RCVR Trigger. This is a shadow register for the RCVR trigger bits (FCR[7:6]).

UART_STET
Address: Operational Base + offset (0x00a0)
Shadow TX Empty Trigger

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	shadow_tx_empty_trigger Shadow TX Empty Trigger. This is a shadow register for the TX empty trigger bits (FCR[5:4]).

UART_HTX
Address: Operational Base + offset (0x00a4)
Halt TX

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	halt_tx_en This register is use to halt transmissions for testing, so that the transmit FIFO can be filled by the master when FIFOs are implemented and enabled. 0 = Halt TX disabled 1 = Halt TX enabled

UART_DMASA
Address: Operational Base + offset (0x00a8)
DMA Software Acknowledge

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	WO	0x0	dma_software_ack This register is use to perform a DMA software acknowledge if a transfer needs to be terminated due to an error condition.

UART_CPR
Address: Operational Base + offset (0x00f4)
Component Parameter Register
UART_CPR is UART0's own unique register

Bit	Attr	Reset Value	Description
31:24	RO	0x0	reserved
23:16	RO	0x00	FIFO_MODE 0x00 = 0 0x01 = 16 0x02 = 32 to 0x80 = 2048 0x81- 0xff = reserved
15:14	RO	0x0	reserved
13	RO	0x0	DMA_EXTRA 0 = FALSE 1 = TRUE
12	RO	0x0	UART_ADD_ENCODED_PARAMS 0 = FALSE 1 = TRUE
11	RO	0x0	SHADOW 0 = FALSE 1 = TRUE
10	RO	0x0	FIFO_STAT 0 = FALSE 1 = TRUE
9	RO	0x0	FIFO_ACCESS 0 = FALSE 1 = TRUE
8	RO	0x0	NEW_FEAT 0 = FALSE 1 = TRUE
7	RO	0x0	SIR_LP_MODE 0 = FALSE 1 = TRUE

Bit	Attr	Reset Value	Description
6	RO	0x0	SIR_MODE 0 = FALSE 1 = TRUE
5	RO	0x0	THRE_MODE 0 = FALSE 1 = TRUE
4	RO	0x0	AFCE_MODE 0 = FALSE 1 = TRUE
3:2	RO	0x0	reserved
1:0	RO	0x0	APB_DATA_WIDTH 00 = 8 bits 01 = 16 bits 10 = 32 bits 11 = reserved

UART_UCV
Address: Operational Base + offset (0x00f8)
UART Component Version

Bit	Attr	Reset Value	Description
31:0	RO	0x0330372a	ver ASCII value for each number in the version

UART_CTR
Address: Operational Base + offset (0x00fc)
Component Type Register

Bit	Attr	Reset Value	Description
31:0	RO	0x44570110	peripheral_id This register contains the peripherals identification code.

20.5 Interface Description

Table 20-1 UART Interface Description

Module pin	Dir	Pad name	IOMUX
UART0 Interface			
uart0_sin	I	IO_UART0BTsin_WIFIBTgpio2c0	GRF_GPIO2C_IOMUX[1:0]=2'b01
uart0_sout	O	IO_UART0BTsout_WIFIBTgpio2c1	GRF_GPIO2C_IOMUX[3:2]=2'b01
uart0_cts_n	I	IO_UART0BTctsn_WIFIBTgpio2c2	GRF_GPIO2C_IOMUX[5:4]=2'b01
uart0_rts_n	O	IO_UART0BTrtsn_WIFIBTgpio2c3	GRF_GPIO2C_IOMUX[7:6]=2'b01
UART1 Interface			
uart1_sin	I	IO_MACtxen_UART1BBsin_GMACgpio3b4	GRF_GPIO3B_IOMUX[9:8]=2'b10
uart1_s	O	IO_MACmdio_UART1BBsout_GMACgpio	GRF_GPIO3B_IOMUX[11:10]=

Module pin	Dir	Pad name	IOMUX
out		3b5	2’b10
UART2A Interface			
uart2a_sin	I	IO_SDMMCdata0_UART2DBGAsin_SDMMCgpio4b0	GRF_GPIO4B_IOMUX[1:0]=2’b10
uart2a_sout	O	IO_SDMMCdata1_UART2DBGAsout_HDCPJTAGtrstn_SDMMCgpio4b1	GRF_GPIO4B_IOMUX[3:2]=2’b10
UART2B Interface			
uart2b_sin	I	IO_I2C3HDMIsda_UART2DBGBsin_HDMII2Csda_GPIO1830gpio4c0	GRF_GPIO4C_IOMUX[1:0]=2’b10
uart2b_sout	O	IO_I2C3HDMIscl_UART2DBGBsout_HDMII2Cscl_GPIO1830gpio4c1	GRF_GPIO4C_IOMUX[3:2]=2’b10
UART2C Interface			
uart2c_sin	I	IO_UART2DBGCsin_UARTHDCPsin_GPIO1830gpio4c3	GRF_GPIO4C_IOMUX[7:6]=2’b10
uart2c_sout	O	IO_UART2DBGCsout_UARTHDCPsout_GPIO1830gpio4c4	GRF_GPIO4C_IOMUX[9:8]=2’b10
UART3 Interface			
uart3_sin	I	IO_MACrxclk_UART3GPSsin_GMACgpio3b6	GRF_GPIO3B_IOMUX[13:12]=2’b10
uart3_sout	O	IO_MACcrs_UART3GPSsout_CIFclkoutb_GMACgpio3b7	GRF_GPIO3B_IOMUX[15:14]=2’b10
uart3_cts_n	I	IO_MACcol_UART3GPSctsn_SPDIFtxb_GMACgpio3c0	GRF_GPIO3C_IOMUX[1:0]=2’b10
uart3_rts_n	O	IO_MACtxclk_UART3GPSrtsn_GMACgpio3c1	GRF_GPIO3C_IOMUX[3:2]=2’b10
UART4 Interface			
uart4_sin	I	IO_UART4M0sin_SPI1ECrxd_PMU1830gpio1a7	PMUGRF_GPIO1A_IOMUX[15:14]=2’b01
uart4_sout	O	IO_UART4M0sout_SPI1ECtxd_PMU1830gpio1b0	PMUGRF_GPIO1B_IOMUX[1:0]=2’b01

The I/O interface of UART1 can be chosen by setting GRF_CON_IOMUX[11](uart1sel) bit, if this bit is set to 1, UART1 uses the UART11 I/O interface. The I/O interface of UART2 can be chosen by setting GRF_CON_IOMUX[8](uart2sel) bit, if this bit is set to 1, UART2 uses the UART21 I/O interface.

20.6 Application Notes

20.6.1 None FIFO Mode Transfer Flow

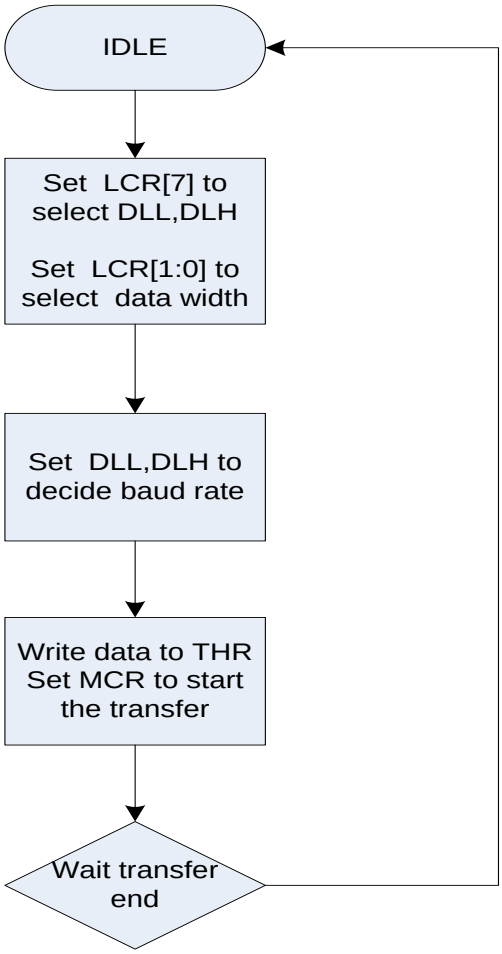


Fig. 20-8 UART none fifo mode

20.6.2 FIFO Mode Transfer Flow

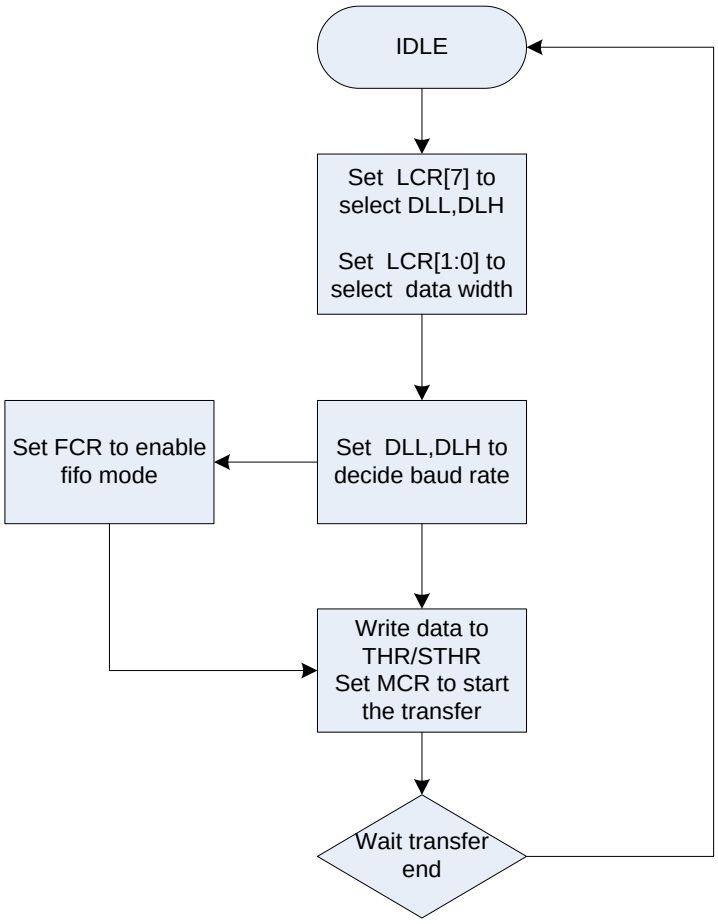


Fig. 20-9 UART fifo mode

The UART is an APB slave performing:
Serial-to-parallel conversion on data received from a peripheral device.
Parallel-to-serial conversion on data transmitted to the peripheral device.
The CPU reads and writes data and control/status information through the APB interface.
The transmitting and receiving paths are buffered with internal FIFO memories enabling up to 64-bytes to be stored independently in both transmit and receive modes. A baud rate generator can generate a common transmit and receive internal clock input. The baud rates will depend on the internal clock frequency. The UART will also provide transmit, receive and exception interrupts to system. A DMA interface is implemented for improving the system performance.

20.6.3 Baud Rate Calculation

UART clock generation

The following figures shows the UART clock generation.
UART0, UART1, UART2, UART3 source clocks can be selected from three PLL outputs (CODEC PLL/GENERAL PLL/USBPHY_480M). UART4 source clocks can be selected from only one PLL outputs (PMU PLL). UART clocks can be generated by 1 to 64 division of its source clock, or can be fractionally divided again, or be provided by XIN24M.

UART baud rate configuration

The following table provides some reference configuration for different UART baud rates.

Table 20-2 UART baud rate configuration

Baud Rate	Reference Configuration
115.2 Kbps	Configure GENERAL PLL to get 648MHz clock output; Divide 648MHz clock by 1152/50625 to get 14.7456MHz clock; Configure UART_DLL to 8.
460.8 Kbps	Configure GENERAL PLL to get 648MHz clock output; Divide 648MHz clock by 1152/50625 to get 14.7456MHz clock; Configure UART_DLL to 2.
921.6 Kbps	Configure GENERAL PLL to get 648MHz clock output;

Baud Rate	Reference Configuration
	Divide 648MHz clock by 1152/50625 to get 14.7456MHz clock; Configure UART_DLL to 1.
1.5 Mbps	Choose GENERAL PLL to get 384MHz clock output; Divide 384MHz clock by 16 to get 24MHz clock; Configure UART_DLL to 1
3 Mbps	Choose GENERAL PLL to get 384MHz clock output; Divide 384MHz clock by 8 to get 48MHz clock; Configure UART_DLL to 1
4 Mbps	Configure GENERAL PLL to get 384MHz clock output; Divide 384MHz clock by 6 to get 64MHz clock; Configure UART_DLL to 1

1.6.4 CTS_n and RTS_n Polarity Configurable

The polarity of cts_n and rts_n ports can be configured by GRF registers.

- grf_uart_cts_sel[*] used to configure the polarity of cts_n. Every bit for one UART.
- grf_uart_rts_sel[*] used to configure the polarity of rts_n. Every bit for one UART.
- When grf_uart_cts_sel[*] is configured as 1'b1, cts_n is high active. Otherwise, low active.
- When grf_uart_rts_sel[*] is configured as 1'b1, rts_n is high active. Otherwise, low active.

Chapter 21 GPIO

21

21.1 Overview

GPIO is a programmable General Purpose Programming I/O peripheral. This component is an APB slave device. GPIO controls the output data and direction of external I/O pads. It also can read back the data on external pads using memory-mapped registers.
GPIO supports the following features:

- 32 bits APB bus width
- 32 independently configurable signals
- Separate data registers and data direction registers for each signal
- Software control for each signal, or for each bit of each signal
- Configurable interrupt mode

21.2 Block Diagram

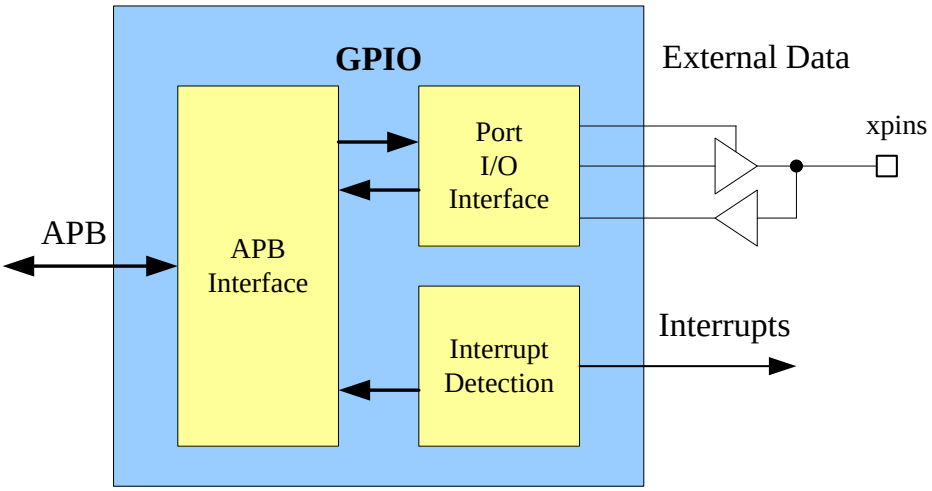


Fig. 21-1 GPIO block diagram

Block descriptions:

APB Interface

The APB Interface implements the APB slave operation. Its data bus width is 32 bits.

Port I/O Interface

External data Interface to or from I/O pads.

Interrupt Detection

Interrupt interface to or from interrupt controller.

21.3 Function Description

21.3.1 Operation

Control Mode (software)

Under software control, the data and direction control for the signal are sourced from the data register (GPIO_SWPORTA_DR) and direction control register (GPIO_SWPORTA_DDR). The direction of the external I/O pad is controlled by a write to the Porta data direction register (GPIO_SWPORTA_DDR). The data written to this memory-mapped register gets mapped onto an output signal, GPIO_PORTA_DDR, of the GPIO peripheral. This output signal controls the direction of an external I/O pad.
The data written to the Porta data register (GPIO_SWPORTA_DR) drives the output buffer of the I/O pad. External data are input on the external data signal, GPIO_EXT_PORTA. Reading the external signal register (GPIO_EXT_PORTA) shows the value on the signal, regardless of the direction. This register is read-only, meaning that it cannot be written from the APB software interface.

Reading External Signals

The data on the GPIO_EXT_PORTA external signal can always be read. The data on the external GPIO signal is read by an APB read of the memory-mapped register, GPIO_EXT_PORTA.
An APB read to the GPIO_EXT_PORTA register yields a value equal to that which is on the GPIO_EXT_PORTA signal.

Interrupts

Port A can be programmed to accept external signals as interrupt sources on any of the bits of the signal. The type of interrupt is programmable with one of the following settings:

- Active-high and level
- Active-low and level
- Rising edge
- Falling edge

The interrupts can be masked by programming the GPIO_INTMASK register. The interrupt status can be read before masking (called raw status) and after masking.
The interrupts are combined into a single interrupt output signal, which has the same polarity as the individual interrupts. In order to mask the combined interrupt, all individual interrupts have to be masked. The single combined interrupt does not have its own mask bit.
Whenever Port A is configured for interrupts, the data direction must be set to Input. If the data direction register is reprogrammed to Output, then any pending interrupts are not lost. However, no new interrupts are generated.
For edge-detected interrupts, the ISR can clear the interrupt by writing a 1 to the GPIO_PORTA_EOI register for the corresponding bit to disable the interrupt. This write also clears the interrupt status and raw status registers. Writing to the GPIO_PORTA_EOI register has no effect on level-sensitive interrupts. If level-sensitive interrupts cause the processor to interrupt, then the ISR can poll the GPIO_INT_RAWSTATUS register until the interrupt source disappears, or it can write to the GPIO_INTMASK register to mask the interrupt before exiting the ISR. If the ISR exits without masking or disabling the interrupt prior to exiting, then the level-sensitive interrupt repeatedly requests an interrupt until the interrupt is cleared at the source.

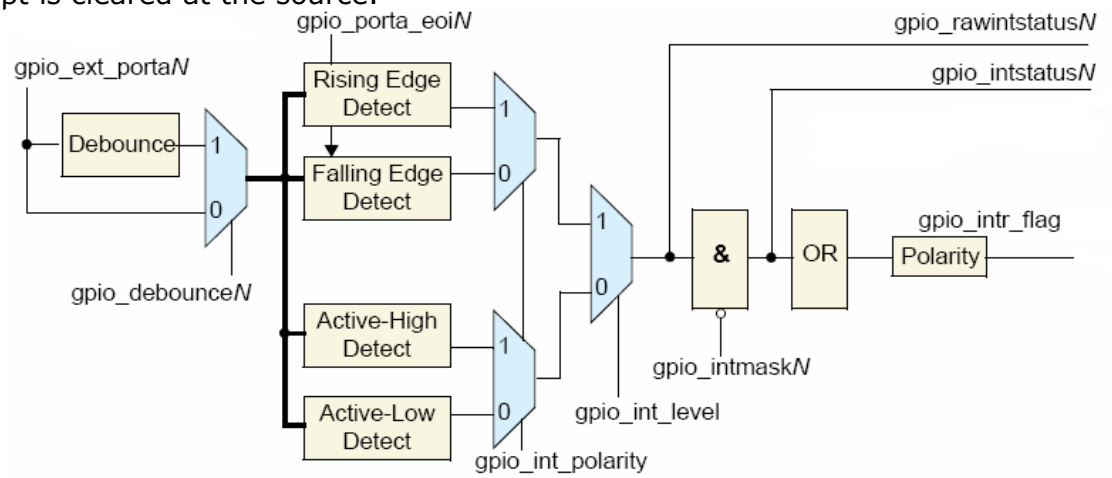


Fig. 21-2 GPIO Interrupt RTL Block Diagram

Debounce operation

Port A has been configured to include the debounce capability interrupt feature. The external signal can be debounced to remove any spurious glitches that are less than one period of the external debouncing clock.
When input interrupt signals are debounced using a debounce clock (pclk), the signals must be active for a minimum of two cycles of the debounce clock to guarantee that they are registered. Any input pulse widths less than a debounce clock period are bounced. A pulse width between one and two debounce clock widths may or may not propagate, depending on its phase relationship to the debounce clock. If the input pulse spans two rising edges of the debounce clock, it is registered. If it spans only one rising edge, it is not registered.

Synchronization of Interrupt Signals to the System Clock

Interrupt signals are internally synchronized to pclk. Synchronization to pclk must occur for

edge-detect signals. With level-sensitive interrupts, synchronization is optional and under software control (GPIO_LS_SYNC).

21.3.2 Programming

Programming Considerations

- Reading from an unused location or unused bits in a particular register always returns zeros. There is no error mechanism in the APB.
- Programming the GPIO registers for interrupt capability, edge-sensitive or level-sensitive interrupts, and interrupt polarity should be completed prior to enabling the interrupts on Port A in order to prevent spurious glitches on the interrupt lines to the interrupt controller.
- Writing to the interrupt clear register clears an edge-detected interrupt and has no effect on a level-sensitive interrupt.

9 GPIOs' hierarchy in the chip

GPIO0, GPIO1, GPIO2 are in PD_PERI subsystem.

21.4 Register Description

This section describes the control/status registers of the design. Software should read and write these registers using 32-bits accesses. There are 3 GPIOs (GPIO0 ~ GPIO2), and each of them has same register group. Therefore, 3 GPIOs' register groups have 3 different base addresses.

21.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
GPIO_SWPORTA_DR	0x0000	W	0x00000000	Port A data register
GPIO_SWPORTA_DDR	0x0004	W	0x00000000	Port A data direction register
GPIO_INTEN	0x0030	W	0x00000000	Interrupt enable register
GPIO_INTMASK	0x0034	W	0x00000000	Interrupt mask register
GPIO_INTTYPE_LEVEL	0x0038	W	0x00000000	Interrupt level register
GPIO_INT_POLARITY	0x003c	W	0x00000000	Interrupt polarity register
GPIO_INT_STATUS	0x0040	W	0x00000000	Interrupt status of port A
GPIO_INT_RAWSTATUS	0x0044	W	0x00000000	Raw Interrupt status of port A
GPIO_DEBOUNCE	0x0048	W	0x00000000	Debounce enable register
GPIO_PORTA_EOI	0x004c	W	0x00000000	Port A clear interrupt register
GPIO_EXT_PORTA	0x0050	W	0x00000000	Port A external port register
GPIO_LS_SYNC	0x0060	W	0x00000000	Level_sensitive synchronization enable register

*Notes:Size:**B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access*

21.4.2 Detail Register Description

GPIO_SWPORTA_DR

Address: Operational Base + offset (0x0000)

Port A data register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	gpio_swporta_dr Values written to this register are output on the I/O signals for Port A if the corresponding data direction bits for Port A are set to Output mode. The value read back is equal to the last value written to this register.

GPIO_SWPORTA_DDR
Address: Operational Base + offset (0x0004)
Port A data direction register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	gpio_swporta_ddr Values written to this register independently control the direction of the corresponding data bit in Port A. 0: Input (default) 1: Output

GPIO_INTEN
Address: Operational Base + offset (0x0030)
Interrupt enable register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	gpio_int_en Allows each bit of Port A to be configured for interrupts. Whenever a 1 is written to a bit of this register, it configures the corresponding bit on Port A to become an interrupt; otherwise, Port A operates as a normal GPIO signal. Interrupts are disabled on the corresponding bits of Port A if the corresponding data direction register is set to Output. 0: Configure Port A bit as normal GPIO signal (default) 1: Configure Port A bit as interrupt

GPIO_INTMASK
Address: Operational Base + offset (0x0034)
Interrupt mask register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	gpio_int_mask Controls whether an interrupt on Port A can create an interrupt for the interrupt controller by not masking it. Whenever a 1 is written to a bit in this register, it masks the interrupt generation capability for this signal; otherwise interrupts are allowed through. 0: Interrupt bits are unmasked (default) 1: Mask interrupt

GPIO_INTTYPE_LEVEL
Address: Operational Base + offset (0x0038)
Interrupt level register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	gpio_inttype_level Controls the type of interrupt that can occur on Port A. 0: Level-sensitive (default) 1: Edge-sensitive

GPIO_INT_POLARITY
Address: Operational Base + offset (0x003c)
Interrupt polarity register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	gpio_int_polarity Controls the polarity of edge or level sensitivity that can occur on input of Port A. 0: Active-low (default) 1: Active-high

GPIO_INT_STATUS
Address: Operational Base + offset (0x0040)
Interrupt status of port A

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	gpio_int_status Interrupt status of Port A

GPIO_INT_RAWSTATUS
Address: Operational Base + offset (0x0044)
Raw Interrupt status of port A

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	gpio_int_rawstatus Raw interrupt of status of Port A (premasking bits)

GPIO_DEBOUNCE
Address: Operational Base + offset (0x0048)
Debounce enable register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	gpio_debounce Controls whether an external signal that is the source of an interrupt needs to be debounced to remove any spurious glitches. Writing a 1 to a bit in this register enables the debouncing circuitry. A signal must be valid for two periods of an external clock before it is internally processed. 0: No debounce (default) 1: Enable debounce

GPIO_PORTA_EOI
Address: Operational Base + offset (0x004c)
Port A clear interrupt register

Bit	Attr	Reset Value	Description
31:0	WO	0x00000000	gpio_porta_eoi Controls the clearing of edge type interrupts from Port A. When a 1 is written into a corresponding bit of this register, the interrupt is cleared. All interrupts are cleared when Port A is not configured for interrupts. 0: No interrupt clear (default) 1: Clear interrupt

GPIO_EXT_PORTA
Address: Operational Base + offset (0x0050)
Port A external port register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	gpio_ext_porta When Port A is configured as Input, then reading this location reads the values on the signal. When the data direction of Port A is set as Output, reading this location reads the data register for Port A.

GPIO_LS_SYNC
Address: Operational Base + offset (0x0060)
Level_sensitive synchronization enable register

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	gpio_ls_sync Writing a 1 to this register results in all level-sensitive interrupts being synchronized to pclk_intr. 0: No synchronization to pclk_intr (default) 1: Synchronize to pclk_intr

21.5 Interface Description

Table 21-1 GPIO interface description

Module Pin	Dir	Pad Name	IOMUX Setting
GPIO0 Interface			
gpio0_porta[6:0]	I/O	GPIO0_A[6:0]	GRF_GPIO0A_IOMUX[13:0]=14'h0
gpio0_porta[15:8]	I/O	GPIO0_B[7:0]	GRF_GPIO0B_IOMUX[15:0]=16'h0
gpio0_porta[22:16]	I/O	GPIO0_C[6:0]	GRF_GPIO0C_IOMUX[11:0]=12'h0
gpio0_porta[31:24]	I/O	GPIO0_D[7:0]	GRF_GPIO0D_IOMUX[15:4]=16'h0
GPIO1 Interface			
gpio1_porta[7:0]	I/O	GPIO1_A[7:0]	GRF_GPIO1A_IOMUX[15:0]=16'h0
gpio1_porta[9:8]	I/O	GPIO1_B[1:0]	GRF_GPIO1B_IOMUX[3:0]=4'h0
GPIO2 Interface			
gpio2_porta[7:0]	I/O	GPIO2_A[7:0]	GRF_GPIO2A_IOMUX[15:0]=16'h0
gpio2_porta[15:8]	I/O	GPIO2_B[7:0]	GRF_GPIO2B_IOMUX[15:0]=16'h0

Module Pin	Dir	Pad Name	IOMUX Setting
gpio2_porta[23:16]	I/O	GPIO2_C[7:0]	GRF_GPIO2C_IOMUX[15:0]=16'h0
gpio2_porta[31:24]	I/O	GPIO2_D[7:0]	GRF_GPIO2D_IOMUX[15:0]=16'h0
GPIO3 Interface			
gpio3_porta[4:0]	I/O	GPIO3_A[4:0]	GRF_GPIO3A_IOMUX[9:0]=10'h0
gpio3_porta[15:8]	I/O	GPIO3_B[7:0]	GRF_GPIO3B_IOMUX[15:0]=16'h0
gpio3_porta[22:16]	I/O	GPIO3_C[6:0]	GRF_GPIO3C_IOMUX[13:0]=14'h0
GPIO4 Interface			
gpio4_porta[23:16]	I/O	GPIO4_C[7:0]	GRF_GPIO4C_IOMUX[15:0]=16'h0
gpio4_porta[26:24]	I/O	GPIO4_D[2:0]	GRF_GPIO4D_IOMUX[5:0]=6'h0

21.6 Application Notes

Steps to set GPIO's direction

- Write GPIO_SWPORT_DDR[x] as 1 to set this gpio as output direction and Write GPIO_SWPORT_DDR[x] as 0 to set this gpio as input direction.
- Default GPIO's direction is input direction.

Steps to set GPIO's level

- Write GPIO_SWPORT_DDR[x] as 1 to set this gpio as output direction.
- Write GPIO_SWPORT_DR[x] as v to set this GPIO's value.

Steps to get GPIO's level

- Write GPIO_SWPORT_DDR[x] as 0 to set this gpio as input direction.
- Read from GPIO_EXT_PORT[x] to get GPIO's value

Steps to set GPIO as interrupt source

- Write GPIO_SWPORT_DDR[x] as 0 to set this gpio as input direction.
- Write GPIO_INTTYPE_LEVEL[x] as v1 and write GPIO_INT_POLARITY[x] as v2 to set interrupt type
- Write GPIO_INTEN[x] as 1 to enable GPIO's interrupt

Note: Please switch iomux to GPIO mode first!

Chapter 22 I2C Interface

22

22.1 Overview

The Inter-Integrated Circuit (I2C) is a two wired (SCL and SDA), bi-directional serial bus that provides an efficient and simple method of information exchange between devices. This I2C bus controller supports master mode acting as a bridge between AMBA protocol and generic I2C bus system.

I2C Controller supports the following features:

- Item Compatible with I2C-bus
- AMBA APB slave interface
- Supports master mode of I2C bus
- Software programmable clock frequency and transfer rate up to 400Kbit/sec
- Supports 7 bits and 10 bits addressing modes
- Interrupt or polling driven multiple bytes data transfer
- Clock stretching and wait state generation

22.2 Block Diagram

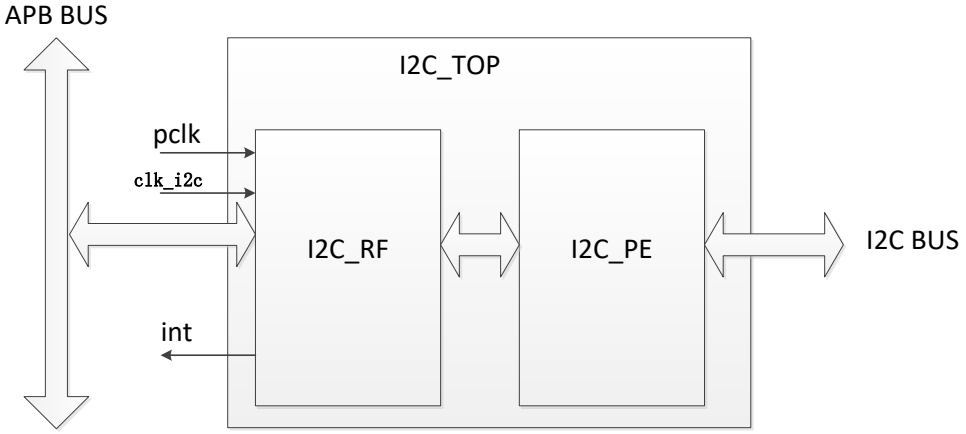


Fig. 22-1 I2C architecture

22.2.1 I2C_RF

I2C_RF module is used to control the I2C controller operation by the host with APB interface. It implements the register set and the interrupt functionality. The CSR component operates synchronously with the pclk clock.

22.2.2 I2C_PE

I2C_PE module implements the I2C master operation for transmit data to and receive data from other I2C devices. The I2C master controller operates synchronously with the clk_i2c.

22.2.3 I2C_TOP

I2C_TOP module is the top module of the I2C controller.

22.3 Function Description

This chapter provides a description about the functions and behavior under various conditions.

The I2C controller supports only Master function. It supports the 7-bits/10-bits addressing mode and support general call address. The maximum clock frequency and transfer rate can be up to 1Mbit/sec.

The operations of I2C controller is divided to 2 parts and described separately: initialization and master mode programming.

22.3.1 Initialization

The I2C controller is based on AMBA APB bus architecture and usually is part of a SOC. So before I2C operates, some system setting and configuration must be conformed, which

includes:

- I2C interrupt connection type: CPU interrupt scheme should be considered. If the I2C interrupt is connected to extra Interrupt Controller module, we need decide the INTC vector.
- I2C Clock Rate: The I2C controller uses the APB clock/clk_i2c as the working clock. The APB clock will determine the I2C bus clock, clk_i2c is the function clk, up to 200MHz. The correct register setting is subject to the system requirement.

22.3.2 Master Mode Programming

- SCL Clock
  When the I2C controller is programmed in Master mode, the SCL frequency is determined by I2C_CLKDIV register. The SCL frequency is calculated by the following formula:
  SCL Divisor = 8*(CLKDIVL + 1 + CLKDIVH + 1)
  SCL = clk_i2c/ SCLK Divisor

- Data Receiver Register Access
  When the I2C controller received MRXCNT bytes data， CPU can get the data through register RXDATA0 ~ RXDATA7. The controller can receive up to 32 bytes' data in one transaction.
  When MRXCNT register is written, the I2C controller will start to drive SCL to receive data.

- Transmit Transmitter Register
  Data to transmit are written to TXDATA0~7 by CPU. The controller can transmit up to 32 bytes' data in one transaction. The lower byte will be transmitted first.
  When MTXCNT register is written, the I2C controller will start to transmit data.

- Start Command
  Write 1 to I2C_CON[3], the controller will send I2C start command.

- Stop Command
  Write 1 to I2C_CON[4], the controller will send I2C stop command

- I2C Operation mode
  There are four i2c operation modes.
  - When I2C_CON[2:1] is 2'b00, the controller transmit all valid data in TXDATA0~TXDATA7 byte by byte. The controller will transmit lower byte first.
  - When I2C_CON[2:1] is 2'b01,the controller will transmit device address in MRXADDR first (Write/Read bit = 0) and then transmit device register address in MRXRADDR. After that, the controller will assert restart signal and resend MRXADDR (Write/Read bit = 1). At last, the controller enter receive mode.
  - When I2C_CON[2:1] is 2'b10, the controller is in receive mode, it will trigger clock to read MRXCNT byte data.
  - When I2C_CON[2:1] is 2'b11, the controller will transmit device address in MRXADDR first (Write/Read bit = 1) and then transmit device register address in MRXRADDR . After that, the controller will assert restart signal and resend MRXADDR (Write/Read bit = 1). At last, the controller enter receive mode.

- Read/Write Command
  - When I2C_OPMODE(I2C_CON[2:1]) is 2'b01 or 2'b11, the Read/Write command bit is decided by controller itself.
  - In RX only mode (I2C_CON[2:1] is 2'b10), the Read/Write command bit is decided by MRXADDR[0].
  - In TX only mode (I2C_CON[[2:1] is 2'b00), the Read/Write command bit is decided by TXDATA[0].

- Master Interrupt Condition

There are 7 interrupt bits in I2C_ISR register related to master mode.

- Byte transmitted finish interrupt (Bit 0): The bit is asserted when Master completed transmitting a byte.
- Byte received finish interrupt (Bit 1): The bit is asserted when Master completed receiving a byte.
- MTXCNT bytes data transmitted finish interrupt (Bit 2): The bit is asserted when Master completed transmitting MTXCNT bytes.
- MRXCNT bytes data received finish interrupt (Bit 3): The bit is asserted when Master completed receiving MRXCNT bytes.
- Start interrupt (Bit 4): The bit is asserted when Master finished asserting start command to I2C bus.
- Stop interrupt (Bit 5): The bit is asserted when Master finished asserting stop command to I2C bus.
- NAK received interrupt (Bit 6): The bit is asserted when Master received a NAK handshake.

- Last byte acknowledge control
  - If I2C_CON[5] is 1, the I2C controller will transmit NAK handshake to slave when the last byte received in RX only mode.
  - If I2C_CON[5] is 0, the I2C controller will transmit ACK handshake to slave when the last byte received in RX only mode.

- How to handle NAK handshake received
  - If I2C_CON[6] is 1, the I2C controller will stop all transactions when NAK handshake received. And the software should take responsibility to handle the problem.
  - If I2C_CON[6] is 0, the I2C controller will ignore all NAK handshake received.

- I2C controller data transfer waveform
  - Bit transferring
    - Data Validity
      The SDA line must be stable during the high period of SCL, and the data on SDA line can only be changed when SCL is in low state.

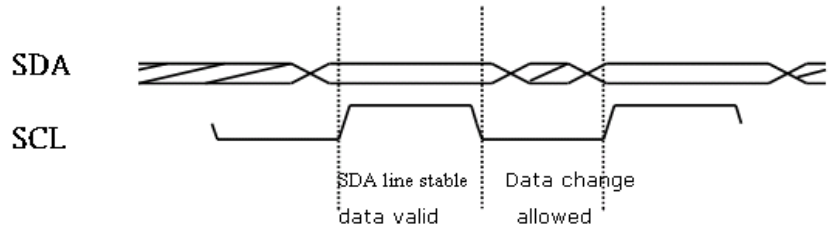


Fig. 22-2 I2C DATA Validity

    - START and STOP conditions
      START condition occurs when SDA goes low while SCL is in high period. STOP condition is generated when SDA line goes high while SCL is in high state.

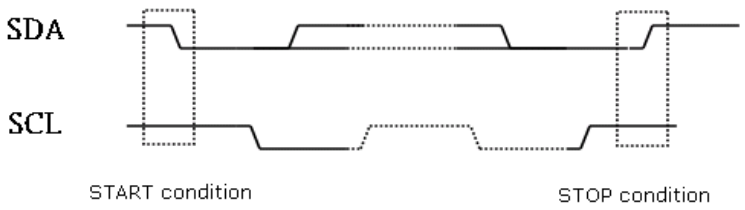


Fig. 22-3 I2C Start and stop conditions

    - Data transfer
      - Acknowledge

After a byte of data transferring (clocks labeled as 1~8), in 9th clock the receiver must assert an ACK signal on SDA line, if the receiver pulls SDA line to low, it means "ACK", on the contrary, it's "NOT ACK".

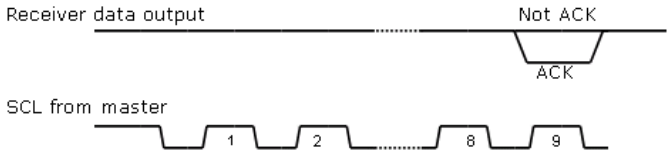


Fig. 22-4 I2C Acknowledge

- Byte transfer
  The master own I2C bus might initiate multi byte to transfer to a slave.The transfer starts from a "START" command and ends in a "STOP"command. After every byte transfer, the receiver must reply an ACK to transmitter.

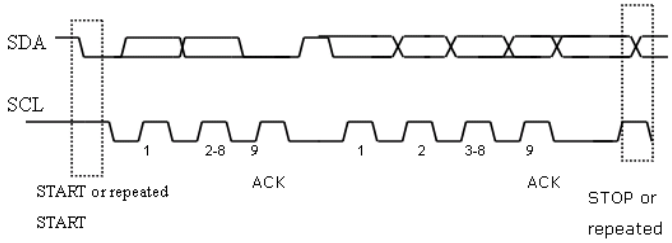


Fig. 22-5 I2C byte transfer

22.4 Register Description

22.4.1 Registers Summary Registers Summary

Name	Offset	Size	Reset Value	Description
RKI2C_CON	0x0000	W	0x00000000	control register
RKI2C_CLKDIV	0x0004	W	0x00000001	clock divider register
RKI2C_MRXADDR	0x0008	W	0x00000000	the slave address accessed for master rx mode
RKI2C_MRXRADDR	0x000c	W	0x00000000	the slave register address accessed for master rx mode
RKI2C_MTXCNT	0x0010	W	0x00000000	master transmit count
RKI2C_MRXCNT	0x0014	W	0x00000000	master rx count
RKI2C_IEN	0x0018	W	0x00000000	interrupt enable register
RKI2C_IPD	0x001c	W	0x00000000	interrupt pending register
RKI2C_FCNT	0x0020	W	0x00000000	finished count
RKI2C_SCL_OE_DB	0x0024	W	0x00000020	slave hold debounce configure register
RKI2C_TXDATA0	0x0100	W	0x00000000	I2C tx data register 0
RKI2C_TXDATA1	0x0104	W	0x00000000	I2C tx data register 1
RKI2C_TXDATA2	0x0108	W	0x00000000	I2C tx data register 2
RKI2C_TXDATA3	0x010c	W	0x00000000	I2C tx data register 3
RKI2C_TXDATA4	0x0110	W	0x00000000	I2C tx data register 4
RKI2C_TXDATA5	0x0114	W	0x00000000	I2C tx data register 5
RKI2C_TXDATA6	0x0118	W	0x00000000	I2C tx data register 6
RKI2C_TXDATA7	0x011c	W	0x00000000	I2C tx data register 7
RKI2C_RXDATA0	0x0200	W	0x00000000	I2C rx data register 0

Name	Offset	Size	Reset Value	Description
RKI2C_RXDATA1	0x0204	W	0x00000000	I2C rx data register 1
RKI2C_RXDATA2	0x0208	W	0x00000000	I2C rx data register 2
RKI2C_RXDATA3	0x020c	W	0x00000000	I2C rx data register 3
RKI2C_RXDATA4	0x0210	W	0x00000000	I2C rx data register 4
RKI2C_RXDATA5	0x0214	W	0x00000000	I2C rx data register 5
RKI2C_RXDATA6	0x0218	W	0x00000000	I2C rx data register 6
RKI2C_RXDATA7	0x021c	W	0x00000000	I2C rx data register 7
RKI2C_ST	0x0220	W	0x00000000	status debug register

*Notes:Size:**B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access*

22.4.2 Detail Register Description

RKI2C_CON

Address: Operational Base + offset (0x0000)
control register

Bit	Attr	Reset Value	Description
31:16	RO	0x0000	version rki2c version version information
15:14	RW	0x0	stop_setup staop setup config TSU;sto = (stop_setup + 1) * T(SCL_HIGH) + Tclk_i2c
13:12	RW	0x0	start_setup start setup config TSU;sta = (start_setup + 1) * T(SCL_HIGH) + Tclk_i2c THD;sta = (start_setup + 2) * T(SCL_HIGH) - Tclk_i2c
11	RO	0x0	reserved
10:8	RW	0x0	data_upd_st SDA update point config Used to config sda change state when scl is low, used to adjust setup/hold time 4'bn:Thold = (n + 1) * Tclk_i2c Note: 0 <= n <= 5
7	RO	0x0	reserved
6	RW	0x0	act2nak operation when NAK handshake is received 1'b0: ignored 1'b1: stop transaction
5	RW	0x0	ack last byte acknowledge control in master receive mode 1'b0: ACK 1'b1: NAK
4	RW	0x0	stop stop enable stop enable, when this bit is written to 1, I2C will generate stop signal.

Bit	Attr	Reset Value	Description
3	RW	0x0	start start enable start enable, when this bit is written to 1, I2C will generate start signal.
2:1	RW	0x0	i2c_mode i2c mode select 2'b00: transmit only 2'b01: transmit address (device + register address) --> restart --> transmit address -> receive only 2'b10: receive only 2'b11: transmit address (device + register address, write/read bit is 1) --> restart --> transmit address (device address) --> receive data
0	RW	0x0	i2c_en i2c module enable 1'b0:not enable 1'b1:enable

RKI2C_CLKDIV
Address: Operational Base + offset (0x0004)
clock divider register

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	CLKDIVH scl high level clock count T(SCL_HIGH) = Tclk_i2c * (CLKDIVH + 1) * 8
15:0	RW	0x0001	CLKDIVL scl low level clock count T(SCL_LOW) = Tclk_i2c * (CLKDIVL + 1) * 8

RKI2C_MRXADDR
Address: Operational Base + offset (0x0008)
the slave address accessed for master rx mode

Bit	Attr	Reset Value	Description
31:27	RO	0x0	reserved
26	RW	0x0	addhvld address high byte valid 1'b0:invalid 1'b1:valid
25	RW	0x0	addmvld address middle byte valid 1'b0:invalid 1'b1:valid

Bit	Attr	Reset Value	Description
24	RW	0x0	addlvld address low byte valid 1'b0:invalid 1'b1:valid
23:0	RW	0x000000	saddr master address register the lowest bit indicate write or read 24 bits address register

RKI2C_MRXRADDR
Address: Operational Base + offset (0x000c)
the slave register address accessed for master rx mode

Bit	Attr	Reset Value	Description
31:27	RO	0x0	reserved
26	RW	0x0	sraddhvld address high byte valid 1'b0:invalid 1'b1:valid
25	RW	0x0	sraddmvld address middle byte valid 1'b0:invalid 1'b1:valid
24	RW	0x0	sraddlvld address low byte valid 1'b0:invalid 1'b1:valid
23:0	RW	0x000000	sraddr slave register address accessed 24 bits register address

RKI2C_MTXCNT
Address: Operational Base + offset (0x0010)
master transmit count

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5:0	RW	0x00	mtxcnt master transmit count 6 bits counter

RKI2C_MRXCNT
Address: Operational Base + offset (0x0014)
master rx count

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved

Bit	Attr	Reset Value	Description
5:0	RW	0x00	mrxcnt master rx count 6 bits counter

RKI2C_IEN
Address: Operational Base + offset (0x0018)
interrupt enable register

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7	RW	0x0	slavehdsclen slave hold scl interrupt enable 1'b0:disable 1'b1:enable
6	RW	0x0	nakrcvien NAK handshake received interrupt enable 1'b0:disable 1'b1:enable
5	RW	0x0	stopien stop operation finished interrupt enable 1'b0:disable 1'b1:enable
4	RW	0x0	startien start operation finished interrupt enable 1'b0:disable 1'b1:enable
3	RW	0x0	mbrfien MRXCNT data received finished interrupt enable 1'b0:disable 1'b1:enable
2	RW	0x0	mbtfien MTXCNT data transfer finished interrupt enable 1'b0:disable 1'b1:enable
1	RW	0x0	brfien byte rx finished interrupt enable 1'b0:disable 1'b1:enable
0	RW	0x0	btfien byte tx finished interrupt enable 1'b0:disable 1'b1:enable

RKI2C_IPD
Address: Operational Base + offset (0x001c)
interrupt pending register

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7	RW	0x0	slavehdsclipd slave hold scl interrupt pending bit 1'b0:no interrupt available 1'b1:slave hold scl interrupt appear, write 1 to clear
6	W1C	0x0	nakrcvipd NAK handshake received interrupt pending bit 1'b0:no interrupt available 1'b1:NAK handshake received interrupt appear, write 1 to clear
5	W1C	0x0	stopipd stop operation finished interrupt pending bit 1'b0:no interrupt available 1'b1:stop operation finished interrupt appear, write 1 to clear
4	W1C	0x0	startipd start operation finished interrupt pending bit 1'b0:no interrupt available 1'b1:start operation finished interrupt appear, write 1 to clear
3	W1C	0x0	mbrfipd MRXCNT data received finished interrupt pending bit 1'b0:no interrupt available 1'b1:MRXCNT data received finished interrupt appear, write 1 to clear
2	W1C	0x0	mbtfipd MTXCNT data transfer finished interrupt pending bit 1'b0:no interrupt available 1'b1:MTXCNT data transfer finished interrupt appear, write 1 to clear
1	W1C	0x0	brfipd byte rx finished interrupt pending bit 1'b0:no interrupt available 1'b1:byte rx finished interrupt appear, write 1 to clear
0	W1C	0x0	btfipd byte tx finished interrupt pending bit 1'b0:no interrupt available 1'b1:byte tx finished interrupt appear, write 1 to clear

RKI2C_FCNT

Address: Operational Base + offset (0x0020)

finished count

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5:0	RO	0x00	fcnt finished count the count of data which has been transmitted or received for debug purpose

RKI2C_SCL_OE_DB
Address: Operational Base + offset (0x0024)
slave hold debounce configure register

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RW	0x20	scl_oe_db slave hold scl debounce cycles for debounce (unit: Tclk_i2c)

RKI2C_TXDATA0
Address: Operational Base + offset (0x0100)
I2C tx data register 0

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	txdata0 data0 to be transmitted 32 bits data

RKI2C_TXDATA1
Address: Operational Base + offset (0x0104)
I2C tx data register 1

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	txdata1 data1 to be transmitted 32 bits data

RKI2C_TXDATA2
Address: Operational Base + offset (0x0108)
I2C tx data register 2

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	txdata2 data2 to be transmitted 32 bits data

RKI2C_TXDATA3
Address: Operational Base + offset (0x010c)
I2C tx data register 3

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	txdata3 data3 to be transmitted 32 bits data

RKI2C_TXDATA4
Address: Operational Base + offset (0x0110)
I2C tx data register 4

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	txdata4 data4 to be transmitted 32 bits data

RKI2C_TXDATA5
Address: Operational Base + offset (0x0114)
I2C tx data register 5

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	txdata5 data5 to be transmitted 32 bits data

RKI2C_TXDATA6
Address: Operational Base + offset (0x0118)
I2C tx data register 6

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	txdata6 data6 to be transmitted 32 bits data

RKI2C_TXDATA7
Address: Operational Base + offset (0x011c)
I2C tx data register 7

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	txdata7 data7 to be transmitted 32 bits data

RKI2C_RXDATA0
Address: Operational Base + offset (0x0200)
I2C rx data register 0

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	rxdata0 data0 received 32 bits data

RKI2C_RXDATA1
Address: Operational Base + offset (0x0204)
I2C rx data register 1

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	rxdata1 data1 received 32 bits data

RKI2C_RXDATA2
Address: Operational Base + offset (0x0208)
I2C rx data register 2

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	rxdata2 data2 received 32 bits data

RKI2C_RXDATA3
Address: Operational Base + offset (0x020c)
I2C rx data register 3

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	rxdata3 data3 received 32 bits data

RKI2C_RXDATA4
Address: Operational Base + offset (0x0210)
I2C rx data register 4

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	rxdata4 data4 received 32 bits data

RKI2C_RXDATA5
Address: Operational Base + offset (0x0214)
I2C rx data register 5

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	rxdata5 data5 received 32 bits data

RKI2C_RXDATA6
Address: Operational Base + offset (0x0218)
I2C rx data register 6

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	rxdata6 data6 received 32 bits data

RKI2C_RXDATA7
Address: Operational Base + offset (0x021c)
I2C rx data register 7

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	rxdata7 data7 received 32 bits data

RKI2C_ST
Address: Operational Base + offset (0x0220)
status debug register

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1	RO	0x0	scl_st scl status 1'b0: scl status low 1'b0: scl status high
0	RO	0x0	sda_st sda status 1'b0: sda status low 1'b0: sda status high

22.5 Interface Description

Table 22-1 I2C Interface Description

Module pin	Direction	Pad name	IOMUX
I2C0 Interface			
i2c0_sda	I/O	IO_SPI3PMUrxd_I2C0PMUsda_PMU1830gpio1b7	PMUGRF_GPIO1B_IOMUX[15:14]=2'b10
i2c0_scl	I/O	IO_SPI3PMUtxd_I2C0PMUscl_PMU1830gpio1c0	PMUGRF_GPIO1C_IOMUX[1:0]=2'b10
I2C1 Interface			
i2c1_sda	I/O	IO_I2C1AUDIOCAMsda_TRACEclk_AUDIOgpio4a1	GRF_GPIO4A_IOMUX[3:2]=2'b01
i2c1_scl	I/O	IO_I2C1AUDIOCAMscl_TRACEdata8_AUDIOgpio4a2	GRF_GPIO4A_IOMUX[5:4]=2'b01
I2C2 Interface			
i2c2_sda	I/O	IO_VOPdata0_I2C2TPsda_CIFdata0_BT656gpio2a0	GRF_GPIO2A_IOMUX[1:0]=2'b10
i2c2_scl	I/O	IO_VOPdata1_I2C2TPscl_CIFdata1_BT656gpio2a1	GRF_GPIO2A_IOMUX[3:2]=2'b10
I2C3 Interface			
i2c3_sda	I/O	IO_I2C3HDMIsda_UART2DBGBsin_HDMII2Csda_GPIO1830gpio4c0	GRF_GPIO4C_IOMUX[1:0]=2'b01
i2c3_scl	I/O	IO_I2C3HDMIscl_UART2DBGBsout_HDMII2Cscl_GPIO1830gpio4c1	GRF_GPIO4C_IOMUX[3:2]=2'b01
I2C4 Interface			
i2c4_sda	I/O	IO_I2C4SENSORsda_PMU1830gpio1b3	PMUGRF_GPIO1B_IOMUX[7:6]=2'b01
i2c4_scl	I/O	IO_I2C4SENSORscl_PMU1830gpio1b4	PMUGRF_GPIO1B_IOMUX[9:8]=2'b01
I2C5 Interface			
i2c5_sda	I/O	IO_MACrxer_I2C5TRACKPADsda_GMACgpio3b2	GRF_GPIO3B_IOMUX[5:4]=2'b10
i2c5_scl	I/O	IO_MACclk_I2C5TRACKPADscl_GMACgpio3b3	GRF_GPIO3B_IOMUX[7:6]=2'b10
I2C6 Interface			
i2c6_sda	I/O	IO_SPI2TPMrxd_I2C6TPMsda_CIFhref_BT656gpio2b1	GRF_GPIO2B_IOMUX[3:2]=2'b10
i2c6_scl	I/O	IO_SPI2TPMtxd_I2C6TPMscl_CIFclkin_BT656gpio2b2	GRF_GPIO2B_IOMUX[5:4]=2'b10
I2C7 Interface			

Module pin	Direction	Pad name	IOMUX
i2c7_sda	I/O	IO_VOPdata7_I2C7NFCsda_CIFdata7_BT656gpio2a7	GRF_GPIO2A_IOMUX[15:14]=2'b10
i2c7_scl	I/O	IO_VOPdclk_I2C7NFCscl_CIFvsync_BT656gpio2b0	GRF_GPIO2B_IOMUX[1:0]=2'b10
I2C8 Interface			
i2c8_sda	I/O	IO_I2C8DCDCsda_PMU1830gpio1c4	PMUGRF_GPIO1C_IOMUX[9:8]=2'b01
i2c8_scl	I/O	IO_I2C8DCDCscl_PMU1830gpio1c5	PMUGRF_GPIO1C_IOMUX[11:10]=2'b01

22.6 Application Notes

The I2C controller core operation flow chart below is to describe how the software configures and performs an I2C transaction through this I2C controller core. Descriptions are divided into 3 sections, transmit only mode, receive only mode, and mix mode. Users are strongly advised to follow

- Transmit only mode (I2C_CON[1:0]=2'b00)

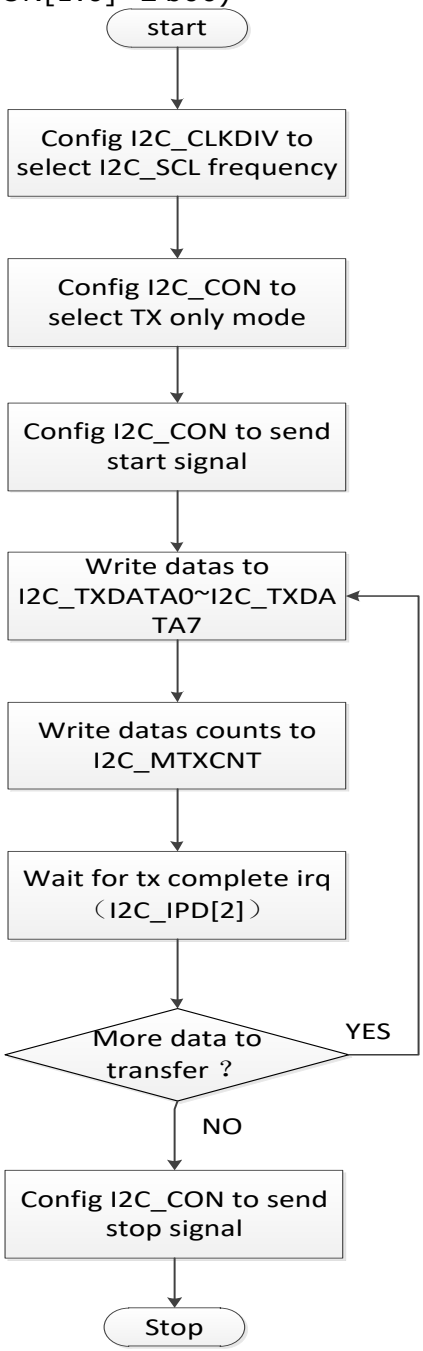


Fig. 22-6 I2C Flow chat for transmit only mode

- Receive only mode (I2C_CON[1:0]=2'b10)

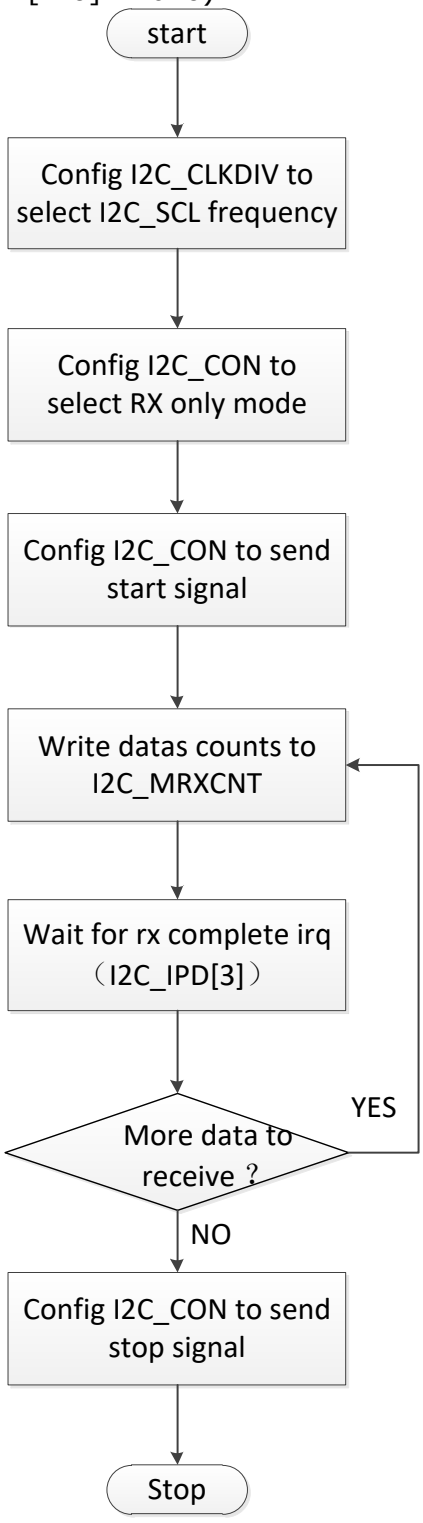


Fig. 22-7 I2C Flow chat for receive only mode

- Mix mode (I2C_CON[1:0]=2'b01 or I2C_CON[1:0]=2'b11)

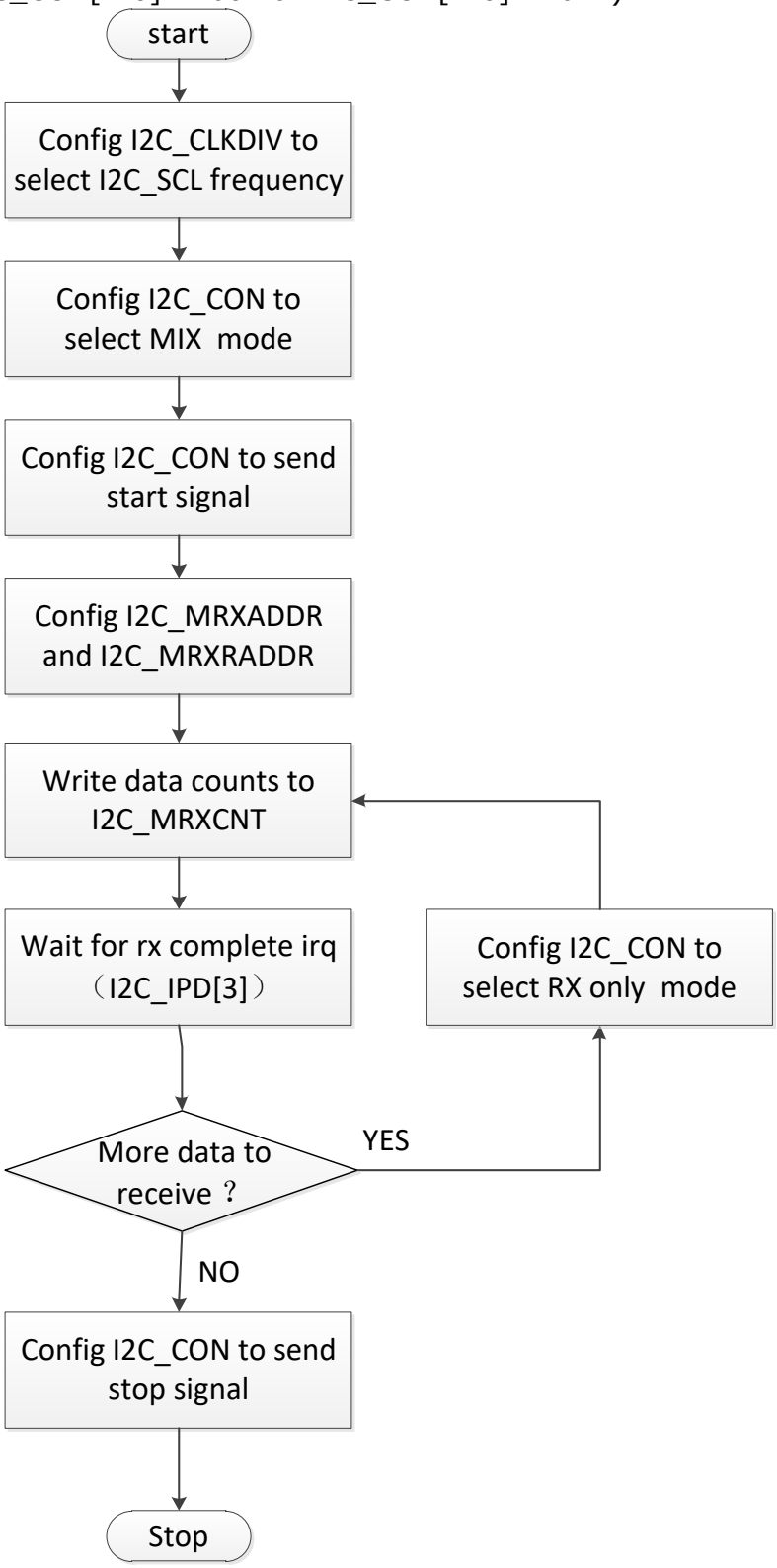


Fig. 22-8 I2C Flow chat for mix mode

Chapter 23 I2S/PCM Controller

23

23.1 Overview

The I2S/PCM controller is designed for interfacing between the AHB bus and the I2S bus. The I2S bus (Inter-IC sound bus) is a serial link for digital audio data transfer between devices in the system and be invented by Philips Semiconductor. Now it is widely used by many semiconductor manufacturers.
Devices often use the I2S bus are ADC, DAC, DSP, CPU, etc. With the I2S interface, we can connect audio devices and the embedded SoC platform together and provide an audio interface solution for the system.
Not only I2S but also PCM mode surround audio output and stereo input are supported in I2S/PCM controller.
There are three I2S/PCM controllers embedded in the design, I2S0, I2S1 and I2S2.
Different features between I2S/PCM controllers are as follows.

- Support eight internal 32-bit wide and 32-location deep FIFOs, four for transmitting and four for receiving audio data for I2S0
- Support two internal 32-bit wide and 32-location deep FIFOs, one for transmitting and one for receiving audio data for I2S1
- Support four internal 32-bit wide and 32-location deep FIFOs, four for transmitting audio data for I2S2
- Support 10 channels audio data transmitting and receiving in total in I2S mode for I2S0, 2 channels audio data transmitting and 2 channels audio data receiving for I2S1, 8 channels audio data transmitting for I2S2
- Support up to 192kHz sample rate for I2S0 and I2S1, 768kHz sample rate for I2S2

Common features for I2S0, I2S1 and I2S2 are as follows.

- Support AHB bus interface
- Support 16 ~ 32 bits audio data transfer
- Support master and slave mode
- Support DMA handshake interface and configurable DMA water level
- Support transmit FIFO empty, underflow, receive FIFO full, overflow interrupt and all interrupts can be masked
- Support configurable water level of transmit FIFO empty and receive FIFO full interrupt
- Support combine interrupt output
- Support 2 channels audio receiving in PCM mode
- Support I2S normal, left and right justified mode serial audio data transfer
- Support PCM early, late1, late2, late3 mode serial audio data transfer
- Support MSB or LSB first serial audio data transfer
- Support 16 to 31 bit audio data left or right justified in 32-bit wide FIFO
- Support two 16-bit audio data store together in one 32-bit wide location
- Support 2 independent LRCK signals, one for receiving and one for transmitting audio data. Single LRCK can be used for transmitting and receiving data if the sample rate are the same
- Support configurable SCLK and LRCK polarity

23.2 Block Diagram

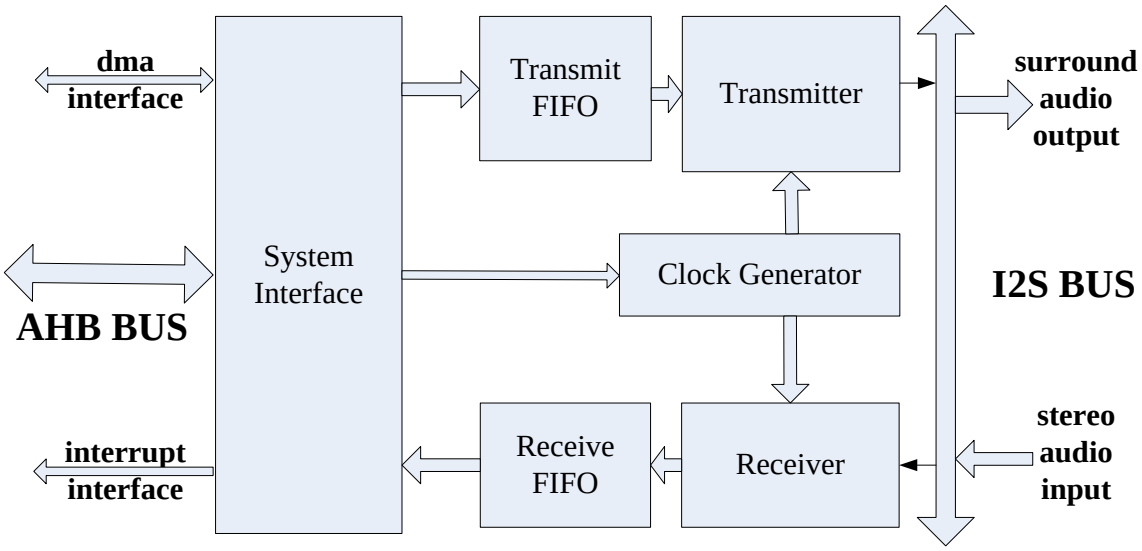


Fig. 23-1 I2S/PCM controller (8 channel) Block Diagram

System Interface

The system interface implements the AHB slave operation. It contains not only control registers of transmitter and receiver inside but also interrupt and DMA handshake interface.

Clock Generator

The Clock Generator implements clock generation function. The input source clock to the module is MCLK_I2S, and by the divider of the module, the clock generator generates SCLK and LRCK to transmitter and receiver.

Transmitter

The Transmitter implements transmission operation. The transmitter can act as either master or slave, with I2S or PCM mode surround serial audio interface.

Receiver

The Receiver implements receive operation. The receiver can act as either master or slave, with I2S or PCM mode stereo serial audio interface.

Transmit FIFO

The Transmit FIFO is the buffer to store transmitted audio data. The size of the FIFO is 32bits x 32.

Receive FIFO

The Receive FIFO is the buffer to store received audio data. The size of the FIFO is 32bits x 32.

23.3 Function description

In the I2S/PCM controller, there are four conditions: transmitter-master & receiver-master; transmitter-master & receiver-slave; transmitter-slave & receiver-master; transmitter-slave & receiver-slave.

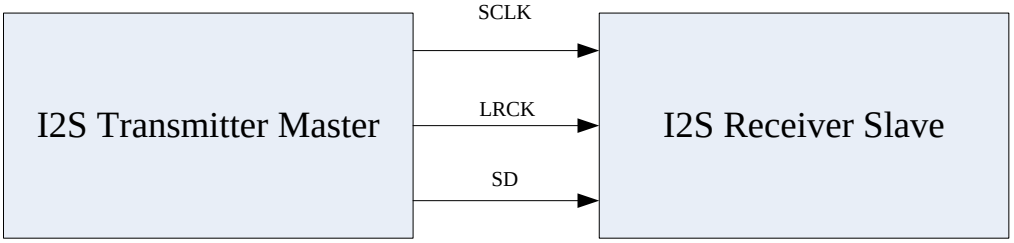


Fig. 23-2 I2S transmitter-master & receiver-slave condition

When transmitter acts as a master, it sends all signals to receiver (slave), and CPU control when to send clock and data to the receiver. When acting as a slave, SD signal still goes from transmitter to receiver, but SCLK and LRCK signals are from receiver (master) to transmitter. Based on three interface specifications, transmitting data should be ready

before transmitter receives SCLK and LRCK signals. CPU should know when the receiver to initialize a transaction and when to send data.

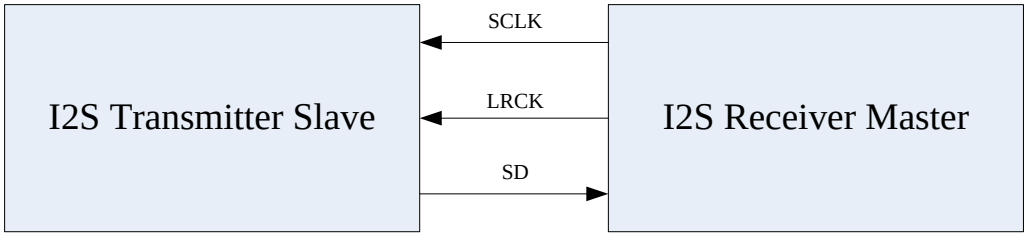


Fig. 23-3 I2S transmitter-slave& receiver-master condition

When the receiver acts as a master, it sends SCLK and LRCK signals to the transmitter (slave) and receives serial data. So CPU must tell the transmitter when to start a transaction for it to prepare transmitting data then the receiver start a transfer and send clock and channel-select signals. When the receiver acts as a slave, CPU should only do initial setting and wait for all signals and then start reading data.

Before transmitting or receiving data, CPU need do initial setting to the I2S register. These includes CPU settings, I2S interface registers settings, and maybe the embedded SoC platform settings. These registers must be set before starting data transfer.

23.3.1 i2s normal mode

This is the waveform of I2S normal mode. For LRCK (i2s_lrck_rx/i2s_lrck_tx) signal, it goes low to indicate left channel and high to right channel. For SD (i2s_sdo, i2s_sdi) signal, it transfers MSB or LSB first and sends the first bit one SCLK clock cycle after LRCK changes. The range of SD signal width is from 16 to 32bits.

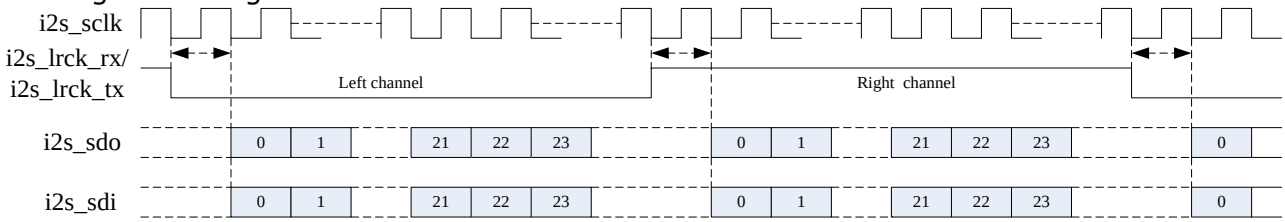


Fig. 23-4 I2S normal mode timing format

23.3.2 i2s left justified mode

This is the waveform of I2S left justified mode. For LRCK (i2s_lrck_rx / i2s_lrck_tx) signal, it goes high to indicate left channel and low to right channel. For SD (i2s_sdo, i2s_sdi) signal, it transfers MSB or LSB first and sends the first bit at the same time when LRCK changes. The range of SD signal width is from 16 to 32bits.

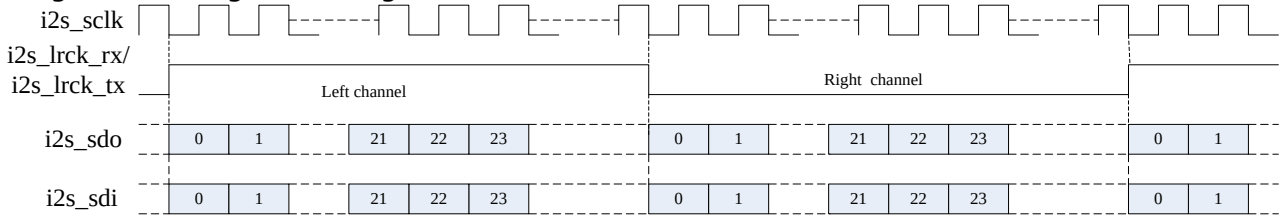


Fig. 23-5 I2S left justified mode timing format

23.3.3 i2s right justified mode

This is the waveform of I2S right justified mode. For LRCK (i2s_lrck_rx / i2s_lrck_tx) signal, it goes high to indicate left channel and low to right channel. For SD (i2s_sdo, i2s_sdi) signal, it transfers MSB or LSB first; but different from I2S normal or left justified mode, its data is aligned to last bit at the edge of the LRCK signal. The range of SD signal width is from 16 to 32bits.

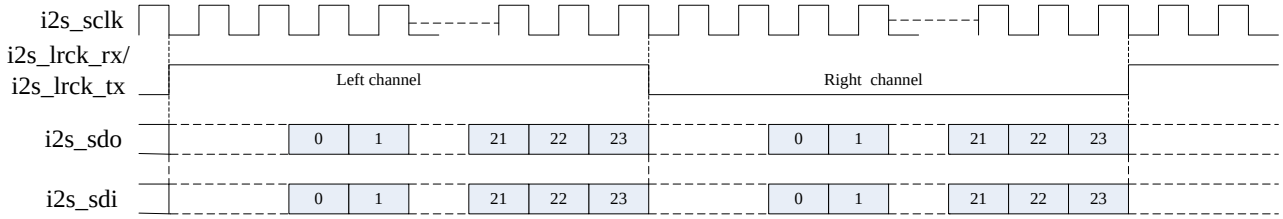


Fig. 23-6 I2S right justified mode timing format

23.3.4 PCM early mode

This is the waveform of PCM early mode. For LRCK (i2s_lrck_rx / i2s_lrck_tx) signal, it goes high to indicate the start of a group of audio channels. For SD (i2s_sdo, i2s_sdi) signal, it transfers MSB or LSB first and sends the first bit at the same time when LRCK goes high. The range of SD signal width is from 16 to 32bits.

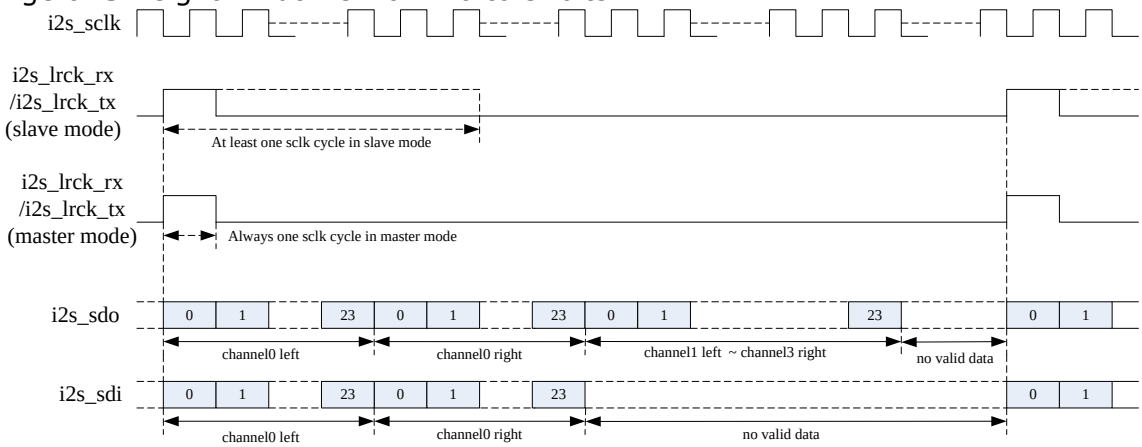


Fig. 23-7 PCM early mode timing format

23.3.5 PCM late1 mode

This is the waveform of PCM late1 mode. For LRCK (i2s_lrck_rx / i2s_lrck_tx) signal, it goes high to indicate the start of a group of audio channels. For SD (i2s_sdo, i2s_sdi) signal, it transfers MSB or LSB first and sends the first bit one SCLK clock cycle after LRCK goes high. The range of SD signal width is from 16 to 32bits.

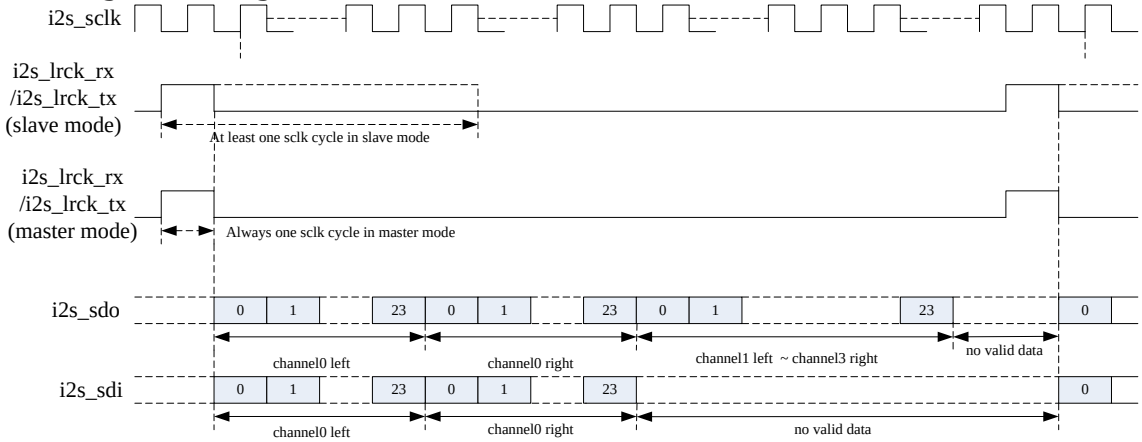


Fig. 23-8 PCM late1 mode timing format

23.3.6 PCM late2 mode

This is the waveform of PCM late2 mode. For LRCK (i2s_lrck_rx / i2s_lrck_tx) signal, it goes high to indicate the start of a group of audio channels. For SD (i2s_sdo, i2s_sdi) signal, it transfers MSB or LSB first and sends the first bit two SCLK clock cycles after LRCK goes high. The range of SD signal width is from 16 to 32bits.

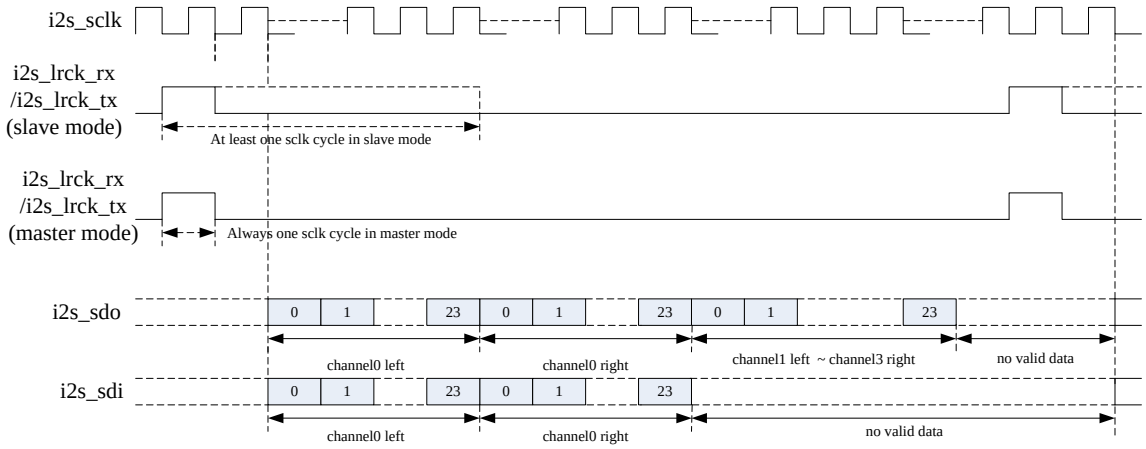


Fig. 23-9 PCM late2 mode timing format

23.3.7 PCM late3 mode

This is the waveform of PCM late3 mode. For LRCK (i2s_lrck_rx / i2s_lrck_tx) signal, it goes high to indicate the start of a group of audio channels. For SD (i2s_sdo, i2s_sdi) signal, it transfers MSB or LSB first and sends the first bit three SCLK clock cycles after LRCK goes high. The range of SD signal width is from 16 to 32bits.

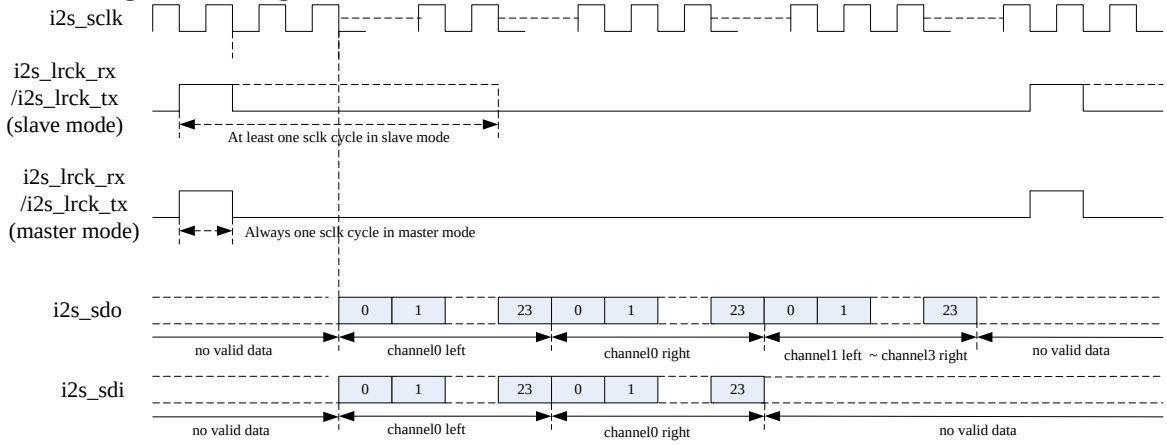


Fig. 23-10 PCM late3 mode timing format

23.4 Register Description

This section describes the control/status registers of the design.

23.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
I2S_TXCR	0x0000	W	0x0000000f	transmit operation control register
I2S_RXCR	0x0004	W	0x0000000f	receive operation control register
I2S_CKR	0x0008	W	0x00071f1f	clock generation register
I2S_TXFIFOLR	0x000c	W	0x00000000	TX FIFO level register
I2S_DMACR	0x0010	W	0x001f0000	DMA control register
I2S_INTCR	0x0014	W	0x00000000	interrupt control register
I2S_INTSR	0x0018	W	0x00000000	interrupt status register
I2S_XFER	0x001c	W	0x00000000	Transfer Start Register
I2S_CLR	0x0020	W	0x00000000	SCLK domain logic clear Register
I2S_TXDR	0x0024	W	0x00000000	Transmit FIFO Data Register
I2S_RXDR	0x0028	W	0x00000000	Receive FIFO Data Register
I2S_RXFIFOLR	0x002c	W	0x00000000	RX FIFO level register

*Notes:Size:**B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access*

23.4.2 Detail Register Description

I2S_TXCR

Address: Operational Base + offset (0x0000)
transmit operation control register

Bit	Attr	Reset Value	Description
31:23	RO	0x0	reserved
22:17	RW	0x00	RCNT right justified counter (Can be written only when XFER[0] bit is 0.) Only valid in I2S Right justified format and slave tx mode is selected. Start to transmit data RCNT sclk cycles after left channel valid.
16:15	RW	0x0	TCSR TX Channel select register 2'b00:two channel 2'b01:four channel 2'b10:six channel 2'b11:eight channel
14	RW	0x0	HWT Halfword word transform (Can be written only when XFER[0] bit is 0.) Only valid when VDW select 16bit data. 0:32 bit data valid from AHB/APB bus. Low 16 bit for left channel and high 16 bit for right channel. 1:low 16bit data valid from AHB/APB bus, high 16 bit data invalid.
13	RO	0x0	reserved
12	RW	0x0	SJM Store justified mode SJM Store justified mode (Can be written only when XFER[1] bit is 0.) 16bit~31bit DATA stored in 32 bits width fifo. This bit is invalid if VDW select 16bit data and HWT select 0, Because every fifo unit contain two 16bit data and 32 bit space is full, it is impossible to choose justified mode. 0:right justified 1:left justified
11	RW	0x0	FBM First Bit Mode (Can be written only when XFER[0] bit is 0.) 0:MSB 1:LSB

Bit	Attr	Reset Value	Description
10:9	RW	0x0	IBM I2S bus mode (Can be written only when XFER[0] bit is 0.) 0:I2S normal 1:I2S Left justified 2:I2S Right justified 3:reserved
8:7	RW	0x0	PBM PCM bus mode (Can be written only when XFER[0] bit is 0.) 0:PCM no delay mode 1:PCM delay 1 mode 2:PCM delay 2 mode 3:PCM delay 3 mode
6	RO	0x0	reserved
5	RW	0x0	TFS Transfer format select (Can be written only when XFER[0] bit is 0.) 0: I2S format 1: PCM format
4:0	RW	0x0f	VDW Valid Data width (Can be written only when XFER[0] bit is 0.) 0~14:reserved 15:16bit 16:17bit 17:18bit 18:19bit …… n:(n+1)bit …… 28:29bit 29:30bit 30:31bit 31:32bit

I2S_RXCR

Address: Operational Base + offset (0x0004)

receive operation control register

Bit	Attr	Reset Value	Description
31:17	RO	0x0	reserved

Bit	Attr	Reset Value	Description
16:15	RW	0x0	RCSR RX Channel select register 2'b00:two channel 2'b01:four channel 2'b10:six channel 2'b11:eight channel
14	RW	0x0	HWT Halfword word transform (Can be written only when XFER[1] bit is 0.) Only valid when VDW select 16bit data. 0:32 bit data valid to AHB/APB bus. Low 16 bit for left channel and high 16 bit for right channel. 1:low 16bit data valid to AHB/APB bus, high 16 bit data invalid.
13	RO	0x0	reserved
12	RW	0x0	SJM Store justified mode (Can be written only when XFER[1] bit is 0.) 16bit~31bit DATA stored in 32 bits width fifo. If VDW select 16bit data, this bit is valid only when HWT select 0.Because if HWT is 1, every fifo unit contain two 16bit data and 32 bit space is full, it is impossible to choose justified mode. 0:right justified 1:left justified
11	RW	0x0	FBM First Bit Mode (Can be written only when XFER[1] bit is 0.) 0:MSB 1:LSB
10:9	RW	0x0	IBM I2S bus mode (Can be written only when XFER[1] bit is 0.) 0:I2S normal 1:I2S Left justified 2:I2S Right justified 3:reserved
8:7	RW	0x0	PBM PCM bus mode (Can be written only when XFER[1] bit is 0.) 0:PCM no delay mode 1:PCM delay 1 mode 2:PCM delay 2 mode 3:PCM delay 3 mode
6	RO	0x0	reserved

Bit	Attr	Reset Value	Description
5	RW	0x0	TFS Transfer format select (Can be written only when XFER[1] bit is 0.) 0:i2s 1:pcm
4:0	RW	0x0f	VDW Valid Data width (Can be written only when XFER[1] bit is 0.) 0~14:reserved 15:16bit 16:17bit 17:18bit 18:19bit …… n:(n+1)bit …… 28:29bit 29:30bit 30:31bit 31:32bit

I2S_CKR
Address: Operational Base + offset (0x0008)
clock generation register

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved
29:28	RW	0x0	TRCM Tx and Rx Common Use 2'b00/2'b11:tx_lrck/rx_lrck are used as synchronous signal for TX /RX respectively. 2'b01:only tx_lrck is used as synchronous signal for TX and RX. 2'b10:only rx_lrck is used as synchronous signal for TX and RX.
27	RW	0x0	MSS Master/slave mode select (Can be written only when XFER[1] or XFER[0] bit is 0.) 0:master mode(sclk output) 1:slave mode(sclk input)
26	RW	0x0	CKP Sclk polarity (Can be written only when XFER[1] or XFER[0] bit is 0.) 0: sample data at posedge sclk and drive data at negedge sclk 1: sample data at negedge sclk and drive data at posedge sclk

Bit	Attr	Reset Value	Description
25	RW	0x0	RLP Receive lrck polarity (Can be written only when XFER[1] or XFER[0] bit is 0.) 0:normal polarity (I2S normal: low for left channel, high for right channel I2S left/right just: high for left channel, low for right channel PCM start signal: high valid) 1:oppsite polarity (I2S normal: high for left channel, low for right channel I2S left/right just: low for left channel, high for right channel PCM start signal: low valid)
24	RW	0x0	TLP Transmit lrck polarity (Can be written only when XFER[1] or XFER[0] bit is 0.) 0:normal polarity (I2S normal: low for left channel, high for right channel I2S left/right just: high for left channel, low for right channel PCM start signal: high valid) 1:oppsite polarity (I2S normal: high for left channel, low for right channel I2S left/right just: low for left channel, high for right channel PCM start signal: low valid)
23:16	RW	0x07	MDIV mclk divider (Can be written only when XFER[1] or XFER[0] bit is 0.) Serial Clock Divider = Fmclk / Ftxsclk-1.(mclk frequecy / txsclk frequecy-1) 0 :Fmclk=Ftxsclk; 1 :Fmclk=2*Ftxsclk; 2,3 :Fmclk=4*Ftxsclk; 4,5 :Fmclk=6*Ftxsclk; …… 2n,2n+1:Fmclk=(2n+2)*Ftxsclk; …… 60,61:Fmclk=62*Ftxsclk; 62,63:Fmclk=64*Ftxsclk; …… 252,253:Fmclk=254*Ftxsclk; 254,255:Fmclk=256*Ftxsclk;

Bit	Attr	Reset Value	Description
15:8	RW	0x1f	RSD Receive sclk divider (Can be written only when XFER[1] or XFER[0] bit is 0.) Receive sclk divider= Fsclk/Frxlrck 0~30:reserved 31: 32fs 32: 33fs 33: 34fs 34: 35fs …… n: (n+1)fs …… 253: 254fs 254: 255fs 255: 256fs
7:0	RW	0x1f	TSD Transmit sclk divider (Can be written only when XFER[1] or XFER[0] bit is 0.) Transmit sclk divider=Ftxsclk/Ftxlrck 0~30:reserved 31: 32fs 32: 33fs 33: 34fs 34: 35fs …… n: (n+1)fs …… 253: 254fs 254: 255fs 255: 256fs

I2S_TXFIFOLR

Address: Operational Base + offset (0x000c)

TX FIFO level register

Bit	Attr	Reset Value	Description
31:24	RO	0x0	reserved
23:18	RO	0x00	TFL3 Transmit FIFO3 Level Contains the number of valid data entries in the transmit FIFO3.
17:12	RO	0x00	TFL2 Transmit FIFO2 Level Contains the number of valid data entries in the transmit FIFO2.
11:6	RO	0x00	TFL1 Transmit FIFO1 Level Contains the number of valid data entries in the transmit FIFO1.

Bit	Attr	Reset Value	Description
5:0	RO	0x00	TFL0 Transmit FIFO0 Level Contains the number of valid data entries in the transmit FIFO0.

I2S_DMACR
Address: Operational Base + offset (0x0010)
DMA control register

Bit	Attr	Reset Value	Description
31:25	RO	0x0	reserved
24	RW	0x0	RDE Receive DMA Enable 0 : Receive DMA disabled 1 : Receive DMA enabled
23:21	RO	0x0	reserved
20:16	RW	0x1f	RDL Receive Data Level This bit field controls the level at which a DMA request is made by the receive logic. The watermark level = DMARDL+1; that is, dma_rx_req is generated when the number of valid data entries in the receive FIFO (RXFIFO0 if RCSR=00;RXFIFO1 if RCSR=01,RXFIFO2 if RCSR=10,RXFIFO3 if RCSR=11)is equal to or above this field value + 1.
15:9	RO	0x0	reserved
8	RW	0x0	TDE Transmit DMA Enable 0 : Transmit DMA disabled 1 : Transmit DMA enabled
7:5	RO	0x0	reserved
4:0	RW	0x00	TDL Transmit Data Level This bit field controls the level at which a DMA request is made by the transmit logic. It is equal to the watermark level; that is, the dma_tx_req signal is generated when the number of valid data entries in the TXFIFO(TXFIFO0 if TCSR=00;TXFIFO1 if TCSR=01,TXFIFO2 if TCSR=10,TXFIFO3 if TCSR=11)is equal to or below this field value.

I2S_INTCR
Address: Operational Base + offset (0x0014)
interrupt control register

Bit	Attr	Reset Value	Description
31:25	RO	0x0	reserved

Bit	Attr	Reset Value	Description
24:20	RW	0x00	RFT Receive FIFO Threshold When the number of receive FIFO entries (RXFIFO0 if RCSR=00; RXFIFO1 if RCSR=01, RXFIFO2 if RCSR=10, RXFIFO3 if RCSR=11) is more than or equal to this threshold plus 1, the receive FIFO full interrupt is triggered.
19	RO	0x0	reserved
18	WO	0x0	RXOIC RX overrun interrupt clear Write 1 to clear RX overrun interrupt.
17	RW	0x0	RXOIE RX overrun interrupt enable 0:disable 1:enable
16	RW	0x0	RXFIE RX full interrupt enable 0:disable 1:enable
15:9	RO	0x0	reserved
8:4	RW	0x00	TFT Transmit FIFO Threshold When the number of transmit FIFO (TXFIFO0 if TCSR=00; TXFIFO1 if TCSR=01, TXFIFO2 if TCSR=10, TXFIFO3 if TCSR=11) entries is less than or equal to this threshold, the transmit FIFO empty interrupt is triggered.
3	RO	0x0	reserved
2	WO	0x0	TXUIC TX underrun interrupt clear Write 1 to clear TX underrun interrupt.
1	RW	0x0	TXUIE TX underrun interrupt enable 0:disable 1:enable
0	RW	0x0	TXEIE TX empty interrupt enable 0:disable 1:enable

I2S_INTSR

Address: Operational Base + offset (0x0018)

interrupt status register

Bit	Attr	Reset Value	Description
31:18	RO	0x0	reserved

Bit	Attr	Reset Value	Description
17	RO	0x0	RXOI RX overrun interrupt 0:inactive 1:active
16	RO	0x0	RXFI RX full interrupt 0:inactive 1:active
15:2	RO	0x0	reserved
1	RO	0x0	TXUI TX underrun interrupt 0:inactive 1:active
0	RO	0x0	TXEI TX empty interrupt 0:inactive 1:active

I2S_XFER
Address: Operational Base + offset (0x001c)
Transfer Start Register

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1	RW	0x0	RXS RX Transfer start bit 0:stop RX transfer. 1:start RX transfer
0	RW	0x0	TXS TX Transfer start bit 0:stop TX transfer. 1:start TX transfer

I2S_CLR
Address: Operational Base + offset (0x0020)
SCLK domain logic clear Register

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1	RW	0x0	RXC RX logic clear This is a self cleared bit. Write 1 to clear all receive logic.
0	RW	0x0	TXC TX logic clear This is a self cleared bit. Write 1 to clear all transmit logic.

I2S_TXDR
Address: Operational Base + offset (0x0024)
Transmit FIFO Data Register

Bit	Attr	Reset Value	Description
31:0	WO	0x00000000	TXDR Transmit FIFO Data Register When it is written to, data are moved into the transmit FIFO.

I2S_RXDR
Address: Operational Base + offset (0x0028)
Receive FIFO Data Register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	RXDR Receive FIFO Data Register When the register is read, data in the receive FIFO is accessed.

I2S_RXFIFOLR
Address: Operational Base + offset (0x002c)
RX FIFO level register

Bit	Attr	Reset Value	Description
31:24	RO	0x0	reserved
23:18	RO	0x00	RFL3 Receive FIFO3 Level Contains the number of valid data entries in the receive FIFO3.
17:12	RO	0x00	RFL2 Receive FIFO2 Level Contains the number of valid data entries in the receive FIFO2.
11:6	RU	0x00	RFL1 Receive FIFO1 Level Contains the number of valid data entries in the receive FIFO1.
5:0	RO	0x00	RFL0 Receive FIFO0 Level Contains the number of valid data entries in the receive FIFO0.

23.5 Interface description

Table 23-1 I2S Interface Description

Module Pin	Direction	Pad Name	IOMUX Setting
Interface for i2s0			
i2s_mclk	I/O	IO_I2Sclk_TRACEctl_LPM0wfi_AUDIOgpio4a0	GRF_GPIO4A_IOMUX[1:0]=2'b01
i2s0_sclk	I/O	IO_I2S0sclk_TRACEdata0_A72CORE0wfi_AUDIOgpio3d0	GRF_GPIO3D_IOMUX[1:0]=2'b01
i2s0_lrck_rx	I/O	IO_I2S0lrckrx_TRACEdata1_A72CORE1wfi_AUDIOgpio3d1	GRF_GPIO3D_IOMUX[3:2]=2'b01
i2s0_lrck_tx	I/O	IO_I2S0lrcktx_TRACEdata2_A53CORE0wfi_AUDIOgpio3d2	GRF_GPIO3D_IOMUX[5:4]=2'b01
i2s0_sdo0	O	IO_I2S0sdo0_TRACEdata7_A53L2wfi_AUDIOgpio3d7	GRF_GPIO3D_IOMUX[15:14]=2'b01
i2s0_sdo1	O	IO_I2S0sdi3sdo1_TRACEdata6_A72L2wfi_AUDIOgpio3d6	GRF_GPIO3D_IOMUX[13:12]=2'b01
i2s0_sdo2	O	IO_I2S0sdi2sdo2_TRACEdata5_A	GRF_GPIO3D_IOMUX[11:10]=2'b01

Module Pin	Direction	Pad Name	IOMUX Setting
		53CORE3wfi_AUDIOgpio3d5	
i2s0_sdo3	O	IO_I2S0sdi1sdo3_TRACEdata4_A53CORE2wfi_AUDIOgpio3d4	GRF_GPIO3D_IOMUX[9:8]=2'b01
i2s0_sdi0	I	IO_I2S0sdi0_TRACEdata3_A53CORE1wfi_AUDIOgpio3d3	GRF_GPIO3D_IOMUX[7:6]=2'b01
i2s0_sdi1	I	IO_I2S0sdi1sdo3_TRACEdata4_A53CORE2wfi_AUDIOgpio3d4	GRF_GPIO3D_IOMUX[9:8]=2'b01
i2s0_sdi2	I	IO_I2S0sdi2sdo2_TRACEdata5_A53CORE3wfi_AUDIOgpio3d5	GRF_GPIO3D_IOMUX[11:10]=2'b01
i2s0_sdi3	I	IO_I2S0sdi3sdo1_TRACEdata6_A72L2wfi_AUDIOgpio3d6	GRF_GPIO3D_IOMUX[13:12]=2'b01
Interface for i2s1			
i2s1_mclk	I/O	IO_I2Sclk_TRACEctl_LPM0wfi_AUDIOgpio4a0	GRF_GPIO4A_IOMUX[1:0]=2'b01
i2s1_sclk	I/O	IO_I2S1sclk_TRACEdata9_AUDIOgpio4a3	GRF_GPIO4A_IOMUX[7:6]=2'b01
i2s1_lrck_rx	I/O	IO_I2S1lrckrx_TRACEdata10_AUDIOgpio4a4	GRF_GPIO4A_IOMUX[9:8]=2'b01
i2s1_lrck_tx	I/O	IO_I2S1lrcktx_TRACEdata11_AUDIOgpio4a5	GRF_GPIO4A_IOMUX[11:10]=2'b01
i2s1_sdi	I	IO_I2S1sdi0_AUDIOgpio4a6	GRF_GPIO4A_IOMUX[13:12]=2'b01
i2s1_sdo	O	IO_I2S1sdo0_AUDIOgpio4a7	GRF_GPIO4A_IOMUX[15:14]=2'b01

Notes: I=input, O=output, I/O=input/output, bidirectional

The i2s0_sdix(x=1,2,3) and i2s0_sdox(x=1,2,3) signals shares the same IO, the direction is configured by setting GRF_SOC_CON8[13:11]. GRF_SOC_CON8[11] controls the direction of IO_I2S0sdi3sdo1_TRACEdata6_A72L2wfi_AUDIOgpio3d6, GRF_SOC_CON8[12] corresponds to IO_I2S0sdi2sdo2_TRACEdata5_A53CORE3wfi_AUDIOgpio3d5 and GRF_SOC_CON8[13] corresponds to IO_I2S0sdi1sdo3_TRACEdata4_A53CORE2wfi_AUDIOgpio3d4.

The I2S2 module is connected to the audio interface of HDMI and DP, which supports 8 channels audio data transmitting.

Table 23-3 I2S Interface Between I2S2 and HDMI

Module Pin	Direction	Module Pin	Direction
i2s2_sclk_out	O	ii2sclk	I
i2s2_tx_lrck_out	O	ii2slrclk	I
i2s2_sdo[3:0]	O	ii2sdata[3:0]	I

Table 23-4 I2S Interface Between I2S2 and DP

Module Pin	Direction	Module Pin	Direction
i2s2_sclk_out	O	source_i2s_clk	I
i2s2_tx_lrck_out	O	source_i2s_ws	I
i2s2_8ch_sdo[3:0]	O	source_i2s_data[3:0]	I

23.6 Application Notes

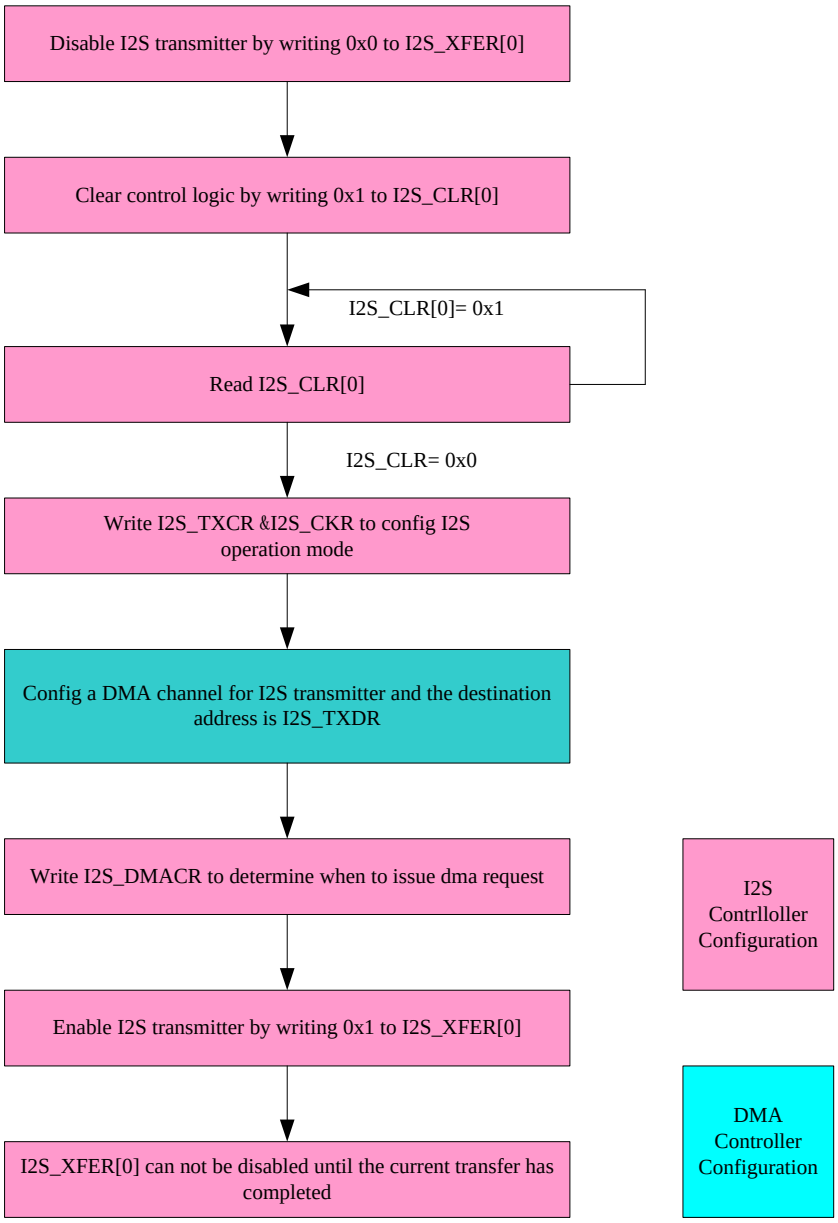


Fig. 23-11 I2S/PCM controller transmit operation flow chart

Chapter 24 Serial Peripheral Interface (SPI)

24.1 Overview

The serial peripheral interface is an APB slave device. A four wire full duplex serial protocol from Motorola. There are four possible combinations for the serial clock phase and polarity. The clock phase (SCPH) determines whether the serial transfer begins with the falling edge of slave select signals or the first edge of the serial clock. The slave select line is held high when the SPI is idle or disabled. This SPI controller can work as either master or slave mode.

SPI Controller supports the following features:

- Support Motorola SPI,TI Synchronous Serial Protocol and National Semiconductor Micro wire interface
- Support 32-bit APB bus
- Support two internal 16-bit wide and 32-location deep FIFOs, one for transmitting and the other for receiving serial data
- Support two chip select signals in master mode
- Support 4,8,16 bit serial data transfer
- Support configurable interrupt polarity
- Support asynchronous APB bus and SPI clock
- Support master and slave mode
- Support DMA handshake interface and configurable DMA water level
- Support transmit FIFO empty, underflow, receive FIFO full, overflow, interrupt and all interrupts can be masked
- Support configurable water level of transmit FIFO empty and receive FIFO full interrupt
- Support combine interrupt output
- Support up to half of SPI clock frequency transfer in master mode and one sixth of SPI clock frequency transfer in slave mode
- Support full and half duplex mode transfer
- Stop transmitting SCLK if transmit FIFO is empty or receive FIFO is full in master mode
- Support configurable delay from chip select active to SCLK active in master mode
- Support configurable period of chip select inactive between two parallel data in master mode
- Support big and little endian, MSB and LSB first transfer
- Support two 8-bit audio data store together in one 16-bit wide location
- Support sample RXD 0~3 SPI clock cycles later
- Support configurable SCLK polarity and phase
- Support fix and incremental address access to transmit and receive FIFO

24.2 Block Diagram

The SPI Controller comprises with:

- AMBA APB interface and DMA Controller Interface
- Transmit and receive FIFO controllers and an FSM controller
- Register block
- Shift control and interrupt

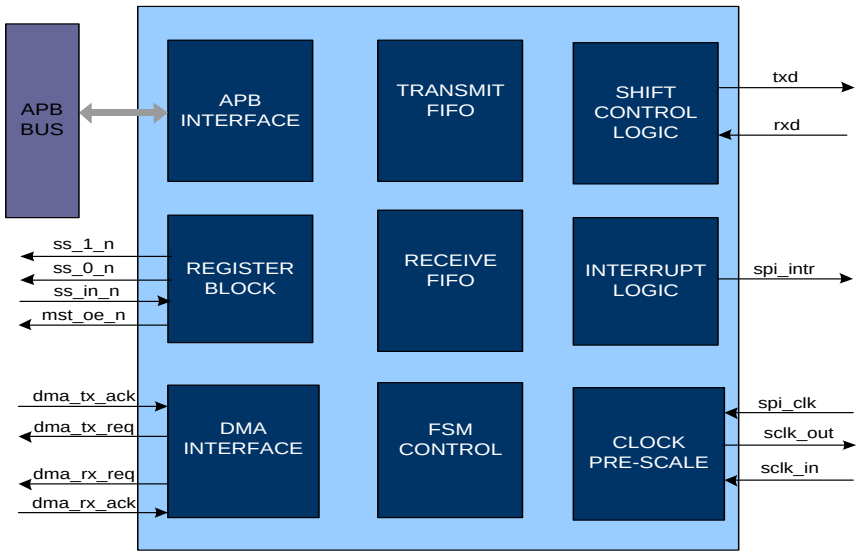


Fig. 24-1 SPI Controller Block diagram

APB INTERFACE

The host processor accesses data, control, and status information on the SPI through the APB interface. The SPI supports APB data bus widths of 32 bits and 8 or 16 bits when reading or writing internal FIFO if data frame size(SPI_CTRL0[1:0]) is set to 8 bits.

DMA INTERFACE

This block has a handshaking interface to a DMA Controller to request and control transfers. The APB bus is used to perform the data transfer to or from the DMA Controller.

FIFO LOGIC

For transmit and receive transfers, data transmitted from the SPI to the external serial device is written into the transmit FIFO. Data received from the external serial device into the SPI is pushed into the receive FIFO. Both fifos are 32x16bits.

FSM CONTROL

Control the state's transformation of the design.

REGISTER BLOCK

All registers in the SPI are addressed at 32-bit boundaries to remain consistent with the APB bus. Where the physical size of any register is less than 32-bits wide, the upper unused bits of the 32-bit boundary are reserved. Writing to these bits has no effect; reading from these bits returns 0.

SHIFT CONTROL

Shift control logic shift the data from the transmit fifo or to the receive fifo. This logic automatically right-justifies receive data in the receive FIFO buffer.

INTERRUPT CONTROL

The SPI supports combined and individual interrupt requests, each of which can be masked. The combined interrupt request is the ORed result of all other SPI interrupts after masking.

24.3 Function Description

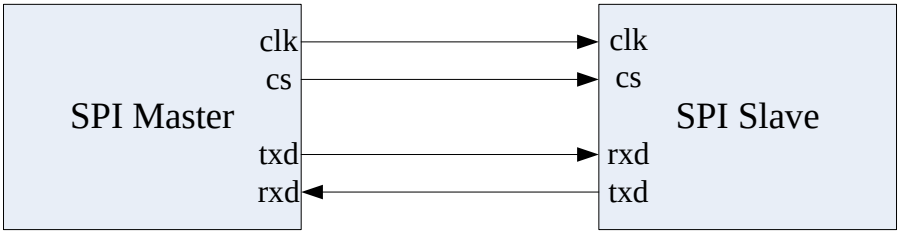


Fig. 24-2 SPI Master and Slave Interconnection

The SPI controller support dynamic switching between master and slave in a system. The diagram show how the SPI controller connects with other SPI devices.

Operation Modes

The SPI can be configured in the following two fundamental modes of operation: Master Mode when SPI_CTRLR0 [20] is 1'b0, Slave Mode when SPI_CTRLR0 [20] is 1'b1.

Transfer Modes

The SPI operates in the following three modes when transferring data on the serial bus.

1). Transmit and Receive

When SPI_CTRLR0 [19:18]== 2'b00, both transmit and receive logic are valid.

2).Transmit Only

When SPI_CTRLR0 [19:18] == 2'b01, the receive data are invalid and should not be stored in the receive FIFO.

3).Receive Only

When SPI_CTRLR0 [19:18]== 2'b10, the transmit data are invalid.

Clock Ratios

A summary of the frequency ratio restrictions between the bit-rate clock (sclk_out/sclk_in) and the SPI peripheral clock (spi_clk) are described as,

When SPI Controller works as master, the F_{spi_clk}>= 2 × (maximum F_{sclk_out})

When SPI Controller works as slave, the F_{spi_clk}>= 6 × (maximum F_{sclk_in})

With the SPI, the clock polarity (SCPOL) configuration parameter determines whether the inactive state of the serial clock is high or low. To transmit data, both SPI peripherals must have identical serial clock phase (SCPH) and clock polarity (SCPOL) values. The data frame can be 4/8/16 bits in length.

When the configuration parameter SCPH = 0, data transmission begins on the falling edge of the slave select signal. The first data bit is captured by the master and slave peripherals on the first edge of the serial clock; therefore, valid data must be present on the txd and rxd lines prior to the first serial clock edge. The following two figures show a timing diagram for a single SPI data transfer with SCPH = 0. The serial clock is shown for configuration parameters SCPOL = 0 and SCPOL = 1.

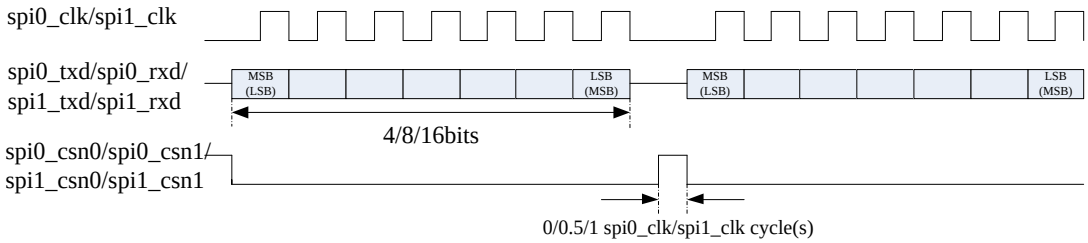


Fig. 24-3 SPI Format (SCPH=0 SCPOL=0)

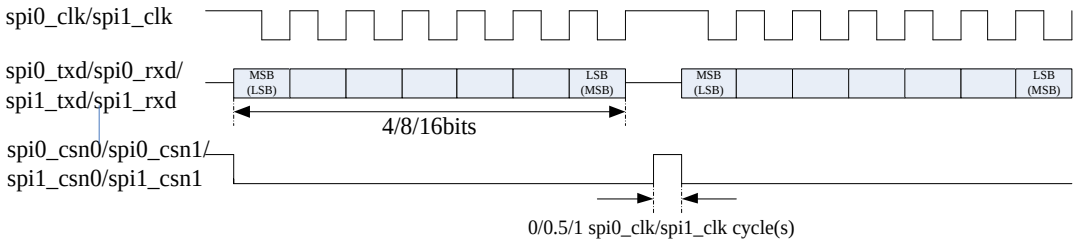


Fig. 24-4 SPI Format (SCPH=0 SCPOL=1)

When the configuration parameter SCPH = 1, both master and slave peripherals begin transmitting data on the first serial clock edge after the slave select line is activated. The first data bit is captured on the second (trailing) serial clock edge. Data are propagated by the master and slave peripherals on the leading edge of the serial clock. During continuous data frame transfers, the slave select line may be held active-low until the last bit of the last frame has been captured. The following two figures show the timing diagram for the SPI format when the configuration parameter SCPH = 1.

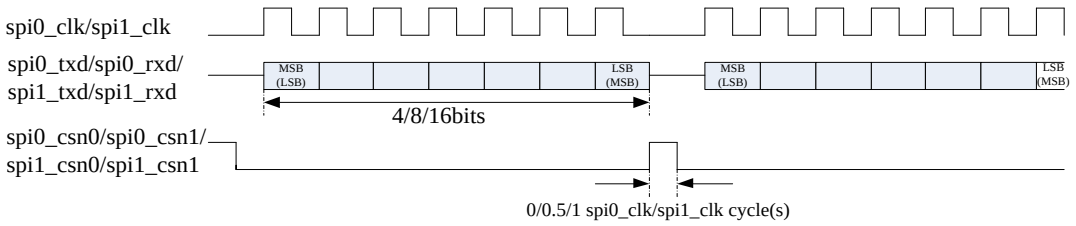


Fig. 24-5 SPI Format (SCPH=1 SCPOL=0)

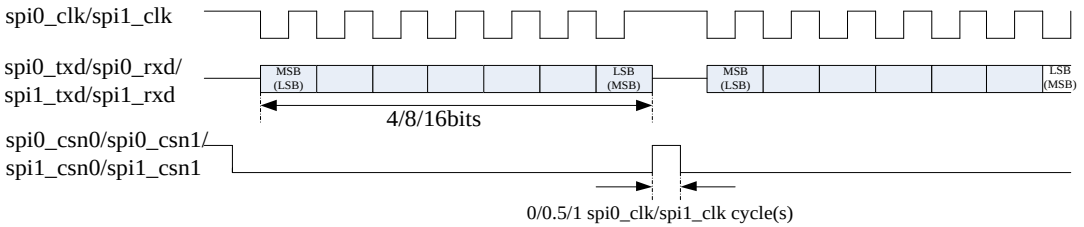


Fig. 24-6 SPI Format (SCPH=1 SCPOL=1)

24.4 Register Description

24.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
SPI_CTRLR0	0x0000	W	0x00000002	Control Register 0
SPI_CTRLR1	0x0004	W	0x00000000	Control Register 1
SPI_ENR	0x0008	W	0x00000000	SPI Enable
SPI_SER	0x000c	W	0x00000000	Slave Enable Register
SPI_BAUDR	0x0010	W	0x00000000	Baud Rate Select
SPI_TXFTLR	0x0014	W	0x00000000	Transmit FIFO Threshold Level
SPI_RXFTLR	0x0018	W	0x00000000	Receive FIFO Threshold Level
SPI_TXFLR	0x001c	W	0x00000000	Transmit FIFO Level
SPI_RXFLR	0x0020	W	0x00000000	Receive FIFO Level
SPI_SR	0x0024	W	0x0000000c	SPI Status
SPI_IPR	0x0028	W	0x00000000	Interrupt Polarity
SPI_IMR	0x002c	W	0x00000000	Interrupt Mask
SPI_ISR	0x0030	W	0x00000000	Interrupt Status
SPI_RISR	0x0034	W	0x00000001	Raw Interrupt Status
SPI_ICR	0x0038	W	0x00000000	Interrupt Clear
SPI_DMACR	0x003c	W	0x00000000	DMA Control
SPI_DMATDLR	0x0040	W	0x00000000	DMA Transmit Data Level
SPI_DMARDLR	0x0044	W	0x00000000	DMA Receive Data Level
SPI_TXDR	0x0400	W	0x00000000	Transmit FIFO Data
SPI_RXDR	0x0800	W	0x00000000	Receive FIFO Data

*Notes:Size:**B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access*

24.4.2 Detail Register Description

SPI_CTRLR0

Address: Operational Base + offset (0x0000)

Control Register 0

Bit	Attr	Reset Value	Description
31:22	RO	0x0	reserved
21	RW	0x0	MTM Microwire Transfer Mode Valid when frame format is set to National Semiconductors Microwire. 1'b0: non-sequential transfer 1'b1: sequential transfer
20	RW	0x0	OPM Operation Mode 1'b0: Master Mode 1'b1: Slave Mode
19:18	RW	0x0	XFM Transfer Mode 2'b00 :Transmit & Receive 2'b01 : Transmit Only 2'b10 : Receive Only 2'b11 :reserved
17:16	RW	0x0	FRF Frame Format 2'b00: Motorola SPI 2'b01: Texas Instruments SSP 2'b10: National Semiconductors Microwire 2'b11 : Reserved
15:14	RW	0x0	RSD Rxd Sample Delay When SPI is configured as a master, if the rxd data cannot be sampled by the sclk_out edge at the right time, this register should be configured to define the number of the spi_clk cycles after the active sclk_out edge to sample rxd data later when SPI works at high frequency. 2'b00:do not delay 2'b01:1 cycle delay 2'b10:2 cycles delay 2'b11:3 cycles delay
13	RW	0x0	BHT Byte and Halfword Transform Valid when data frame size is 8bit. 1'b0:apb 16bit write/read, spi 8bit write/read 1'b1: apb 8bit write/read, spi 8bit write/read
12	RW	0x0	FBM First Bit Mode 1'b0:first bit is MSB 1'b1:first bit is LSB

Bit	Attr	Reset Value	Description
11	RW	0x0	EM Endian Mode Serial endian mode can be configured by this bit. Apb endian mode is always little endian. 1'b0:little endian 1'b1:big endian
10	RW	0x0	SSD ss_n to sclk_out delay Valid when the frame format is set to Motorola SPI and SPI used as a master. 1'b0: the period between ss_n active and sclk_out active is half sclk_out cycles. 1'b1: the period between ss_n active and sclk_out active is one sclk_out cycle.
9:8	RW	0x0	CSM Chip Select Mode Valid when the frame format is set to Motorola SPI and SPI used as a master. 2'b00: ss_n keep low after every frame data is transferred. 2'b01:ss_n be high for half sclk_out cycles after every frame data is transferred. 2'b10: ss_n be high for one sclk_out cycle after every frame data is transferred. 2'b11:reserved
7	RW	0x0	SCPOL Serial Clock Polarity Valid when the frame format is set to Motorola SPI. 1'b0: Inactive state of serial clock is low 1'b1: Inactive state of serial clock is high
6	RW	0x0	SCPH Serial Clock Phase Valid when the frame format is set to Motorola SPI. 1'b0: Serial clock toggles in middle of first data bit 1'b1: Serial clock toggles at start of first data bit

Bit	Attr	Reset Value	Description
5:2	RW	0x0	CFS Control Frame Size Selects the length of the control word for the Microwire frame format. 4'b0000~0010:reserved 4'b0011:4-bit serial data transfer 4'b0100:5-bit serial data transfer 4'b0101:6-bit serial data transfer 4'b0110:7-bit serial data transfer 4'b0111:8-bit serial data transfer 4'b1000:9-bit serial data transfer 4'b1001:10-bit serial data transfer 4'b1010:11-bit serial data transfer 4'b1011:12-bit serial data transfer 4'b1100:13-bit serial data transfer 4'b1101:14-bit serial data transfer 4'b1110:15-bit serial data transfer 4'b1111:16-bit serial data transfer
1:0	RW	0x2	DFS Data Frame Size Selects the data frame length. 2'b00:4bit data 2'b01:8bit data 2'b10:16bit data 2'b11:reserved

SPI_CTRLR1
Address: Operational Base + offset (0x0004)
Control Register 1

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	NDM Number of Data Frames When Transfer Mode is receive only, this register field sets the number of data frames to be continuously received by the SPI. The SPI continues to receive serial data until the number of data frames received is equal to this register value plus 1, which enables you to receive up to 64 KB of data in a continuous transfer.

SPI_ENR
Address: Operational Base + offset (0x0008)
SPI Enable

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved

Bit	Attr	Reset Value	Description
0	RW	0x0	ENR SPI Enable 1'b1: Enable all SPI operations. 1'b0: Disable all SPI operations Transmit and receive FIFO buffers are cleared when the device is disabled.

SPI_SER
Address: Operational Base + offset (0x000c)
Slave Enable Register

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1	RW	0x0	SER1 Slave 1 Select Enable 1'b1: Enable chip select 1 1'b0: Disable chip select 1 This register is valid only when SPI is configured as a master device.
0	RW	0x0	SER0 Slave Select Enable 1'b1: Enable chip select 0 1'b0: Disable chip select 0 This register is valid only when SPI is configured as a master device.

SPI_BAUDR
Address: Operational Base + offset (0x0010)
Baud Rate Select

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved

Bit	Attr	Reset Value	Description
15:0	RW	0x0000	BAUDR Baud Rate Select SPI Clock Divider. This register is valid only when the SPI is configured as a master device. The LSB for this field is always set to 0 and is unaffected by a write operation, which ensures an even value is held in this register. If the value is 0, the serial output clock (sclk_out) is disabled. The frequency of the sclk_out is derived from the following equation: Fsclk_out = Fspi_clk/ SCKDV Where SCKDV is any even value between 2 and 65534. For example: for Fspi_clk = 3.6864MHz and SCKDV =2 Fsclk_out = 3.6864/2= 1.8432MHz

SPI_TXFTLR
Address: Operational Base + offset (0x0014)
Transmit FIFO Threshold Level

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4:0	RW	0x00	TXFTLR Transmit FIFO Threshold Level When the number of transmit FIFO entries is less than or equal to this value, the transmit FIFO empty interrupt is triggered.

SPI_RXFTLR
Address: Operational Base + offset (0x0018)
Receive FIFO Threshold Level

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4:0	RW	0x00	RXFTLR Receive FIFO Threshold Level When the number of receive FIFO entries is greater than or equal to this value + 1, the receive FIFO full interrupt is triggered.

SPI_TXFLR
Address: Operational Base + offset (0x001c)
Transmit FIFO Level

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5:0	RO	0x00	TXFLR Transmit FIFO Level Contains the number of valid data entries in the transmit FIFO.

SPI_RXFLR
Address: Operational Base + offset (0x0020)
Receive FIFO Level

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5:0	RO	0x00	RXFLR Receive FIFO Level Contains the number of valid data entries in the receive FIFO.

SPI_SR
Address: Operational Base + offset (0x0024)
SPI Status

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4	RO	0x0	RFF Receive FIFO Full 1'b0: Receive FIFO is not full 1'b1: Receive FIFO is full
3	RO	0x1	RFE Receive FIFO Empty 1'b0: Receive FIFO is not empty 1'b1: Receive FIFO is empty
2	RO	0x1	TFE Transmit FIFO Empty 1'b0: Transmit FIFO is not empty 1'b1: Transmit FIFO is empty
1	RO	0x0	TFF Transmit FIFO Full 1'b0: Transmit FIFO is not full 1'b1: Transmit FIFO is full
0	RO	0x0	BSF SPI Busy Flag When set, indicates that a serial transfer is in progress; when cleared indicates that the SPI is idle or disabled. 1'b0: SPI is idle or disabled 1'b1: SPI is actively transferring data

SPI_IPR
Address: Operational Base + offset (0x0028)
Interrupt Polarity

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved

Bit	Attr	Reset Value	Description
0	RW	0x0	IPR Interrupt Polarity Interrupt Polarity Register 1'b0:Active Interrupt Polarity Level is HIGH 1'b1: Active Interrupt Polarity Level is LOW

SPI_IMR
Address: Operational Base + offset (0x002c)
Interrupt Mask

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4	RW	0x0	RFFIM Receive FIFO Full Interrupt Mask 1'b0: spi_rxf_intr interrupt is masked 1'b1: spi_rxf_intr interrupt is not masked
3	RW	0x0	RFOIM Receive FIFO Overflow Interrupt Mask 1'b0: spi_rxo_intr interrupt is masked 1'b1: spi_rxo_intr interrupt is not masked
2	RW	0x0	RFUIM Receive FIFO Underflow Interrupt Mask 1'b0: spi_rxu_intr interrupt is masked 1'b1: spi_rxu_intr interrupt is not masked
1	RW	0x0	TFOIM Transmit FIFO Overflow Interrupt Mask 1'b0: spi_txo_intr interrupt is masked 1'b1: spi_txo_intr interrupt is not masked
0	RW	0x0	TFEIM Transmit FIFO Empty Interrupt Mask 1'b0: spi_txe_intr interrupt is masked 1'b1: spi_txe_intr interrupt is not masked

SPI_ISR
Address: Operational Base + offset (0x0030)
Interrupt Status

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4	RO	0x0	RFFIS Receive FIFO Full Interrupt Status 1'b0: spi_rxf_intr interrupt is not active after masking 1'b1: spi_rxf_intr interrupt is full after masking
3	RO	0x0	RFOIS Receive FIFO Overflow Interrupt Status 1'b0: spi_rxo_intr interrupt is not active after masking 1'b1: spi_rxo_intr interrupt is active after masking

Bit	Attr	Reset Value	Description
2	RO	0x0	RFUIS Receive FIFO Underflow Interrupt Status 1'b0: spi_rxu_intr interrupt is not active after masking 1'b1: spi_rxu_intr interrupt is active after masking
1	RO	0x0	TFOIS Transmit FIFO Overflow Interrupt Status 1'b0: spi_txo_intr interrupt is not active after masking 1'b1: spi_txo_intr interrupt is active after masking
0	RO	0x0	TFEIS Transmit FIFO Empty Interrupt Status 1'b0: spi_txe_intr interrupt is not active after masking 1'b1: spi_txe_intr interrupt is active after masking

SPI_RISR
Address: Operational Base + offset (0x0034)
Raw Interrupt Status

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4	RO	0x0	RFFRIS Receive FIFO Full Raw Interrupt Status 1'b0: spi_rxf_intr interrupt is not active prior to masking 1'b1: spi_rxf_intr interrupt is full prior to masking
3	RO	0x0	RFORIS Receive FIFO Overflow Raw Interrupt Status 1'b0 = spi_rxo_intr interrupt is not active prior to masking 1'b1 = spi_rxo_intr interrupt is active prior to masking
2	RO	0x0	RFURIS Receive FIFO Underflow Raw Interrupt Status 1'b0: spi_rxu_intr interrupt is not active prior to masking 1'b1: spi_rxu_intr interrupt is active prior to masking
1	RO	0x0	TFORIS Transmit FIFO Overflow Raw Interrupt Status 1'b0: spi_txo_intr interrupt is not active prior to masking 1'b1: spi_txo_intr interrupt is active prior to masking
0	RO	0x1	TFERIS Transmit FIFO Empty Raw Interrupt Status 1'b0: spi_txe_intr interrupt is not active prior to masking 1'b1: spi_txe_intr interrupt is active prior to masking

SPI_ICR
Address: Operational Base + offset (0x0038)
Interrupt Clear

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved

Bit	Attr	Reset Value	Description
3	WO	0x0	CTFOI Clear Transmit FIFO Overflow Interrupt Write 1 to Clear Transmit FIFO Overflow Interrupt
2	WO	0x0	CRFOI Clear Receive FIFO Overflow Interrupt Write 1 to Clear Receive FIFO Overflow Interrupt
1	WO	0x0	CRFUI Clear Receive FIFO Underflow Interrupt Write 1 to Clear Receive FIFO Underflow Interrupt
0	WO	0x0	CCI Clear Combined Interrupt Write 1 to Clear Combined Interrupt

SPI_DMACR
Address: Operational Base + offset (0x003c)
DMA Control

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1	RW	0x0	TDE Transmit DMA Enable 1'b0: Transmit DMA disabled 1'b1: Transmit DMA enabled
0	RW	0x0	RDE Receive DMA Enable 1'b0: Receive DMA disabled 1'b1: Receive DMA enabled

SPI_DMATDLR
Address: Operational Base + offset (0x0040)
DMA Transmit Data Level

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4:0	RW	0x00	TDL Transmit Data Level This bit field controls the level at which a DMA request is made by the transmit logic. It is equal to the watermark level; that is, the dma_tx_req signal is generated when the number of valid data entries in the transmit FIFO is equal to or below this field value, and Transmit DMA Enable (DMACR[1]) = 1.

SPI_DMARDLR
Address: Operational Base + offset (0x0044)
DMA Receive Data Level

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved

Bit	Attr	Reset Value	Description
4:0	RW	0x00	RDL Receive Data Level This bit field controls the level at which a DMA request is made by the receive logic. The watermark level = DMARDL+1; that is, dma_rx_req is generated when the number of valid data entries in the receive FIFO is equal to or above this field value + 1, and Receive DMA Enable(DMACR[0])=1.

SPI_TXDR
Address: Operational Base + offset (0x0048)
Transmit FIFO Data

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	WO	0x0000	TXDR Transimt FIFO Data Register. When it is written to, data are moved into the transmit FIFO.

SPI_RXDR
Address: Operational Base + offset (0x004c)
Receive FIFO Data

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	RXDR Receive FIFO Data Register. When the register is read, data in the receive FIFO is accessed.

24.5 Interface Description

Table 24-1 SPI interface description

Module Pin	Direction	Pad Name	IOMUX Setting
spi0_sclk	I/O	IO_MACrxd0_SPI0NORCODECclk_GMACgpio3a6	GRF_GPI03A_IOMUX[13:12]=2'b10
spi0_rxd	I	IO_MACtxd0_SPI0NORCODECrxd_GMACgpio3a4	GRF_GPI03A_IOMUX[9:8]=2'b10
spi0_txd	O	IO_MACtxd1_SPI0NORCODECtxd_GMACgpio3a5	GRF_GPI03A_IOMUX[11:10]=2'b10
spi0_csn0	I/O	IO_MACrxd1_SPI0NORCODECcsn0_GMACgpio3a7	GRF_GPI03A_IOMUX[15:14]=2'b10
spi0_csn1	O	IO_MACmdc_SPI0NORCODECcsn1_GMACgpio3b0	GRF_GPI03B_IOMUX[1:0]=2'b10
spi1_sclk	I/O	IO_PMUM0JTAGtck_SPI1ECclk_PMU1830gpio1b1	PMUGRF_GPIO1B_IOMUX[3:2]=2'b10
spi1_rxd	I	IO_UART4M0sin_SPI1ECrxd_PMU1830gpio1a7	PMUGRF_GPIO1A_IOMUX[15:14]=2'b10
spi1_txd	O	IO_UART4M0sout_SPI1ECtxd_PMU1830gpio1b0	PMUGRF_GPIO1B_IOMUX[1:0]=2'b10
spi1_csn0	I/O	IO_PMUM0JTAGtms_SPI1ECcsn	PMUGRF_GPIO1B_IOMUX[5:4]

Module Pin	Direction	Pad Name	IOMUX Setting
		0_PMU1830gpio1b2	=2'b10
spi2_sclk	I/O	IO_SPI2TPMclk_VOPden_CIFclkouta_BT656gpio2b3	GRF_GPI02B_IOMUX[7:6]=2'b01
spi2_rxd	I	IO_SPI2TPMrxd_I2C6TPMsda_CIFhref_BT656gpio2b1	GRF_GPI02B_IOMUX[3:2]=2'b01
spi2_txd	O	IO_SPI2TPMtxd_I2C6TPMscl_CIFclkin_BT656gpio2b2	GRF_GPI02B_IOMUX[5:4]=2'b01
spi2_csn0	I/O	IO_SPI2TPMcsn0_BT656gpio2b4	GRF_GPI02B_IOMUX[9:8]=2'b01
spi3_sclk	I/O	IO_SPI3PMUclk_PMU1830gpio1c1	PMUGRF_GPIO1C_IOMUX[3:2]=2'b10
spi3_rxd	I	IO_SPI3PMUrxd_I2C0PMUsda_PMU1830gpio1b7	PMUGRF_GPIO1B_IOMUX[15:14]=2'b10
spi3_txd	O	IO_SPI3PMUtxd_I2C0PMUscl_PMU1830gpio1c0	PMUGRF_GPIO1C_IOMUX[1:0]=2'b10
spi3_csn0	I/O	IO_SPI3PMUcsn0_PMU1830gpio1c2	PMUGRF_GPIO1C_IOMUX[5:4]=2'b10
spi4_sclk	I/O	IO_MACrxd2_SPI4EXPclk_TRACEdata14_GMACgpio3a2	GRF_GPI03A_IOMUX[5:4]=2'b10
spi4_rxd	I	IO_MACtxd2_SPI4EXPrxd_TRACEdata12_GMACgpio3a0	GRF_GPI03A_IOMUX[1:0]=2'b10
spi4_txd	O	IO_MACtxd3_SPI4EXPtxd_TRACEdata13_GMACgpio3a1	GRF_GPI03A_IOMUX[3:2]=2'b10
spi4_csn0	I/O	IO_MACrxd3_SPI4EXPcsn0_TRACEdata15_GMACgpio3a3	GRF_GPI03A_IOMUX[7:6]=2'b10
spi5_sclk	I/O	IO_SDIOdata2_SPI5EXPPLUSclk_WIFIBTgpio2c6	GRF_GPI02C_IOMUX[13:12]=2'b10
spi5_rxd	I	IO_SDIOdata0_SPI5EXPPLUSrxd_WIFIBTgpio2c4	GRF_GPI02C_IOMUX[9:8]=2'b10
spi5_txd	O	IO_SDIOdata1_SPI5EXPPLUStxd_WIFIBTgpio2c5	GRF_GPI02C_IOMUX[11:10]=2'b10
spi5_csn0	I/O	IO_SDIOdata3_SPI5EXPPLUScsn0_WIFIBTgpio2c7	GRF_GPI02C_IOMUX[15:14]=2'b10

Notes: I=input, O=output, I/O=input/output, bidirectional. spi_csn1 can only be used in master mode

24.6 Application Notes

Clock Ratios

A summary of the frequency ratio restrictions between the bit-rate clock (sclk_out/sclk_in) and the SPI peripheral clock (spi_clk) are described as,

When SPI Controller works as master, the Fspi_clk>= 2 × (maximum Fsclk_out)

When SPI Controller works as slave, the Fspi_clk>= 6 × (maximum Fsclk_in)

Master Transfer Flow

When configured as a serial-master device, the SPI initiates and controls all serial transfers. The serial bit-rate clock, generated and controlled by the SPI, is driven out on the sclk_out line. When the SPI is disabled (SPI_ENR = 0), no serial transfers can occur and sclk_out is held in "inactive" state, as defined by the serial protocol under which it operates.

Slave Transfer Flow

When the SPI is configured as a slave device, all serial transfers are initiated and controlled by the serial bus master.

When the SPI serial slave is selected during configuration, it enables its txd data onto the serial bus. All data transfers to and from the serial slave are regulated on the serial clock line (sclk_in), driven from the serial-master device. Data are propagated from the serial slave on one edge of the serial clock line and sampled on the opposite edge.

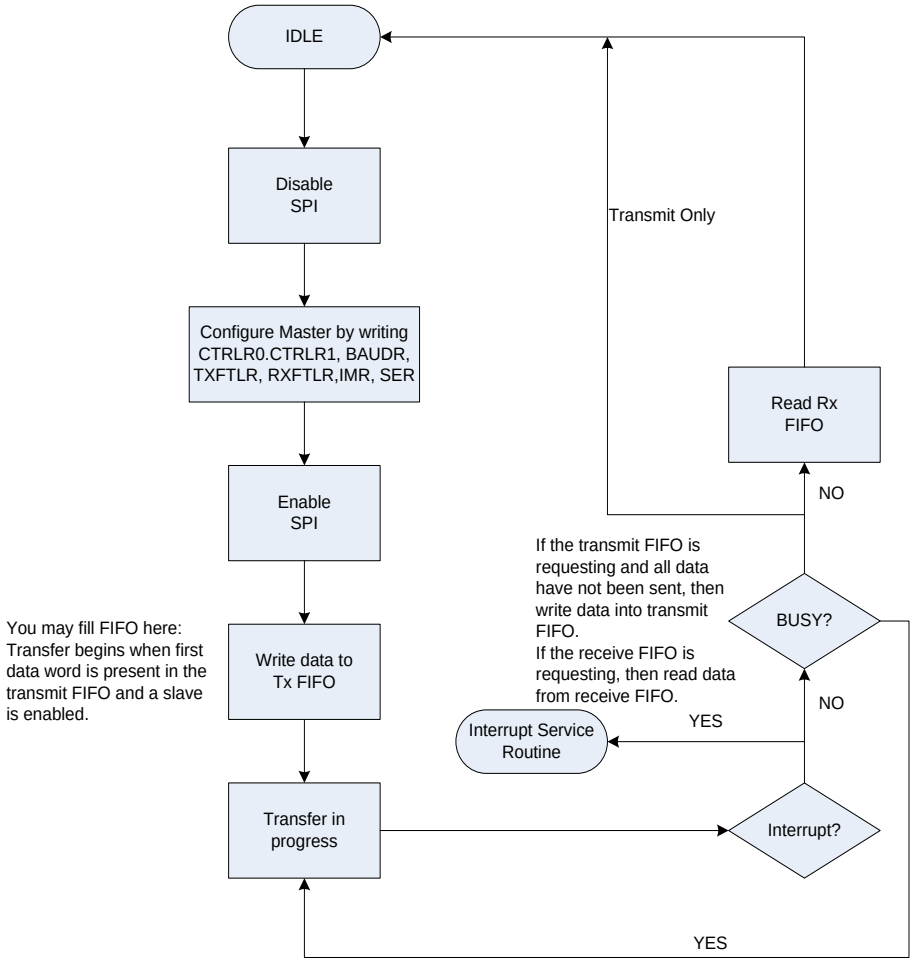


Fig. 24-7 SPI Master transfer flow diagram

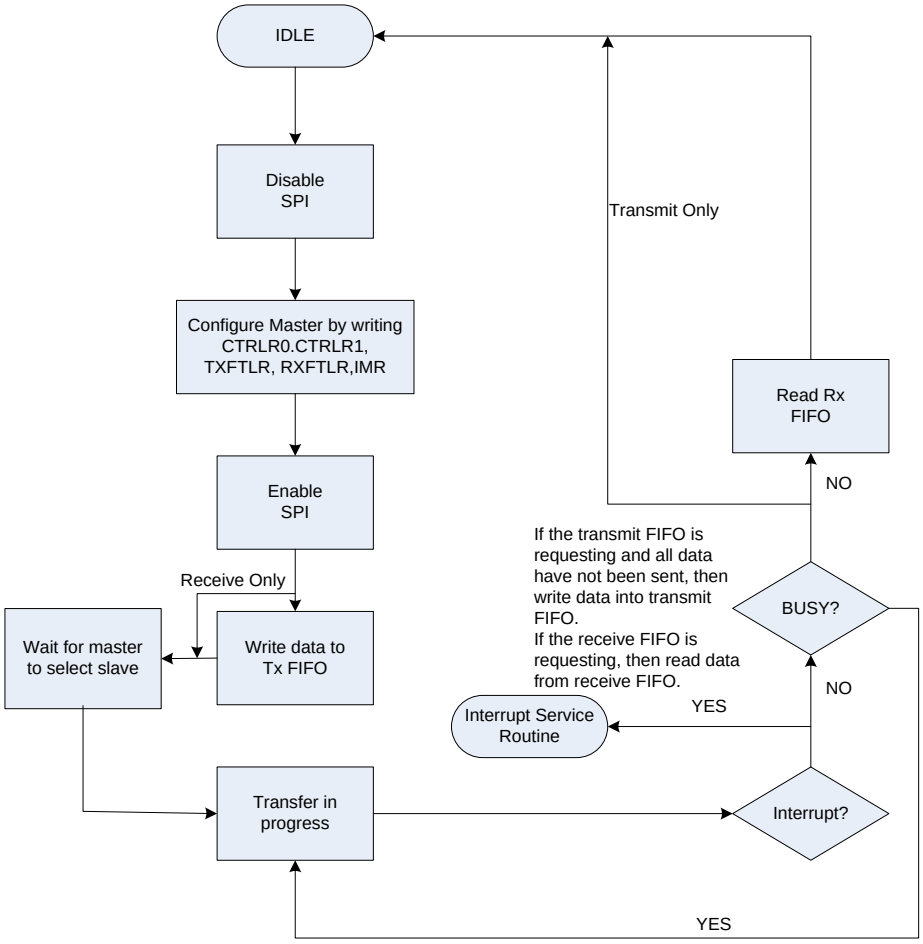


Fig. 24-8 SPI Slave transfer flow diagram

Chapter 25 SPDIF Transmitter

25.1 Overview

The SPDIF transmitter is a self-clocking, serial and unidirectional interface for the interconnection of digital audio equipment in consumer and professional applications which uses linear PCM coded audio samples.
When used in professional application, the interface is primarily intended to carry monophonic or stereophonic programmes at a 48 kHz sampling frequency with a resolution of up to 24bits per sample. It may alternatively be used to carry signals sampled at 32 kHz or 44.1 kHz.
When used in consumer application, the interface is primarily intended to carry stereophonic programmes with a resolution of up to 20 bits per sample, an extension to 24 bits per sample being possible.
When used for other purposes, the interface is primarily intended to carry audio data coded other than linear PCM coded audio samples. Provision is also made to allow the interface to carry data related to computer software or signals coded using non-linear PCM. The maximum sample frequency can be up to 768 kHz for the non-linear PCM mode.
In all cases, the clock references and auxiliary information are transmitted along with the programme.

- Supports one internal 32-bit wide and 32-location deep sample data buffer
- Supports two 16-bit audio data store together in one 32-bit wide location
- Supports AHB bus interface
- Supports biphase format stereo audio data output
- Supports DMA handshake interface and configurable DMA water level
- Supports sample data buffer empty, block terminate and user data interrupt
- Supports combine interrupt output
- Supports 16 to 31 bit audio data left or right justified in 32-bit wide sample data buffer
- Support 16, 20, 24 bits audio data transfer in linear PCM mode
- Support non-linear PCM transfer

25.2 Block Diagram

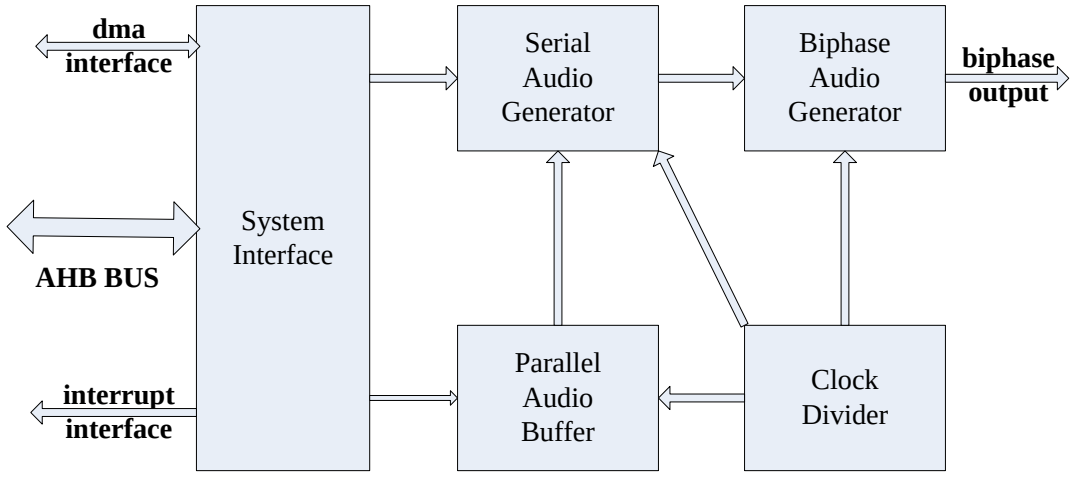


Fig. 25-1 SPDIF transmitter Block Diagram

System Interface
The system interface implements the AHB slave operation. It contains not only control registers of transmitters and receiver inside but also interrupt and DMA handshake interface.
Clock Divider
The Clock Divider implements clock generation function. The input source clock to the module is MCLK. By the divider of the module, the clock divider generates work clock for digital audio data transformation.
Parallel Audio Buffer

The Parallel Audio Buffer is the buffer to store transmitted audio data. The size of the FIFO is 32bits x 32.

Serial Audio Converter

The Serial Audio Converter reads parallel audio data from the Parallel Audio Buffer and converts it to serial audio data.

Biphase Audio Generator

The Biphase Audio Generator reads serial audio data from the Serial Audio Converter and generates biphase audio data based on IEC-60958 standard.

25.3 Function description

25.3.1 Frame Format

A frame is uniquely composed of two sub-frames. For linear coded audio applications, the rate of transmission of frames corresponds exactly to the source sampling frequency.
In the 2-channel operation mode, the samples taken from both channels are transmitted by time multiplexing in consecutive sub-frames. The first sub-frame(left channel in stereophonic operation and primary channel in monophonic operation) normally use preamble M. However, the preamble is changed to preamble B once every 192 frame to identify the start of the block structure used to organize the channel status information. The second sub-frame (right in stereophonic operation and secondary channel in monophonic operation) always use preamble W.

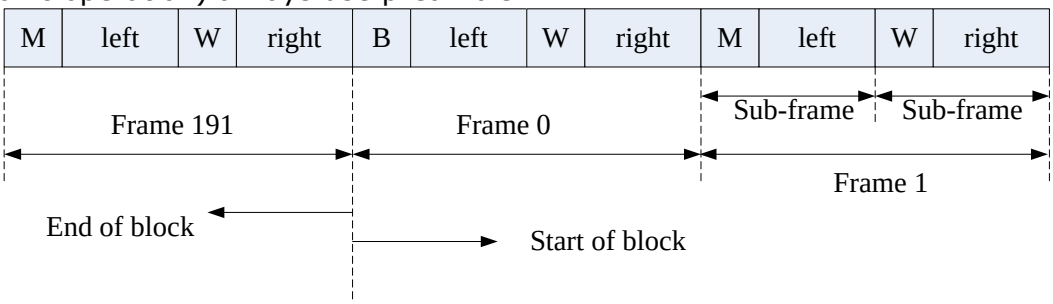


Fig. 25-2 SPDIF Frame Format

In the single channel operation mode in a professional application, the frame format is the same as in the 2-channel mode. Data is carried only in the first sub-frame and may be duplicated in the second sub-frame. If the second sub-frame is not carrying duplicate data, then time slot 28 (validity flag) shall be set to logical '1' (not valid).

25.3.2 Sub-frame Format

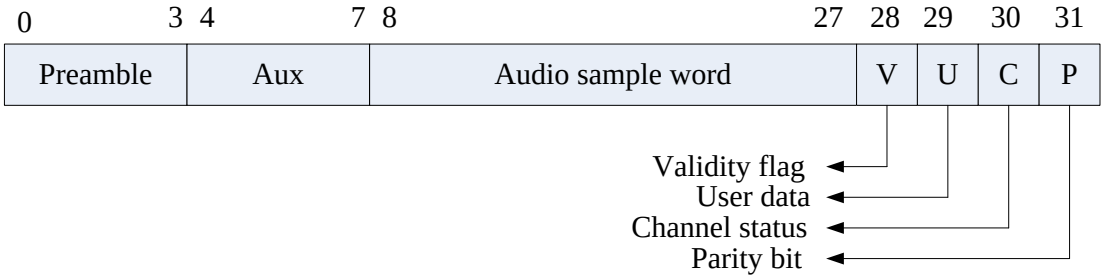


Fig. 25-3 SPDIF Sub-frame Format

Each sub-frame is divided into 32 time slots, numbered from 0 to 31. Time slot 0 to 3 carries one of the three permitted preambles. Time slot 4 to 27 carry the audio sample word in linear 2's complement representation. The MSB is carried by time slot 27. When a 24-bit coding range is used, the LSB is in time slot 4. When a 20-bit coding range is used, time slot 8 to 27 carry the audio sample word with the LSB in time slot 8.Time slot 4 to 7 may be used for other application. Under these circumstances, the bits in the time slot 4 to 7 are designated auxiliary sample bits.
If the source provides fewer bits than the interface allows (either 24 or 20), the unused LSBs are set to a logical '0'. For a non-linear PCM audio application or a data application the main data field may carry any other information. Time slot 28 carries the validity flag associated with the main data field. Time slot 29 carries 1 bit of the user data associated with the audio channel transmitted in the same sub-frame. Time slot 30 carries one bit of

the channel status words associated with the main data field channel transmitted in the same sub-frame. Time slot 31 carries a parity bit such that time slots 4 to 31 inclusive carries an even number of ones and an even number of zeros.

25.3.3 Channel Coding

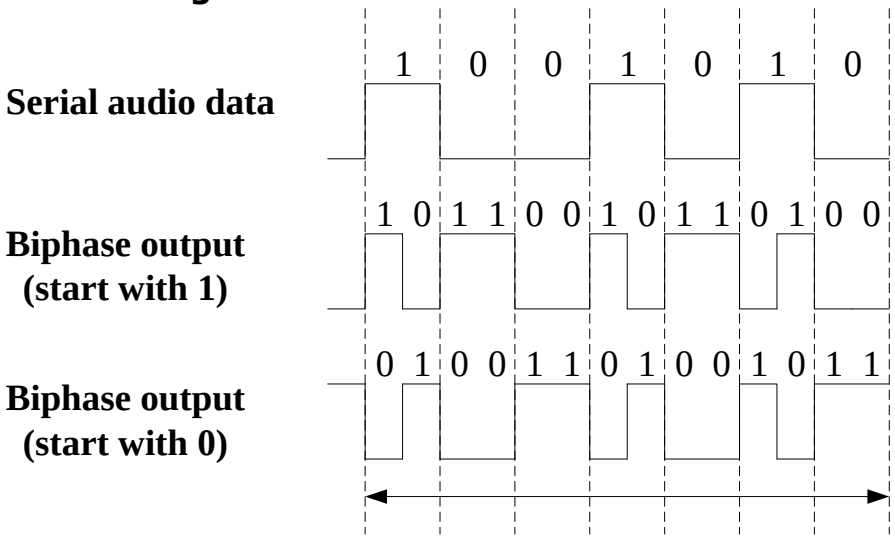


Fig. 25-4 SPDIF Channel Coding

To minimize the direct current component on the transmission line, to facilitate clock recovery from the data stream and to make the interface insensitive to the polarity of connections, time slots 4 to 31 are encoded in biphase-mark.

Each bit to be transmitted is represented by a symbol comprising two consecutive binary states. The first state of a symbol is always different from the second state of the previous symbol. The second state of the symbol is identical to the first if the bit to be transmitted is logical '0'.However, it is different from the first if the bit is logical '1'.

25.3.4 Preamble

Preambles are specific patterns providing synchronization and identification of the sub-frames and blocks.

To achieve synchronization within one sampling period and to make this process completely reliable, these patterns violate the biphase-mark code rules, thereby avoiding the possibility of data imitating the preambles.

A set of three preambles is used. These preambles are transmitted in the time allocated to four time slots (time slots 0 to 3) and are represented by eight successive states. The first state of the preamble is always different from the second state of the previous symbol.

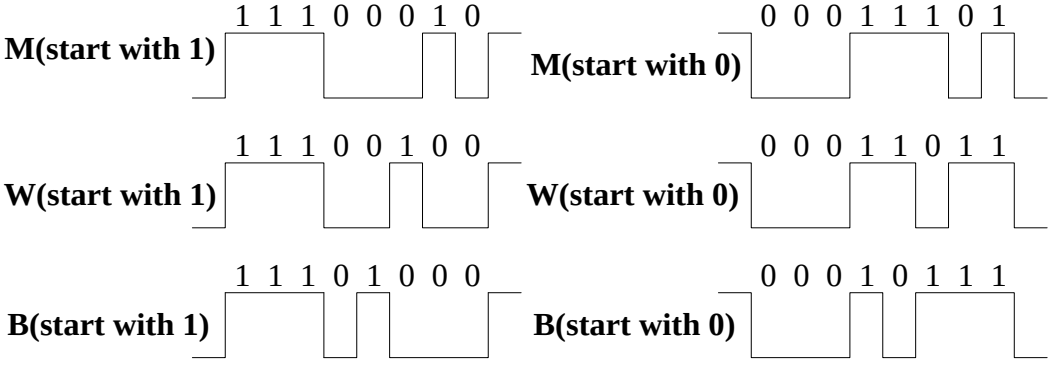


Fig. 25-5 SPDIF Preamble

Like biphase code, these preambles are dc free and provide clock recovery. They differ in at least two states from any valid biphase sequence.

25.3.5 NON-LINEAR PCM ENCODED SOURCE(IEC 61937)

The non-linear PCM encoded audio bitstream is transferred using the basic 16-bit data area of the IEC 60958subframes, i.e. in time slots 12 to 27. Each IEC 60958 frame transfers 32-bit of the non-PCM data in consumer application mode.

If the SPDIF bitstream conveys linear PCM audio, the symbol frequency is 64 times the PCM sampling frequency(32 time slots per PCM sample times two channels). If a non-linear PCM

encoded audio bitstream is conveyed by the interface, the symbol frequency is 64 times the sampling rate of the encoded audio within that bitstream. But in the case where a non-linear PCM encoded audio bitstream is conveyed by the interface containing audio with low sampling frequency, the symbol frequency is 128 times the sampling rate of the encoded audio within that bitstream.
Each data burst contains a burst-preamble consisting of four 16-bit words (Pa, Pb, Pc, Pd), followed by the burst payload which contains data of an encoded audio frame.
The burst-preamble consists of four mandatory fields. Pa and Pb represent a synchronization word. Pc gives information about the type of data and some information/control for the receiver. Pd gives the length of the burst payload, the number of bits or number of bytes according to data-type.
The four preamble words are contained in two sequential SPDIF frames. The frame beginning the data-burst contains preamble word Pa in subframe 0 and Pb in subframe 1. The next frame contains Pc in subframe 0 and Pd in subframe 1. When placed into a SPDIF subframe, the MSB of a 16-bit burst-preamble is placed into timeslot 27 and the LSB is placed into time slot 12.

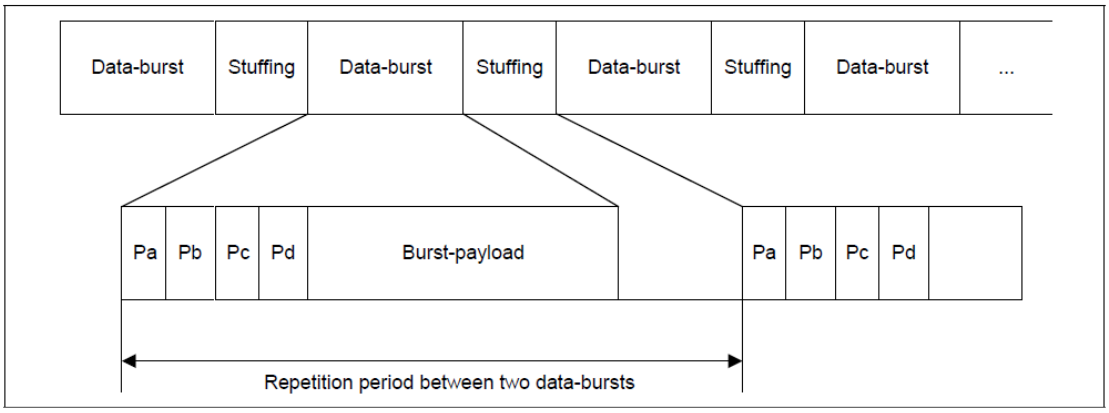


Fig. 25-6 Format of Data-burst

25.4 Register description

25.4.1 Register Summary

Name	Offset	Size	Reset Value	Description
SPDIF_CFGR	0x0000	W	0x00000000	Transfer Configuration Register
SPDIF_SDBLR	0x0004	W	0x00000000	Sample Date Buffer Level Register
SPDIF_DMACR	0x0008	W	0x00000000	DMA Control Register
SPDIF_INTCR	0x000c	W	0x00000000	Interrupt Control Register
SPDIF_INTSR	0x0010	W	0x00000000	Interrupt Status Register
SPDIF_XFER	0x0018	W	0x00000000	Transfer Start Register
SPDIF_SMPDR	0x0020	W	0x00000000	Sample Data Register
SPDIF_VLDFRn	0x0060	W	0x00000000	Validity Flag Register n
SPDIF_USRDRn	0x0090	W	0x00000000	User Data Register n
SPDIF_CHNSRn	0x00c0	W	0x00000000	Channel Status Register n
SPDIF_BURTSINFO	0x0100	W	0x00000000	Channel Burst Info Register
SPDIF_REPETTION	0x0104	W	0x00000000	Channel Repetition Register
SPDIF_BURTSINFO_SHD	0x0108	W	0x00000000	Shadow Channel Burst Info Register
SPDIF_REPETTION_SHD	0x010c	W	0x00000000	Shadow Channel Repetition Register
SPDIF_USRDR_SHDn	0x0190	W	0x00000000	Shadow User Data Register n

*Notes:Size:**B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access*

25.4.2 Detail Register Description

SPDIF_CFGR

Address: Operational Base + offset (0x0000)
Transfer Configuration Register

Bit	Attr	Reset Value	Description
31:24	RO	0x0	reserved
23:16	RW	0x00	MCD mclk divider Fmclk/Fsdo This parameter can be calculated by Fmclk/(Fs*128). Fs=the sample frequency be wanted
15:9	RO	0x0	reserved
8	RW	0x0	PCMTYPE PCM type 0: linear PCM 1: non-linear PCM
7	WO	0x0	CLR mclk domain logic clear Write 1 to clear mclk domain logic. Read return zero.
6	RW	0x0	CSE Channel status enable 0: disable 1: enable The bit should be set to 1 when the channel conveys non-linear PCM
5	RW	0x0	UDE User data enable 0: disable 1: enable
4	RW	0x0	VFE Validity flag enable 0: disable 1: enable
3	RW	0x0	ADJ audio data justified 0: Right justified 1: Left justified
2	RW	0x0	HWT Halfword word transform enable 0: disable 1: enable It is valid only when the valid data width is 16bit.
1:0	RW	0x0	VDW Valid data width 00: 16bit 01: 20bit 10: 24bit 11: reserved The valid data width is 16bit only for non-linear PCM

SPDIF_SDBLR
Address: Operational Base + offset (0x0004)
Sample Date Buffer Level Register

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5:0	RW	0x00	SDBLR Sample Date Buffer Level Register Contains the number of valid data entries in the sample data buffer.

SPDIF_DMACR
Address: Operational Base + offset (0x0008)
DMA Control Register

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5	RW	0x0	TDE Transmit DMA Enable 0: Transmit DMA disabled 1: Transmit DMA enabled
4:0	RW	0x00	TDL Transmit Data Level This bit field controls the level at which a DMA request is made by the transmit logic. It is equal to the watermark level; that is, the dma_tx_req signal is generated when the number of valid data entries in the Sample Date Buffer is equal to or below this field value

SPDIF_INTCR
Address: Operational Base + offset (0x000c)
Interrupt Control Register

Bit	Attr	Reset Value	Description
31:18	RO	0x0	reserved
17	W1C	0x0	UDTIC User Data Interrupt Clear Write '1' to clear the user data interrupt.
16	W1C	0x0	BTTIC Block/Data burst transfer finish interrupt clear Write 1 to clear the interrupt.
15:10	RO	0x0	reserved
9:5	RW	0x00	SDBT Sample Date Buffer Threshold Sample Date Buffer Threshold for empty interrupt

Bit	Attr	Reset Value	Description
4	RW	0x0	SDBEIE Sample Date Buffer empty interrupt enable 0: disable 1: enable
3	RW	0x0	BTTIE Block transfer/repetition period end interrupt enable When enabled, an interrupt will be asserted when the block transfer is finished if the channel conveys linear PCM or when the repetition period is reached if the channel conveys non-linear PCM. 0: disable 1: enable
2	RW	0x0	UDTIE User Data Interrupt 0: disable 1: enable If enabled, an interrupt will be asserted when the content of the user data register is fed into the corresponding shadow register
1:0	RO	0x0	reserved

SPDIF_INTSR
Address: Operational Base + offset (0x0010)
Interrupt Status Register

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4	RW	0x0	SDBEIS Sample Date Buffer empty interrupt status 0: inactive 1: active
3	RW	0x0	BTTIS Block/Data burst transfer interrupt status 0: inactive 1: active
2	RW	0x0	UDTIS User Data Interrupt Status 0: inactive 1: active
1:0	RO	0x0	reserved

SPDIF_XFER
Address: Operational Base + offset (0x0018)
Transfer Start Register

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved

Bit	Attr	Reset Value	Description
0	RW	0x0	XFER Transfer Start Register Transfer Start Register

SPDIF_SMPDR
Address: Operational Base + offset (0x0020)
Sample Data Register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	SMPDR Sample Data Register Sample Data Register

SPDIF_VLDFRn
Address: Operational Base + offset (0x0060)
Validity Flag Register n

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	VLDFR_SUB_1 Validity Flag Subframe 1 Validity Flag Register 0
15:0	RW	0x0000	VLDFR_SUB_0 Validity Flag Subframe 0 Validity Flag for Subframe 0

SPDIF_USRDRn
Address: Operational Base + offset (0x0090)
User Data Register n

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	USR_SUB_1 User Data Subframe 1 User Data Bit for Subframe 1
15:0	RW	0x0000	USR_SUB_0 User Data Subframe 0 User Data Bit for Subframe 0

SPDIF_CHNSRn
Address: Operational Base + offset (0x00c0)
Channel Status Register n

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	CHNSR_SUB_1 Channel Status Subframe 1 Channel Status Bit for Subframe 1
15:0	RW	0x0000	CHNSR_SUB_0 Channel Status Subframe 0 Channel Status Bit for Subframe 0

SPDIF_BURTSINFO
Address: Operational Base + offset (0x00d0)
Channel Burst Info Register

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	PD pd Preamble Pd for non-linear pcm, indicating the length of burst payload in unit of bytes or bits.
15:13	RW	0x0	BSNUM Bitstream Number This field indicates the bitstream number. Usually the bitstream number is 0.
12:8	RW	0x00	DATAINFO Data-type-dependent info This field gives the data-type-dependent info
7	RW	0x0	ERRFLAG Error Flag 0: indicates a valid burst-payload 1: indicates that the burst-payload may contain errors

Bit	Attr	Reset Value	Description
6:0	RW	0x00	DATATYPE Data type 0000000: null data 0000001: AC-3 data 0000011: Pause data 0000100: MPEG-1 layer 1 data 0000101: MPEG-1 layer 2 or 3 data or MPEG-2 without extension 0000110: MPEG-2 data with extension 0000111: MPEG-2 AAC 0001000: MPEG-2, layer-1 low sampling frequency 0001001: MPEG-2, layer-2 low sampling frequency 0001010: MPEG-2, layer-3 low sampling frequency 0001011: DTS type I 0001100: DTS type II 0001101: DTS type III 0001110: ATRAC 0001111: ATRAC 2/3 0010000: ATRAC-X 0010001: DTS type IV 0010010: WMA professional type I 0110010: WMA professional type II 1010010: WMA professional type III 1110010: WMA professional type IV 0010011: MPEG-2 AAC low sampling frequency 0110011: MPEG-2 AAC low sampling frequency 1010011: MPEG-2 AAC low sampling frequency 1110011: MPEG-2 AAC low sampling frequency 0010100: MPEG-4 AAC 0110100: MPEG-4 AAC 1010100: MPEG-4 AAC 1110100: MPEG-4 AAC 0010101: Enhanced AC-3 0010110: MAT others: reserved

SPDIF_REPETTION

Address: Operational Base + offset (0x0104)

Channel Repetition Register

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	REPETTION Repetition This define the repetition period when the channel conveys non-linear PCM

SPDIF_BURTSINFO_SHD
Address: Operational Base + offset (0x0108)
Shadow Channel Burst Info Register

Bit	Attr	Reset Value	Description
31:16	RO	0x0000	PD pd Preamble Pd for non-linear pcm, indicating the length of burst payload in unit of bytes or bits.
15:13	RO	0x0	BSNUM Bitstream Number This field indicates the bitstream number. Usually the birstream number is 0.
12:8	RO	0x00	DATAINFO Data-type-dependent info This field gives the data-type-dependent info
7	RO	0x0	ERRFLAG Error Flag 0: indicates a valid burst-payload 1: indicates that the burst-payload may contain errors

Bit	Attr	Reset Value	Description
6:0	RO	0x00	DATATYPE Data type 0000000: null data 0000001: AC-3 data 0000011: Pause data 0000100: MPEG-1 layer 1 data 0000101: MPEG-1 layer 2 or 3 data or MPEG-2 without extension 0000110: MPEG-2 data with extension 0000111: MPEG-2 AAC 0001000: MPEG-2, layer-1 low sampling frequency 0001001: MPEG-2, layer-2 low sampling frequency 0001010: MPEG-2, layer-3 low sampling frequency 0001011: DTS type I 0001100: DTS type II 0001101: DTS type III 0001110: ATRAC 0001111: ATRAC 2/3 0010000: ATRAC-X 0010001: DTS type IV 0010010: WMA professional type I 0110010: WMA professional type II 1010010: WMA professional type III 1110010: WMA professional type IV 0010011: MPEG-2 AAC low sampling frequency 0110011: MPEG-2 AAC low sampling frequency 1010011: MPEG-2 AAC low sampling frequency 1110011: MPEG-2 AAC low sampling frequency 0010100: MPEG-4 AAC 0110100: MPEG-4 AAC 1010100: MPEG-4 AAC 1110100: MPEG-4 AAC 0010101: Enhanced AC-3 0010110: MAT others: reserved

SPDIF_REPETTION_SHD
Address: Operational Base + offset (0x010c)
Shadow Channel Repetition Register

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved

Bit	Attr	Reset Value	Description
15:0	RO	0x0000	REPETTION Repetition This register provides the repetition of the bitstream when channel conveys non-linear PCM. In the design, it defines the length between Pa of the two consecutive data-burst. For the same audio format, the definition is different. Please convert the actual repetition in order to comply with the design.

SPDIF_USRDR_SHDn
Address: Operational Base + offset (0x0190)
Shadow User Data Register n

Bit	Attr	Reset Value	Description
31:16	RO	0x0000	USR_SUB_1 User Data Subframe 1 User Data Bit for Subframe 1
15:0	RO	0x0000	USR_SUB_0 User Data Subframe 0 User Data Bit for Subframe 0

25.5 Interface description

Table 25-1 SPDIF Interface Description

Module Pin	Direction	Pad Name	IOMUX Setting
spdif_8ch_sdo	O	IO_SPDIFtx_GPIO1830gpio4c5	GRF_GPIO4C_IOMUX[11:10]=2'b01
spdif_8ch_sdo	O	IO_MACcol_UART3GPSctsn_SPDIFtxb_GMACgpio3c0	GRF_GPIO3C_IOMUX[1:0]=2'b11

The output of SPDIF module which signals as spdif_8ch_sdo is also connected to the audio interface of HDMI and DP.

Table 25-2 Interface Between SPDIF and HDMI

Module Pin	Direction	Module Pin	Direction
mclk_spdif_8ch	O	ispdifclk	I
spdif_8ch_sdo	O	ispdifdata	I

Table 25-3 Interface Between SPDIF and DP

Module Pin	Direction	Module Pin	Direction
spdif_8ch_sdo	O	source_spdif_din	I

25.6 Application Notes

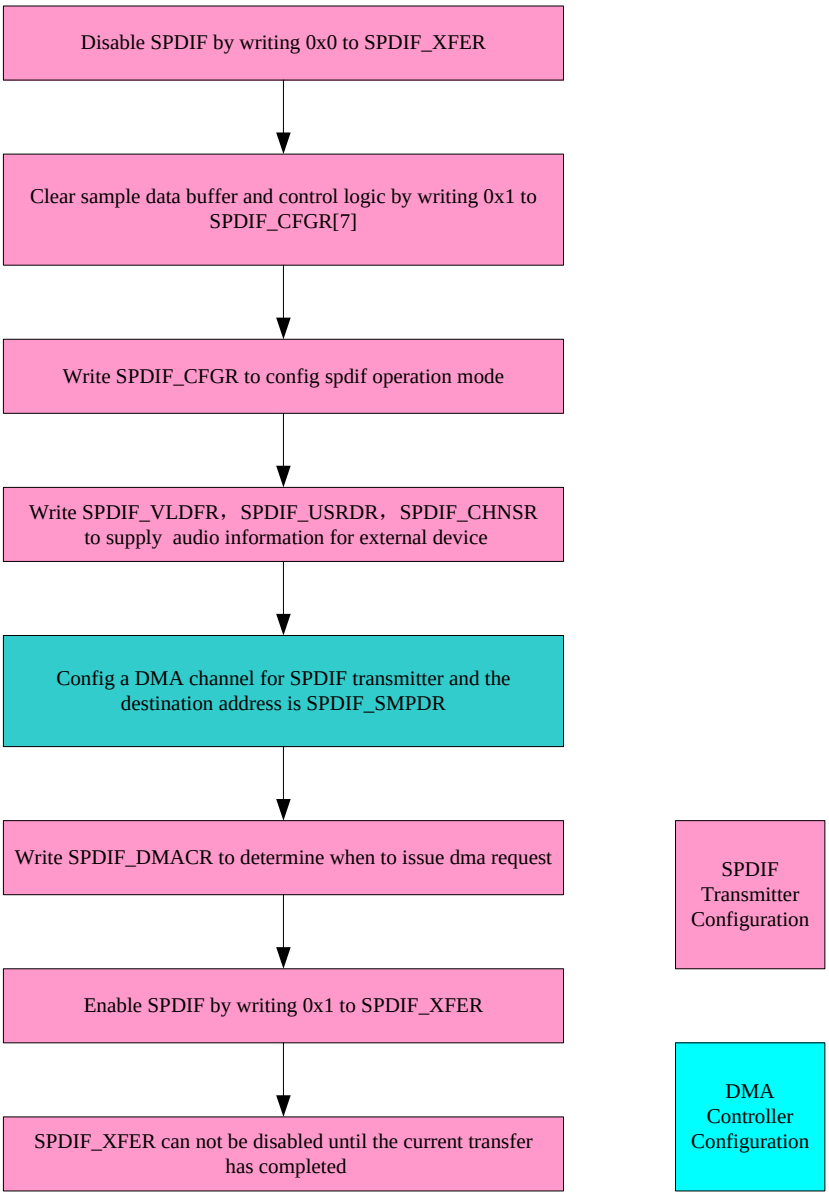


Fig. 25-7 SPDIF transmitter operation flow chart

25.6.1 Channel Status Bit and Validity Flag Bit

Normally the channel status bits and validity flag bits are not necessarily updated frequently. If it is desired to change the channel status bits or validity flag, please write to the corresponding register after a block termination interrupt is asserted. The new value will take effect immediately.

25.6.2 User Data Bit

As the user data bits are updated frequently, the design takes use of the shadow register mechanism to store and convey the user data bit. When the SPDIF interface is disabled, the values of the shadow user data registers keeps the same with the corresponding user data registers. After the SPDIF starts, any change of the user data register will not go to the corresponding shadow user data registers until an user data interrupt is asserted. Therefore before the SPDIF transfer starts, prepare the first 384 user data bits by writing them to the SPDIF_USRDR registers. After the SPDIF transfer starts, writing the second 384 user data bits to the SPDIF_USRDR registers. Then wait for the assertion of user data interrupt. The second 384 user data bits goes to the shadow registers, and then third 384 user bits are written to SPDIF_USRDR.

25.6.3 Burst Info and Repetition

The shadow register mechanism is also applied to the data of burst info and repetition as the user data. The difference is that the update of shadow register will be taken after

assertion of the block termination interrupt.
It is important to note that the repetition defined in the design is a little different from the repetition defined in IEC-61957. The repetition is always defined as the length (measured in IEC-60958 frame) between Pa of two consecutive data-bursts. Therefore the user needs to calculate the new repetition value if the definition of the repetition is different for some audio formats such as AC-3.

Chapter 26 GMAC Ethernet Interface

26.1 Overview

The GMAC Ethernet Controller provides a complete Ethernet interface from processor to a Reduced Media Independent Interface (RMII) and Reduced Gigabit Media Independent Interface (RGMII) compliant Ethernet PHY.
The GMAC includes a DMA controller. The DMA controller efficiently moves packet data from microprocessor's RAM, formats the data for an IEEE 802.3-2002 compliant packet and transmits the data to an Ethernet Physical Interface (PHY). It also efficiently moves packet data from RXFIFO to microprocessor's RAM.

26.1.1 Feature

- Supports 10/100/1000-Mbps data transfer rates with the RGMII interfaces
- Supports 10/100-Mbps data transfer rates with the RMII interfaces
- Supports both full-duplex and half-duplex operation
  - Supports CSMA/CD Protocol for half-duplex operation
  - Supports packet bursting and frame extension in 1000 Mbps half-duplex operation
  - Supports IEEE 802.3x flow control for full-duplex operation
  - Optional forwarding of received pause control frames to the user application in full-duplex operation
  - Back-pressure support for half-duplex operation
  - Automatic transmission of zero-quanta pause frame on de-assertion of flow control input in full-duplex operation
- Preamble and start-of-frame data (SFD) insertion in Transmit, and deletion in Receive paths
- Automatic CRC and pad generation controllable on a per-frame basis
- Options for Automatic Pad/CRC Stripping on receive frames
- Programmable frame length to support Standard Ethernet frames
- Programmable InterFrameGap (40-96 bit times in steps of 8)
- Supports a variety of flexible address filtering modes:
  - 64-bit Hash filter (optional) for multicast and uni-cast (DA) addresses
  - Option to pass all multicast addressed frames
  - Promiscuous mode support to pass all frames without any filtering for network monitoring
  - Passes all incoming packets (as per filter) with a status report
- Separate 32-bit status returned for transmission and reception packets
- Supports IEEE 802.1Q VLAN tag detection for reception frames
- MDIO Master interface for PHY device configuration and management
- Support detection of LAN wake-up frames and AMD Magic Packet frames
- Support checksum off-load for received IPv4 and TCP packets encapsulated by the Ethernet frame
- Support checking IPv4 header checksum and TCP, UDP, or ICMP checksum encapsulated in IPv4 or IPv6 datagrams
- Comprehensive status reporting for normal operation and transfers with errors
- Support per-frame Transmit/Receive complete interrupt control
- Supports 4-KB receive FIFO depths on reception.
- Supports 2-KB FIFO depth on transmission
- Automatic generation of PAUSE frame control or backpressure signal to the GMAC core based on Receive FIFO-fill (threshold configurable) level
- Handles automatic retransmission of Collision frames for transmission
- Discards frames on late collision, excessive collisions, excessive deferral and underrun conditions
- AXI interface to any CPU or memory
- Software can select the type of AXI burst (fixed and variable length burst) in the AXI Master interface
- Supports internal loopback on the RGMII/RMII for debugging

- Debug status register that gives status of FSMs in Transmit and Receive data-paths and FIFO fill-levels.

26.2 Block Diagram

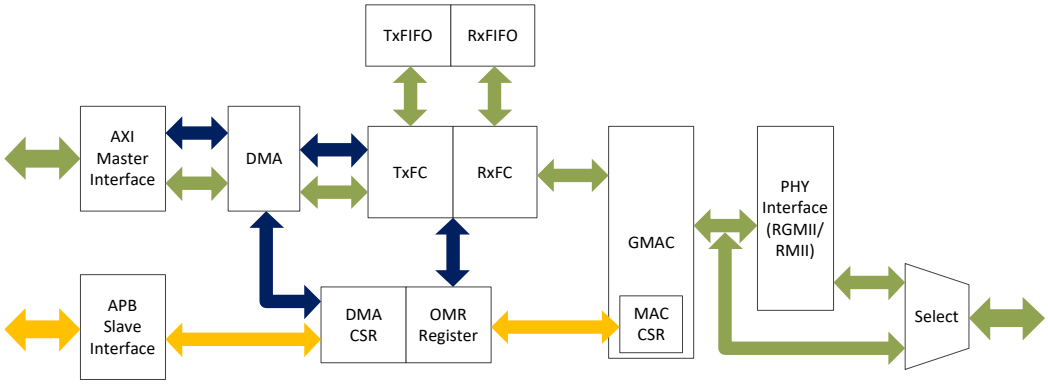


Fig. 26-1 GMACArchitecture

The GMAC is broken up into multiple separate functional units. These blocks are interconnected in the MAC module. The block diagram shows the general flow of data and control signals between these blocks.
The GMAC transfers data to system memory through the AXI master interface. The host CPU uses the APB Slave interface to access the GMAC subsystem's control and status registers (CSRs).
The GMAC supports the PHY interfaces of reduced GMII (RGMII) and reduced MII (RMII).
The Transmit FIFO (Tx FIFO) buffers data read from system memory by the DMA before transmission by the GMAC Core. Similarly, the Receive FIFO (Rx FIFO) stores the Ethernet frames received from the line until they are transferred to system memory by the DMA. These are asynchronous FIFOs, as they also transfer the data between the application clock and the GMAC line clocks.

26.3 Function Description

26.3.1 Frame Structure

Data frames transmitted shall have the frame format shown in Fig. 25-2.

Fig. 26-2 MAC Block Diagram

The preamble <preamble> begins a frame transmission. The bit value of the preamble field consists of 7 octets with the following bit values:
10101010 10101010 10101010 10101010 10101010 10101010 10101010
The SFD (start frame delimiter) <sfd> indicates the start of a frame and follows the preamble.The bit value is 10101011.
The data in a well formed frame shall consist of N octet's data.

26.3.2 RMII Interface timing diagram

The Reduced Media Independent Interface (RMII) specification reduces the pin count between Ethernet PHYs and Switch ASICs (only in 10/100 mode). According to the IEEE 802.3u standard, an MII contains 16 pins for data and control. In devices incorporating multiple MAC or PHY interfaces (such as switches), the number of pins adds significant cost with increase in port count. The RMII specification addresses this problem by reducing the pin count to 7 for each port - a 62.5% decrease in pin count.
The RMII module is instantiated between the GMAC and the PHY. This helps translation of the MAC's MII into the RMII. The RMII block has the following characteristics:

- Supports 10-Mbps and 100-Mbps operating rates. It does not support 1000-Mbps operation.
- Two clock references are sourced externally or CRU, providing independent, 2-bit wide transmit and receive paths.

Transmit Bit Ordering

Each nibble from the MII must be transmitted on the RMII a di-bit at a time with the order of di-bit transmission shown in Fig.1-3. The lower order bits (D1 and D0) are transmitted first followed by higher order bits (D2 and D3).

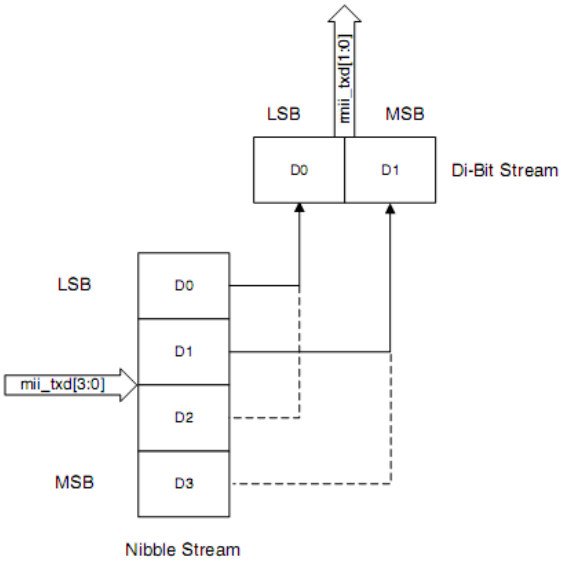


Fig. 26-3 RMII transmission bit ordering

RMII Transmit Timing Diagrams

Fig.1-4 through 1-7 show MII-to-RMII transaction timing.The clk_rmii_i (REF_CLK) frequency is 50MHz in RMII interface.In 10Mb/s mode, as the REF_CLK frequency is 10 times as the data rate, the value on rmii_txd_o[1:0] (TXD[1:0]) shall be valid such that TXD[1:0] may be sampled every 10th cycle,regard-less of the starting cycle within the group and yield the correct frame data.

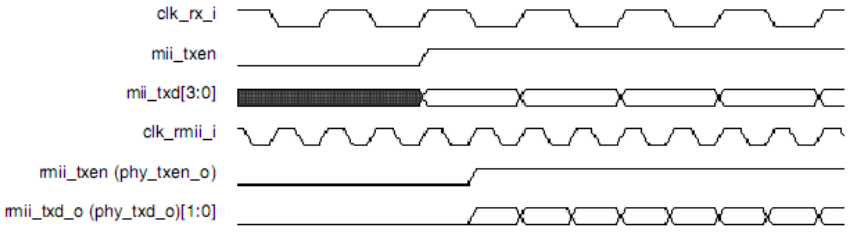


Fig. 26-4 Start of MII and RMII transmission in 100-Mbps mode

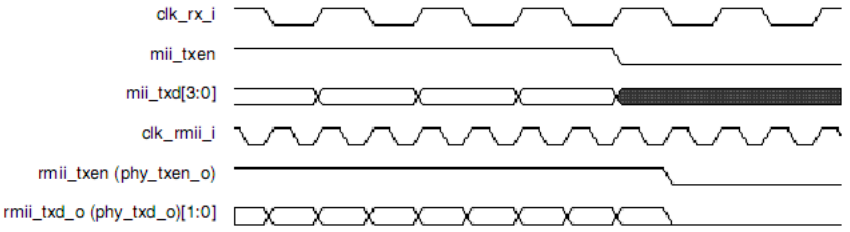


Fig. 26-5 End of MII and RMII Transmission in 100-Mbps Mode

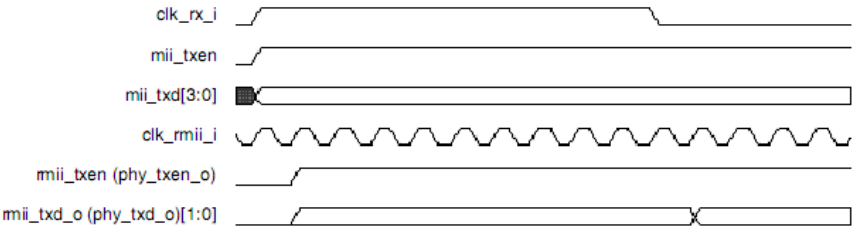


Fig. 26-6 Start of MII and RMII Transmission in 10-Mbps Mode

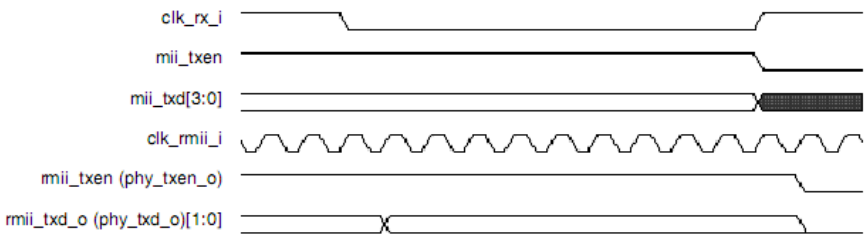


Fig. 26-7 End of MII and RMII Transmission in 10-Mbps Mode

Receive Bit Ordering

Each nibble is transmitted to the MII from the di-bit received from the RMII in the nibble transmission order shown in Fig.1-8. The lower order bits (D0 and D1) are received first, followed by the higher order bits (D2 and D3).

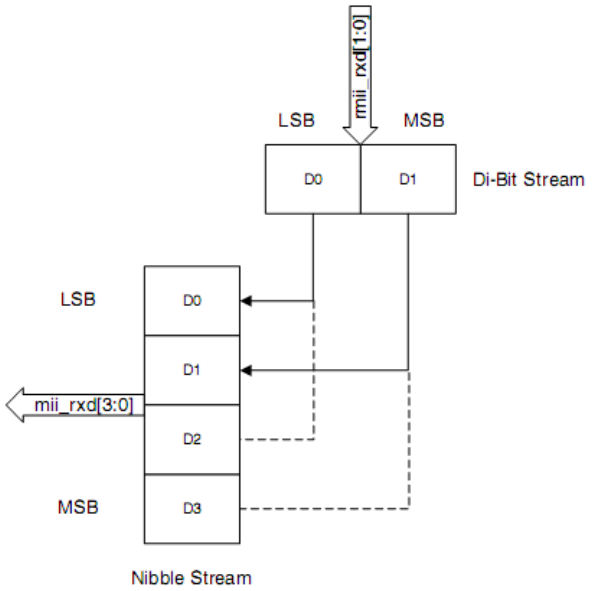


Fig. 26-8 RMII receive bit ordering

26.3.3 RGMII interface

The Reduced Gigabit Media Independent Interface (RGMII) specification reduces the pin count of the interconnection between the GMAC 10/100/1000 controller and the PHY for GMII and MII interfaces. To achieve this, the data path and control signals are reduced and multiplexed together with both the edges of the transmission and receive clocks. For gigabit operation the clocks operate at 125 MHz; for 10/100 operation, the clock rates are 2.5 MHz/25 MHz.

In the GMAC 10/100/1000 controller, the RGMII module is instantiated between the GMAC core's GMII and the PHY to translate the control and data signals between the GMII and RGMII protocols.

The RGMII block has the following characteristics:

- Supports 10-Mbps, 100-Mbps, and 1000-Mbps operation rates.
- For the RGMII block, no extra clock is required because both the edges of the incoming clocks are used.
- The RGMII block extracts the in-band (link speed, duplex mode and link status) status signals from the PHY and provides them to the GMAC core logic for link detection.

26.3.4 Management Interface

The MAC management interface provides a simple, two-wire, serial interface to connect the GMAC and a managed PHY, for the purposes of controlling the PHY and gathering status from the PHY. The management interface consists of a pair of signals that transport the management information across the MII bus: MDIO and MDC.

The GMAC initiates the management write/read operation. The clock gmii_mdc_o(MDC) is a divided clock from the application clock pclk_gmac. The divide factor depends on the clock range setting in the GMII address register. Clock range is set as follows:

Selection	pclk_gmac	MDC Clock
0000	60-100 MHz	pclk_gmac/42

0001	100-150 MHz	pclk_gmac/62
0010	20-35 MHz	pclk_gmac/16
0011	35-60 MHz	pclk_gmac/26
0100	150-250 MHz	pclk_gmac/102
0101	250-300 MHz	pclk_gmac/124
0110, 0111	Reserved	

The MDC is the derivative of the application clock pclk_gmac. The management operation is performed through the gmii_mdi_i, gmii_mdo_o and gmii_mdo_o_e signals. A three-state buffer is implemented in the PAD.
The frame structure on the MDIO line is shown below.

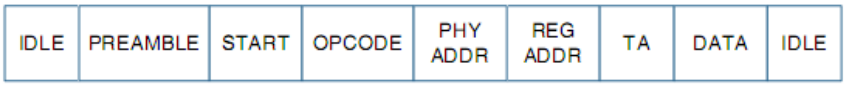


Fig. 26-9 MDIO frame structure

IDLE: The mdio line is three-state; there is no clock on gmii_mdc_o
PREAMBLE: 32 continuous bits of value 1
START: Start-of-frame is 2'b01
OPCODE: 2'b10 for read and 2'b01 for write
PHY ADDR: 5-bit address select for one of 32 PHYs
REG ADDR: Register address in the selected PHY
TA: Turnaround is 2'bZ0 for read and 2'b10 for Write
DATA: Any 16-bit value. In a write operation, the GMAC drives mdio; in a read operation, PHY drives it.

26.3.5 Power Management Block

Power management (PMT) supports the reception of network (remote) wake-up frames and Magic Packet frames. PMT does not perform the clock gate function, but generates interrupts for wake-up frames and Magic Packets received by the GMAC. The PMT block sits on the receiver path of the GMAC and is enabled with remote wake-up frame enable and Magic Packet enable. These enables are in the PMT control and status register and are programmed by the application.
When the power down mode is enabled in the PMT, then all received frames are dropped by the core and they are not forwarded to the application. The core comes out of the power down mode only when either a Magic Packet or a Remote Wake-up frame is received and the corresponding detection is enabled.

Remote Wake-Up Frame Detection

When the GMAC is in sleep mode and the remote wake-up bit is enabled in register GMAC_PMT_CTRL_STA (0x002C), normal operation is resumed after receiving a remote wake-up frame. The application writes all eight wake-up filter registers, by performing a sequential write to address (0028). The application enables remote wake-up by writing a 1 to bit 2 of the register GMAC_PMT_CTRL_STA.
PMT supports four programmable filters that allow support of different receive frame patterns. If the incoming frame passes the address filtering of Filter Command, and if Filter CRC-16 matches the incoming examined pattern, then the wake-up frame is received. Filter_offset (minimum value 12, which refers to the 13th byte of the frame) determines the offset from which the frame is to be examined. Filter Byte Mask determines which bytes of the frame must be examined. The thirty-first bit of Byte Mask must be set to zero.
The remote wake-up CRC block determines the CRC value that is compared with Filter CRC-16. The wake-up frame is checked only for length error, FCS error, dribble bit error, GMII error, collision, and to ensure that it is not a runt frame. Even if the wake-up frame is more than 512 bytes long, if the frame has a valid CRC value, it is considered valid. Wake-up frame detection is updated in the register GMAC_PMT_CTRL_STA for every remote Wake-up frame received. A PMT interrupt to the application triggers a read to the GMAC_PMT_CTRL_STA register to determine reception of a wake-up frame.

Magic Packet Detection

The Magic Packet frame is based on a method that uses Advanced Micro Device's Magic Packet technology to power up the sleeping device on the network. The GMAC receives a specific packet of information, called a Magic Packet, addressed to the node on the network.

Only Magic Packets that are addressed to the device or a broadcast address will be checked to determine whether they meet the wake-up requirements. Magic Packets that pass the address filtering (unicast or broadcast) will be checked to determine whether they meet the remote Wake-on-LAN data format of 6 bytes of all ones followed by a GMAC Address appearing 16 times.
The application enables Magic Packet wake-up by writing a 1 to Bit 1 of the register GMAC_PMT_CTRL_STA. The PMT block constantly monitors each frame addressed to the node for a specific Magic Packet pattern. Each frame received is checked for a 48'hFF_FF_FF_FF_FF_FF pattern following the destination and source address field. The PMT block then checks the frame for 16 repetitions of the GMAC address without any breaks or interruptions. In case of a break in the 16 repetitions of the address, the 48'hFF_FF_FF_FF_FF_FF pattern is scanned for again in the incoming frame. The 16 repetitions can be anywhere in the frame, but must be preceded by the synchronization stream (48'hFF_FF_FF_FF_FF_FF). The device will also accept a multicast frame, as long as the 16 duplications of the GMAC address are detected.
If the MAC address of a node is 48'h00_11_22_33_44_55, then the GMAC scans for the data sequence:
Destination Address Source Address ……. FF FFFFFFFFF
00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55
00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55
00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55
00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55
…CRC
Magic Packet detection is updated in the PMT Control and Status register for Magic Packet received. A PMT interrupt to the Application triggers a read to the PMT CSR to determine whether a Magic Packet frame has been received.

26.3.6 MAC Management Counters

The counters in the MAC Management Counters (MMC) module can be viewed as an extension of the register address space of the CSR module. The MMC module maintains a set of registers for gathering statistics on the received and transmitted frames. These include a control register for controlling the behavior of the registers, two 32-bit registers containing interrupts generated (receive and transmit), and two 32-bit registers containing masks for the Interrupt register (receive and transmit). These registers are accessible from the Application through the MAC Control Interface (MCI). Non-32-bit accesses are allowed as long as the address is word-aligned.
The organization of these registers is shown in Register Description. The MMCs are accessed using transactions, in the same way the CSR address space is accessed. The Register Description in this chapter describe the various counters and list the address for each of the statistics counters. This address will be used for Read/Write accesses to the desired transmit/receive counter.
The MMC module gathers statistics on encapsulated IPv4, IPv6, TCP, UDP, or ICMP payloads in received Ethernet frames.

26.4 Register Description

26.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
GMAC_MAC_CONF	0x0000	W	0x00000000	MAC Configuration Register
GMAC_MAC_FRM_FILT	0x0004	W	0x00000000	MAC Frame Filter
GMAC_HASH_TAB_HI	0x0008	W	0x00000000	Hash Table High Register
GMAC_HASH_TAB_LO	0x000c	W	0x00000000	Hash Table Low Register
GMAC_GMII_ADDR	0x0010	W	0x00000000	GMII Address Register
GMAC_GMII_DATA	0x0014	W	0x00000000	GMII Data Register
GMAC_FLOW_CTRL	0x0018	W	0x00000000	Flow Control Register

Name	Offset	Size	Reset Value	Description
GMAC_VLAN_TAG	0x001c	W	0x00000000	VLAN Tag Register
GMAC_DEBUG	0x0024	W	0x00000000	Debug register
GMAC_PMT_CTRL_STA	0x002c	W	0x00000000	PMT Control and Status Register
GMAC_INT_STATUS	0x0038	W	0x00000000	Interrupt Status Register
GMAC_INT_MASK	0x003c	W	0x00000000	Interrupt Mask Register
GMAC_MAC_ADDR0_HI	0x0040	W	0x0000ffff	MAC Address0 High Register
GMAC_MAC_ADDR0_LO	0x0044	W	0xffffffff	MAC Address0 Low Register
GMAC_AN_CTRL	0x00c0	W	0x00000000	AN Control Register
GMAC_AN_STATUS	0x00c4	W	0x00000008	AN Status Register
GMAC_AN_ADV	0x00c8	W	0x000001e0	Auto Negotiation Advertisement Register
GMAC_AN_LINK_PART_AB	0x00cc	W	0x00000000	Auto Negotiation Link Partner Ability Register
GMAC_AN_EXP	0x00d0	W	0x00000000	Auto Negotiation Expansion Register
GMAC_INTF_MODE_STA	0x00d8	W	0x00000000	RGMII Status Register
GMAC_MMC_CTRL	0x0100	W	0x00000000	MMC Control Register
GMAC_MMC_RX_INTR	0x0104	W	0x00000000	MMC Receive Interrupt Register
GMAC_MMC_TX_INTR	0x0108	W	0x00000000	MMC Transmit Interrupt Register
GMAC_MMC_RX_INT_MSK	0x010c	W	0x00000000	MMC Receive Interrupt Mask Register
GMAC_MMC_TX_INT_MSK	0x0110	W	0x00000000	MMC Transmit Interrupt Mask Register
GMAC_MMC_TXOCTETCNT_GB	0x0114	W	0x00000000	MMC TX OCTET Good and Bad Counter
GMAC_MMC_TXFRMCNT_GB	0x0118	W	0x00000000	MMC TX Frame Good and Bad Counter
GMAC_MMC_TXUNDFLWERR	0x0148	W	0x00000000	MMC TX Underflow Error
GMAC_MMC_TXCARERR	0x0160	W	0x00000000	MMC TX Carrier Error
GMAC_MMC_TXOCTETCNT_G	0x0164	W	0x00000000	MMC TX OCTET Good Counter
GMAC_MMC_TXFRMCNT_G	0x0168	W	0x00000000	MMC TX Frame Good Counter
GMAC_MMC_RXFRMCNT_GB	0x0180	W	0x00000000	MMC RX Frame Good and Bad Counter
GMAC_MMC_RXOCTETCNT_GB	0x0184	W	0x00000000	MMC RX OCTET Good and Bad Counter
GMAC_MMC_RXOCTETCNT_G	0x0188	W	0x00000000	MMC RX OCTET Good Counter
GMAC_MMC_RXMCFRMCNT_G	0x0190	W	0x00000000	MMC RX Multicast Frame Good Counter
GMAC_MMC_RXCRCERR	0x0194	W	0x00000000	MMC RX Carrier

Name	Offset	Size	Reset Value	Description
GMAC_MMC_RXLENERR	0x01c8	W	0x00000000	MMC RX Length Error
GMAC_MMC_RXFIFOOVRFLW	0x01d4	W	0x00000000	MMC RX FIFO Overflow
GMAC_MMC_IPC_INT_MSK	0x0200	W	0x00000000	MMC Receive Checksum Offload Interrupt Mask Register
GMAC_MMC_IPC_INTR	0x0208	W	0x00000000	MMC Receive Checksum Offload Interrupt Register
GMAC_MMC_RXIPV4GFRM	0x0210	W	0x00000000	MMC RX IPV4 Good Frame
GMAC_MMC_RXIPV4HDERRFRM	0x0214	W	0x00000000	MMC RX IPV4 Head Error Frame
GMAC_MMC_RXIPV6GFRM	0x0224	W	0x00000000	MMC RX IPV6 Good Frame
GMAC_MMC_RXIPV6HDERRFRM	0x0228	W	0x00000000	MMC RX IPV6 Head Error Frame
GMAC_MMC_RXUDPERRFRM	0x0234	W	0x00000000	MMC RX UDP Error Frame
GMAC_MMC_RXTCPERRFRM	0x023c	W	0x00000000	MMC RX TCP Error Frame
GMAC_MMC_RXICMPERRFRM	0x0244	W	0x00000000	MMC RX ICMP Error Frame
GMAC_MMC_RXIPV4HDERROCT	0x0254	W	0x00000000	MMC RX OCTET IPV4 Head Error
GMAC_MMC_RXIPV6HDERROCT	0x0268	W	0x00000000	MMC RX OCTET IPV6 Head Error
GMAC_MMC_RXUDPERROCT	0x0274	W	0x00000000	MMC RX OCTET UDP Error
GMAC_MMC_RXTCPERROCT	0x027c	W	0x00000000	MMC RX OCTET TCP Error
GMAC_MMC_RXICMPERROCT	0x0284	W	0x00000000	MMC RX OCTET ICMP Error
GMAC_BUS_MODE	0x1000	W	0x00020101	Bus Mode Register
GMAC_TX_POLL_DEMAND	0x1004	W	0x00000000	Transmit Poll Demand Register
GMAC_RX_POLL_DEMAND	0x1008	W	0x00000000	Receive Poll Demand Register
GMAC_RX_DESC_LIST_ADDR	0x100c	W	0x00000000	Receive Descriptor List Address Register
GMAC_TX_DESC_LIST_ADDR	0x1010	W	0x00000000	Transmit Descriptor List Address Register
GMAC_STATUS	0x1014	W	0x00000000	Status Register
GMAC_OP_MODE	0x1018	W	0x00000000	Operation Mode Register
GMAC_INT_ENA	0x101c	W	0x00000000	Interrupt Enable Register
GMAC_OVERFLOW_CNT	0x1020	W	0x00000000	Missed Frame and Buffer Overflow Counter Register
GMAC_REC_INT_WDT_TIMER	0x1024	W	0x00000000	Receive Interrupt Watchdog Timer Register
GMAC_AXI_BUS_MODE	0x1028	W	0x00110001	AXI Bus Mode Register

Name	Offset	Size	Reset Value	Description
GMAC_AXI_STATUS	0x102c	W	0x00000000	AXI Status Register
GMAC_CUR_HOST_TX_DESC	0x1048	W	0x00000000	Current Host Transmit Descriptor Register
GMAC_CUR_HOST_RX_DESC	0x104c	W	0x00000000	Current Host Receive Descriptor Register
GMAC_CUR_HOST_TX_BUF_ADDR	0x1050	W	0x00000000	Current Host Transmit Buffer Address Register
GMAC_CUR_HOST_RX_BUF_ADDR	0x1054	W	0x00000000	Current Host Receive Buffer Address Register

*Notes:Size:**B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access*

26.4.2 Detail Register Description

GMAC_MAC_CONF

Address: Operational Base + offset (0x0000)

MAC Configuration Register

Bit	Attr	Reset Value	Description
31:25	RO	0x0	reserved
24	RW	0x0	TC Transmit Configuration in RGMII When set, this bit enables the transmission of duplex mode, link speed, and link up/down information to the PHY in the RGMII ports. When this bit is reset, no such information is driven to the PHY.
23	RW	0x0	WD Watchdog Disable When this bit is set, the GMAC disables the watchdog timer on the receiver, and can receive frames of up to 16,384 bytes. When this bit is reset, the GMAC allows no more than 2,048 bytes (10,240 if JE is set high) of the frame being received and cuts off any bytes received after that.
22	RW	0x0	JD Jabber Disable When this bit is set, the GMAC disables the jabber timer on the transmitter, and can transfer frames of up to 16,384 bytes. When this bit is reset, the GMAC cuts off the transmitter if the application sends out more than 2,048 bytes of data (10,240 if JE is set high) during transmission.
21	RW	0x0	BE Frame Burst Enable When this bit is set, the GMAC allows frame bursting during transmission in GMII Half-Duplex mode.
20	RO	0x0	reserved

Bit	Attr	Reset Value	Description
19:17	RW	0x0	IFG Inter-Frame Gap These bits control the minimum IFG between frames during transmission. 3'b000: 96 bit times 3'b001: 88 bit times 3'b010: 80 bit times ... 3'b111: 40 bit times
16	RW	0x0	DCRS Disable Carrier Sense During Transmission When set high, this bit makes the MAC transmitter ignore the (G)MII CRS signal during frame transmission in Half-Duplex mode. This request results in no errors generated due to Loss of Carrier or No Carrier during such transmission. When this bit is low, the MAC transmitter generates such errors due to Carrier Sense and will even abort the transmissions.
15	RW	0x0	PS Port Select Selects between GMII and MII: 1'b0: GMII (1000 Mbps) 1'b1: MII (10/100 Mbps)
14	RW	0x0	FES Speed Indicates the speed in Fast Ethernet (MII) mode: 1'b0: 10 Mbps 1'b1: 100 Mbps
13	RW	0x0	DO Disable Receive Own When this bit is set, the GMAC disables the reception of frames when the gmii_txen_o is asserted in Half-Duplex mode. When this bit is reset, the GMAC receives all packets that are given by the PHY while transmitting.
12	RW	0x0	LM Loopback Mode When this bit is set, the GMAC operates in loopback mode at GMII/MII. The (G)MII Receive clock input (clk_rx_i) is required for the loopback to work properly, as the Transmit clock is not looped-back internally.
11	RW	0x0	DM Duplex Mode When this bit is set, the GMAC operates in a Full-Duplex mode where it can transmit and receive simultaneously. This bit is RO with default value of 1'b1 in Full-Duplex-only configuration.

Bit	Attr	Reset Value	Description
10	RW	0x0	IPC Checksum Offload When this bit is set, the GMAC calculates the 16-bit one's complement of the one's complement sum of all received Ethernet frame payloads. It also checks whether the IPv4 Header checksum (assumed to be bytes 25-26 or 29-30 (VLAN-tagged) of the received Ethernet frame) is correct for the received frame and gives the status in the receive status word. The GMAC core also appends the 16-bit checksum calculated for the IP header datagram payload (bytes after the IPv4 header) and appends it to the Ethernet frame transferred to the application (when Type 2 COE is deselected). When this bit is reset, this function is disabled. When Type 2 COE is selected, this bit, when set, enables IPv4 checksum checking for received frame payloads TCP/UDP/ICMP headers. When this bit is reset, the COE function in the receiver is disabled and the corresponding PCE and IP HCE status bits are always cleared.
9	RW	0x0	DR Disable Retry When this bit is set, the GMAC will attempt only 1 transmission. When a collision occurs on the GMII/MII, the GMAC will ignore the current frame transmission and report a Frame Abort with excessive collision error in the transmit frame status. When this bit is reset, the GMAC will attempt retries based on the settings of BL.
8	RW	0x0	LUD Link Up/Down Indicates whether the link is up or down during the transmission of configuration in RGMII interface: 1'b0: Link Down 1'b1: Link Up
7	RW	0x0	ACS Automatic Pad/CRC Stripping When this bit is set, the GMAC strips the Pad/FCS field on incoming frames only if the length's field value is less than or equal to 1,500 bytes. All received frames with length field greater than or equal to 1,501 bytes are passed to the application without stripping the Pad/FCS field. When this bit is reset, the GMAC will pass all incoming frames to the Host unmodified.

Bit	Attr	Reset Value	Description
6:5	RW	0x0	BL Back-Off Limit The Back-Off limit determines the random integer number (r) of slot time delays (4,096 bit times for 1000 Mbps and 512 bit times for 10/100 Mbps) the GMAC waits before rescheduling a transmission attempt during retries after a collision. This bit is applicable only to Half-Duplex mode and is reserved (RO) in Full-Duplex-only configuration. 2'b00: k = min (n, 10) 2'b01: k = min (n, 8) 2'b10: k = min (n, 4) 2'b11: k = min (n, 1), Where n = retransmission attempt. The random integer r takes the value in the range 0 = r < 2^k
4	RW	0x0	DC Deferral Check When this bit is set, the deferral check function is enabled in the GMAC. The GMAC will issue a Frame Abort status, along with the excessive deferral error bit set in the transmit frame status when the transmission state machine is deferred for more than 24,288 bit times in 10/100-Mbps mode. If the Core is configured for 1000 Mbps operation, the threshold for deferral is 155,680 bits times. Deferral begins when the transmitter is ready to transmit, but is prevented because of an active CRS (carrier sense) signal on the GMII/MII. Defer time is not cumulative. If the transmitter defers for 10,000 bit times, then transmits, collides, backs off, and then has to defer again after completion of back-off, the deferral timer resets to 0 and restarts. When this bit is reset, the deferral check function is disabled and the GMAC defers until the CRS signal goes inactive.
3	RW	0x0	TE Transmitter Enable When this bit is set, the transmission state machine of the GMAC is enabled for transmission on the GMII/MII. When this bit is reset, the GMAC transmit state machine is disabled after the completion of the transmission of the current frame, and will not transmit any further frames.
2	RW	0x0	RE Receiver Enable When this bit is set, the receiver state machine of the GMAC is enabled for receiving frames from the GMII/MII. When this bit is reset, the GMAC receive state machine is disabled after the completion of the reception of the current frame, and will not receive any further frames from the GMII/MII.
1:0	RO	0x0	reserved

GMAC_MAC_FRM_FILT
Address: Operational Base + offset (0x0004)
MAC Frame Filter

Bit	Attr	Reset Value	Description
31	RW	0x0	RA Receive All When this bit is set, the GMAC Receiver module passes to the Application all frames received irrespective of whether they pass the address filter. The result of the SA/DA filtering is updated (pass or fail) in the corresponding bits in the Receive Status Word. When this bit is reset, the Receiver module passes to the Application only those frames that pass the SA/DA address filter.
30:11	RO	0x0	reserved
10	RW	0x0	HPF Hash or Perfect Filter When set, this bit configures the address filter to pass a frame if it matches either the perfect filtering or the hash filtering as set by HMC or HUC bits. When low and if the HUC/HMC bit is set, the frame is passed only if it matches the Hash filter.
9	RW	0x0	SAF Source Address Filter Enable The GMAC core compares the SA field of the received frames with the values programmed in the enabled SA registers. If the comparison matches, then the SAMatch bit of RxStatus Word is set high. When this bit is set high and the SA filter fails, the GMAC drops the frame. When this bit is reset, then the GMAC Core forwards the received frame to the application and with the updated SA Match bit of the RxStatus depending on the SA address comparison.
8	RW	0x0	SAIF SA Inverse Filtering When this bit is set, the Address Check block operates in inverse filtering mode for the SA address comparison. The frames whose SA matches the SA registers will be marked as failing the SA Address filter. When this bit is reset, frames whose SA does not match the SA registers will be marked as failing the SA Address filter.

Bit	Attr	Reset Value	Description
7:6	RW	0x0	PCF Pass Control Frames These bits control the forwarding of all control frames (including unicast and multicast PAUSE frames). Note that the processing of PAUSE control frames depends only on RFE of Register GMAC_FLOW_CTRL[2]. 2'b00: GMAC filters all control frames from reaching the application. 2'b01: GMAC forwards all control frames except PAUSE control frames to application even if they fail the Address filter. 2'b10: GMAC forwards all control frames to application even if they fail the Address Filter. 2'b11: GMAC forwards control frames that pass the Address Filter.
5	RW	0x0	DBF Disable Broadcast Frames When this bit is set, the AFM module filters all incoming broadcast frames. When this bit is reset, the AFM module passes all received broadcast frames.
4	RW	0x0	PM Pass All Multicast When set, this bit indicates that all received frames with a multicast destination address (first bit in the destination address field is '1') are passed. When reset, filtering of multicast frame depends on HMC bit.
3	RW	0x0	DAIF DA Inverse Filtering When this bit is set, the Address Check block operates in inverse filtering mode for the DA address comparison for both unicast and multicast frames. When reset, normal filtering of frames is performed.
2	RW	0x0	HMC Hash Multicast When set, MAC performs destination address filtering of received multicast frames according to the hash table. When reset, the MAC performs a perfect destination address filtering for multicast frames, that is, it compares the DA field with the values programmed in DA registers.
1	RW	0x0	HUC Hash Unicast When set, MAC performs destination address filtering of unicast frames according to the hash table. When reset, the MAC performs a perfect destination address filtering for unicast frames, that is, it compares the DA field with the values programmed in DA registers.

Bit	Attr	Reset Value	Description
0	RW	0x0	PR Promiscuous Mode When this bit is set, the Address Filter module passes all incoming frames regardless of its destination or source address. The SA/DA Filter Fails status bits of the Receive Status Word will always be cleared when PR is set.

GMAC_HASH_TAB_HI
Address: Operational Base + offset (0x0008)
Hash Table High Register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	HTH Hash Table High This field contains the upper 32 bits of Hash table

GMAC_HASH_TAB_LO
Address: Operational Base + offset (0x000c)
Hash Table Low Register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	HTL Hash Table Low This field contains the lower 32 bits of Hash table

GMAC_GMII_ADDR
Address: Operational Base + offset (0x0010)
GMII Address Register

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:11	RW	0x00	PA Physical Layer Address This field tells which of the 32 possible PHY devices are being accessed
10:6	RW	0x00	GR GMII Register These bits select the desired GMII register in the selected PHY device

Bit	Attr	Reset Value	Description
5:2	RW	0x0	CR APB Clock Range The APB Clock Range selection determines the frequency of the MDC clock as per the pclk_gmac frequency used in your design. The suggested range of pclk_gmac frequency applicable for each value below (when Bit[5] = 0) ensures that the MDC clock is approximately between the frequency range 1.0 MHz - 2.5 MHz. **Selection pclk_gmacMDC Clock** 0000 60-100 MHz pclk_gmac/42 0001 100-150 MHz pclk_gmac/62 0010 20-35 MHz pclk_gmac/16 0011 35-60 MHz pclk_gmac/26 0100 150-250 MHz pclk_gmac/102 0101 250-300 MHz pclk_gmac/124 0110, 0111 Reserved When bit 5 is set, you can achieve MDC clock of frequency higher than the IEEE802.3 specified frequency limit of 2.5 MHz and program a clock divider of lower value. For example, when pclk_gmac is of frequency 100 MHz and you program these bits as "1010", then the resultant MDC clock will be of 12.5 MHz which is outside the limit of IEEE 802.3 specified range. Please program the values given below only if the interfacing chips supports faster MDC clocks. **Selection MDC Clock** 1000 pclk_gmac/4 1001 pclk_gmac/6 1010 pclk_gmac/8 1011 pclk_gmac/10 1100 pclk_gmac/12 1101 pclk_gmac/14 1110 pclk_gmac/16 1111 pclk_gmac/18
1	RW	0x0	GW GMII Write When set, this bit tells the PHY that this will be a Write operation using register GMAC_GMII_DATA. If this bit is not set, this will be a Read operation, placing the data in register GMAC_GMII_DATA.

Bit	Attr	Reset Value	Description
0	W1C	0x0	GB GMII Busy This bit should read a logic 0 before writing to Register GMII_ADDR and Register GMII_DATA. This bit must also be set to 0 during a Write to Register GMII_ADDR. During a PHY register access, this bit will be set to 1'b1 by the Application to indicate that a Read or Write access is in progress. Register GMII_DATA (GMII Data) should be kept valid until this bit is cleared by the GMAC during a PHY Write operation. The Register GMII_DATA is invalid until this bit is cleared by the GMAC during a PHY Read operation. The Register GMII_ADDR (GMII Address) should not be written to until this bit is cleared.

GMAC_GMII_DATA
Address: Operational Base + offset (0x0014)
GMII Data Register

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	GD GMII Data This contains the 16-bit data value read from the PHY after a Management Read operation or the 16-bit data value to be written to the PHY before a Management Write operation.

GMAC_FLOW_CTRL
Address: Operational Base + offset (0x0018)
Flow Control Register

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	PT Pause Time This field holds the value to be used in the Pause Time field in the transmit control frame. If the Pause Time bits is configured to be double-synchronized to the (G)MII clock domain, then consecutive writes to this register should be performed only after at least 4 clock cycles in the destination clock domain.
15:8	RO	0x0	reserved
7	RW	0x0	DZPQ Disable Zero-Quanta Pause When set, this bit disables the automatic generation of Zero-Quanta Pause Control frames on the de-assertion of the flow-control signal from the FIFO layer (MTL or external sideband flow control signal sbd_flowctrl_i/mti_flowctrl_i). When this bit is reset, normal operation with automatic Zero-Quanta Pause Control frame generation is enabled.
6	RO	0x0	reserved

Bit	Attr	Reset Value	Description
5:4	RW	0x0	PLT Pause Low Threshold This field configures the threshold of the PAUSE timer at which the input flow control signal mti_flowctrl_i (or sbd_flowctrl_i) is checked for automatic retransmission of PAUSE Frame. The threshold values should be always less than the Pause Time configured in Bits[31:16]. For example, if PT = 100H (256 slot-times), and PLT = 01, then a second PAUSE frame is automatically transmitted if the mti_flowctrl_i signal is asserted at 228 (256-28) slot-times after the first PAUSE frame is transmitted. **Selection Threshold** 00 Pause time minus 4 slot times 01 Pause time minus 28 slot times 10 Pause time minus 144 slot times 11 Pause time minus 256 slot times Slot time is defined as time taken to transmit 512 bits (64 bytes) on the GMII/MII interface.
3	RW	0x0	UP Unicast Pause Frame Detect When this bit is set, the GMAC will detect the Pause frames with the station's unicast address specified in MAC Address0 High Register and MAC Address0 Low Register, in addition to the detecting Pause frames with the unique multicast address. When this bit is reset, the GMAC will detect only a Pause frame with the unique multicast address specified in the 802.3x standard.
2	RW	0x0	RFE Receive Flow Control Enable When this bit is set, the GMAC will decode the received Pause frame and disable its transmitter for a specified (Pause Time) time. When this bit is reset, the decode function of the Pause frame is disabled.
1	RW	0x0	TFE Transmit Flow Control Enable In Full-Duplex mode, when this bit is set, the GMAC enables the flow control operation to transmit Pause frames. When this bit is reset, the flow control operation in the GMAC is disabled, and the GMAC will not transmit any Pause frames. In Half-Duplex mode, when this bit is set, the GMAC enables the back-pressure operation. When this bit is reset, the backpressure feature is disabled.

Bit	Attr	Reset Value	Description
0	RW	0x0	FCB_BPA Flow Control Busy/Backpressure Activate This bit initiates a Pause Control frame in Full-Duplex mode and activates the backpressure function in Half-Duplex mode if TFE bit is set. In Full-Duplex mode, this bit should be read as 1'b0 before writing to the register GMAC_FLOW_CTRL. To initiate a pause control frame, the application must set this bit to 1'b1. During a transfer of the control frame, this bit will continue to be set to signify that a frame transmission is in progress. After the completion of Pause control frame transmission, the GMAC will reset this bit to 1'b0. The register GMAC_FLOW_CTRL should not be written to until this bit is cleared. In Half-Duplex mode, when this bit is set (and TFE is set), then backpressure is asserted by the GMAC Core. During backpressure, when the GMAC receives a new frame, the transmitter starts sending a JAM pattern resulting in a collision. This control register bit is logically OR'ed with the mti_flowctrl_i input signal for the backpressure function.

GMAC_VLAN_TAG
Address: Operational Base + offset (0x001c)
VLAN Tag Register

Bit	Attr	Reset Value	Description
31:17	RO	0x0	reserved
16	RW	0x0	ETV Enable 12-Bit VLAN Tag Comparison When this bit is set, a 12-bit VLAN identifier, rather than the complete 16-bit VLAN tag, is used for comparison and filtering. Bits[11:0] of the VLAN tag are compared with the corresponding field in the received VLAN-tagged frame. When this bit is reset, all 16 bits of the received VLAN frame's fifteenth and sixteenth bytes are used for comparison.
15:0	RW	0x0000	VL VLAN Tag Identifier for Receive Frames This contains the 802.1Q VLAN tag to identify VLAN frames, and is compared to the fifteenth and sixteenth bytes of the frames being received for VLAN frames. Bits[15:13] are the User Priority, Bit[12] is the Canonical Format Indicator (CFI) and bits[11:0] are the VLAN tag's VLAN Identifier (VID) field. When the ETV bit is set, only the VID (Bits[11:0]) is used for comparison. If VL (VL[11:0] if ETV is set) is all zeros, the GMAC does not check the fifteenth and sixteenth bytes for VLAN tag comparison, and declares all frames with a Type field value of 0x8100 to be VLAN frames.

GMAC_DEBUG
Address: Operational Base + offset (0x0024)
Debug register

Bit	Attr	Reset Value	Description
31:26	RO	0x0	reserved
25	RW	0x0	TFIFO3 When high, it indicates that the MTL TxStatus FIFO is full and hence the MTL will not be accepting any more frames for transmission.
24	RW	0x0	TFIFO2 When high, it indicates that the MTL TxFIFO is not empty and has some data left for transmission.
23	RO	0x0	reserved
22	RW	0x0	TFIFO1 When high, it indicates that the MTL TxFIFO Write Controller is active and transferring data to the TxFIFO.
21:20	RW	0x0	TFIFOSTA This indicates the state of the TxFIFO read Controller: 2'b00: IDLE state 2'b01: READ state (transferring data to MAC transmitter) 2'b10: Waiting for TxStatus from MAC transmitter 2'b11: Writing the received TxStatus or flushing the TxFIFO
19	RW	0x0	PAUSE When high, it indicates that the MAC transmitter is in PAUSE condition (in full-duplex only) and hence will not schedule any frame for transmission
18:17	RW	0x0	TSAT This indicates the state of the MAC Transmit Frame Controller module: 2'b00: IDLE 2'b01: Waiting for Status of previous frame or IFG/backoff period to be over 2'b10: Generating and transmitting a PAUSE control frame (in full duplex mode) 2'b11: Transferring input frame for transmission
16	RW	0x0	TACT When high, it indicates that the MAC GMII/MII transmit protocol engine is actively transmitting data and not in IDLE state.
15:10	RO	0x0	reserved
9:8	RW	0x0	RFIFO This gives the status of the RxFIFO Fill-level: 2'b00: RxFIFO Empty 2'b01: RxFIFO fill-level below flow-control de-activate threshold 2'b10: RxFIFO fill-level above flow-control activate threshold 2'b11: RxFIFO Full
7	RO	0x0	reserved

Bit	Attr	Reset Value	Description
6:5	RW	0x0	RFIFORD It gives the state of the RxFIFO read Controller: 2'b00: IDLE state 2'b01: Reading frame data 2'b10: Reading frame status (or time-stamp) 2'b11: Flushing the frame data and Status
4	RW	0x0	RFIFOWR When high, it indicates that the MTL RxFIFO Write Controller is active and transferring a received frame to the FIFO.
3	RO	0x0	reserved
2:1	RW	0x0	ACT When high, it indicates the active state of the small FIFO Read and Write controllers respectively of the MAC receive Frame Controller module
0	RW	0x0	RDB When high, it indicates that the MAC GMII/MII receive protocol engine is actively receiving data and not in IDLE state.

GMAC_PMT_CTRL_STA
Address: Operational Base + offset (0x002c)
PMT Control and Status Register

Bit	Attr	Reset Value	Description
31	W1C	0x0	WFFRPR Wake-Up Frame Filter Register Pointer Reset When set, resets the Remote Wake-up Frame Filter register pointer to 3'b000. It is automatically cleared after 1 clock cycle.
30:10	RO	0x0	reserved
9	RW	0x0	GU Global Unicast When set, enables any unicast packet filtered by the GMAC (DAF) address recognition to be a wake-up frame.
8:7	RO	0x0	reserved
6	RC	0x0	WFR Wake-Up Frame Received When set, this bit indicates the power management event was generated due to reception of a wake-up frame. This bit is cleared by a read into this register.
5	RC	0x0	MPR Magic Packet Received When set, this bit indicates the power management event was generated by the reception of a Magic Packet. This bit is cleared by a read into this register.
4:3	RO	0x0	reserved

Bit	Attr	Reset Value	Description
2	RW	0x0	WFE Wake-Up Frame Enable When set, enables generation of a power management event due to wake-up frame reception.
1	RW	0x0	MPE Magic Packet Enable When set, enables generation of a power management event due to Magic Packet reception.
0	R/WSC	0x0	PD Power Down When set, all received frames will be dropped. This bit is cleared automatically when a magic packet or Wake-Up frame is received, and Power-Down mode is disabled. Frames received after this bit is cleared are forwarded to the application. This bit must only be set when either the Magic Packet Enable or Wake-Up Frame Enable bit is set high.

GMAC_INT_STATUS
Address: Operational Base + offset (0x0038)
Interrupt Status Register

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7	RO	0x0	MRCOIS MMC Receive Checksum Offload Interrupt Status This bit is set high whenever an interrupt is generated in the MMC Receive Checksum Offload Interrupt Register. This bit is cleared when all the bits in this interrupt register are cleared.
6	RO	0x0	MTIS MMC Transmit Interrupt Status This bit is set high whenever an interrupt is generated in the MMC Transmit Interrupt Register. This bit is cleared when all the bits in this interrupt register are cleared. This bit is only valid when the optional MMC module is selected during configuration.
5	RO	0x0	MRIS MMC Receive Interrupt Status This bit is set high whenever an interrupt is generated in the MMC Receive Interrupt Register. This bit is cleared when all the bits in this interrupt register are cleared. This bit is only valid when the optional MMC module is selected during configuration.
4	RO	0x0	MIS MMC Interrupt Status This bit is set high whenever any of bits 7:5 is set high and cleared only when all of these bits are low. This bit is valid only when the optional MMC module is selected during configuration.

Bit	Attr	Reset Value	Description
3	RO	0x0	PIS PMT Interrupt Status This bit is set whenever a Magic packet or Wake-on-LAN frame is received in Power-Down mode). This bit is cleared when both bits[6:5] are cleared due to a read operation to the register GMAC_PMT_CTRL_STA.
2:1	RO	0x0	reserved
0	RO	0x0	RIS RGMII Interrupt Status This bit is set due to any change in value of the Link Status of RGMII interface. This bit is cleared when the user makes a read operation the RGMII Status register.

GMAC_INT_MASK
Address: Operational Base + offset (0x003c)
Interrupt Mask Register

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3	RW	0x0	PIM PMT Interrupt Mask This bit when set, will disable the assertion of the interrupt signal due to the setting of PMT Interrupt Status bit in Register GMAC_INT_STATUS.
2:1	RO	0x0	reserved
0	RW	0x0	RIM RGMII Interrupt Mask This bit when set, will disable the assertion of the interrupt signal due to the setting of RGMII Interrupt Status bit in Register GMAC_INT_STATUS.

GMAC_MAC_ADDR0_HI
Address: Operational Base + offset (0x0040)
MAC Address0 High Register

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0xffff	A47_A32 MAC Address0 [47:32] This field contains the upper 16 bits (47:32) of the 6-byte first MAC address. This is used by the MAC for filtering for received frames and for inserting the MAC address in the Transmit Flow Control (PAUSE) Frames.

GMAC_MAC_ADDR0_LO
Address: Operational Base + offset (0x0044)
MAC Address0 Low Register

Bit	Attr	Reset Value	Description
31:0	RW	0xffffffff	A31_A0 MAC Address0 [31:0] This field contains the lower 32 bits of the 6-byte first MAC address. This is used by the MAC for filtering for received frames and for inserting the MAC address in the Transmit Flow Control (PAUSE) Frames.

GMAC_AN_CTRL
Address: Operational Base + offset (0x00c0)
AN Control Register

Bit	Attr	Reset Value	Description
31:13	RO	0x0	reserved
12	RW	0x0	ANE Auto-Negotiation Enable When set, will enable the GMAC to perform auto-negotiation with the link partner. Clearing this bit will disable auto-negotiation.
11:10	RO	0x0	reserved
9	R/WSC	0x0	RAN Restart Auto-Negotiation When set, will cause auto-negotiation to restart if the ANE is set. This bit is self-clearing after auto-negotiation starts. This bit should be cleared for normal operation.
8:0	RO	0x0	reserved

GMAC_AN_STATUS
Address: Operational Base + offset (0x00c4)
AN Status Register

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5	RO	0x0	ANC Auto-Negotiation Complete When set, this bit indicates that the auto-negotiation process is completed. This bit is cleared when auto-negotiation is reinitiated.
4	RO	0x0	reserved
3	RO	0x1	ANA Auto-Negotiation Ability This bit is always high, because the GMAC supports auto-negotiation.
2	R/WSC	0x0	LS Link Status When set, this bit indicates that the link is up. When cleared, this bit indicates that the link is down.
1:0	RO	0x0	reserved

GMAC_AN_ADV
Address: Operational Base + offset (0x00c8)
Auto Negotiation Advertisement Register

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15	RO	0x0	NP Next Page Support This bit is tied to low, because the GMAC does not support the next page.
14	RO	0x0	reserved
13:12	RW	0x0	RFE Remote Fault Encoding These 2 bits provide a remote fault encoding, indicating to a link partner that a fault or error condition has occurred.
11:9	RO	0x0	reserved
8:7	RW	0x3	PSE Pause Encoding These 2 bits provide an encoding for the PAUSE bits, indicating that the GMAC is capable of configuring the PAUSE function as defined in IEEE 802.3x.
6	RW	0x1	HD Half-Duplex This bit, when set high, indicates that the GMAC supports Half-Duplex. This bit is tied to low (and RO) when the GMAC is configured for Full-Duplex-only operation.
5	RW	0x1	FD Full-Duplex This bit, when set high, indicates that the GMAC supports Full-Duplex.
4:0	RO	0x0	reserved

GMAC_AN_LINK_PART_AB
Address: Operational Base + offset (0x00cc)
Auto Negotiation Link Partner Ability Register

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15	RO	0x0	NP Next Page Support When set, this bit indicates that more next page information is available. When cleared, this bit indicates that next page exchange is not desired.

Bit	Attr	Reset Value	Description
14	RO	0x0	ACK Acknowledge When set, this bit is used by the auto-negotiation function to indicate that the link partner has successfully received the GMAC's base page. When cleared, it indicates that a successful receipt of the base page has not been achieved.
13:12	RO	0x0	RFE Remote Fault Encoding These 2 bits provide a remote fault encoding, indicating a fault or error condition of the link partner.
11:9	RO	0x0	reserved
8:7	RO	0x0	PSE Pause Encoding These 2 bits provide an encoding for the PAUSE bits, indicating that the link partner's capability of configuring the PAUSE function as defined in IEEE 802.3x.
6	RO	0x0	HD Half-Duplex When set, this bit indicates that the link partner has the ability to operate in Half-Duplex mode. When cleared, the link partner does not have the ability to operate in Half-Duplex mode.
5	RO	0x0	FD Full-Duplex When set, this bit indicates that the link partner has the ability to operate in Full-Duplex mode. When cleared, the link partner does not have the ability to operate in Full-Duplex mode.
4:0	RO	0x0	reserved

GMAC_AN_EXP
Address: Operational Base + offset (0x00d0)
Auto Negotiation Expansion Register

Bit	Attr	Reset Value	Description
31:3	RO	0x0	reserved
2	RO	0x0	NPA Next Page Ability This bit is tied to low, because the GMAC does not support next page function.
1	RO	0x0	NPR New Page Received When set, this bit indicates that a new page has been received by the GMAC. This bit will be cleared when read.
0	RO	0x0	reserved

GMAC_INTF_MODE_STA
Address: Operational Base + offset (0x00d8)

RGMII Status Register

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3	RO	0x0	LST Link Status Indicates whether the link is up (1'b1) or down (1'b0)
2:1	RO	0x0	LSD Link Speed Indicates the current speed of the link: 2'b00: 2.5 MHz 2'b01: 25 MHz 2'b10: 125 MHz
0	RW	0x0	LM Link Mode Indicates the current mode of operation of the link: 1'b0: Half-Duplex mode 1'b1: Full-Duplex mode

GMAC_MMC_CTRL

Address: Operational Base + offset (0x0100)

MMC Control Register

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5	RW	0x0	FHP Full-Half preset When low and bit4 is set, all MMC counters get preset to almost-half value. All octet counters get preset to 0x7FFF_F800 (half - 2K Bytes) and all frame-counters gets preset to 0x7FFF_FFF0 (half - 16) When high and bit4 is set, all MMC counters get preset to almost-full value. All octet counters get preset to 0xFFFF_F800 (full - 2K Bytes) and all frame-counters gets preset to 0xFFFF_FFF0 (full - 16)
4	R/W SC	0x0	CP Counters Preset When set, all counters will be initialized or preset to almost full or almost half as per Bit5 above. This bit will be cleared automatically after 1 clock cycle. This bit along with bit5 is useful for debugging and testing the assertion of interrupts due to MMC counter becoming half-full or full.

Bit	Attr	Reset Value	Description
3	RW	0x0	MCF MMC Counter Freeze When set, this bit freezes all the MMC counters to their current value. (None of the MMC counters are updated due to any transmitted or received frame until this bit is reset to 0. If any MMC counter is read with the Reset on Read bit set, then that counter is also cleared in this mode.)
2	RW	0x0	ROR Reset on Read When set, the MMC counters will be reset to zero after Read (self-clearing after reset). The counters are cleared when the least significant byte lane (bits[7:0]) is read.
1	RW	0x0	CSR Counter Stop Rollover When set, counter after reaching maximum value will not roll over to zero
0	R/W SC	0x0	CR Counters Reset When set, all counters will be reset. This bit will be cleared automatically after 1 clock cycle

GMAC_MMC_RX_INTR
Address: Operational Base + offset (0x0104)
MMC Receive Interrupt Register

Bit	Attr	Reset Value	Description
31:22	RO	0x0	reserved
21	RW	0x0	INT21 The bit is set when the rxfifooverflow counter reaches half the maximum value, and also when it reaches the maximum value.
20:19	RO	0x0	reserved
18	RC	0x0	INT18 The bit is set when the rxlengtherror counter reaches half the maximum value, and also when it reaches the maximum value.
17:6	RO	0x0	reserved
5	RW	0x0	INT5 The bit is set when the rxcrcerror counter reaches half the maximum value, and also when it reaches the maximum value.
4	RC	0x0	INT4 The bit is set when the rxmulticastframes_g counter reaches half the maximum value, and also when it reaches the maximum value.
3	RO	0x0	reserved
2	RC	0x0	INT2 The bit is set when the rxoctetcount_g counter reaches half the maximum value, and also when it reaches the maximum value.

Bit	Attr	Reset Value	Description
1	RC	0x0	INT1 The bit is set when the rxoctetcount_gb counter reaches half the maximum value, and also when it reaches the maximum value.
0	RC	0x0	INT0 The bit is set when the rxframecount_gb counter reaches half the maximum value, and also when it reaches the maximum value.

GMAC_MMC_TX_INTR
Address: Operational Base + offset (0x0108)
MMC Transmit Interrupt Register

Bit	Attr	Reset Value	Description
31:22	RO	0x0	reserved
21	RC	0x0	INT21 The bit is set when the txframecount_g counter reaches half the maximum value, and also when it reaches the maximum value.
20	RC	0x0	INT20 The bit is set when the txoctetcount_g counter reaches half the maximum value, and also when it reaches the maximum value.
19	RC	0x0	INT19 The bit is set when the txcarriererror counter reaches half the maximum value, and also when it reaches the maximum value.
18:14	RO	0x0	reserved
13	RC	0x0	INT13 The bit is set when the txunderflowerror counter reaches half the maximum value, and also when it reaches the maximum value.
12:2	RO	0x0	reserved
1	RC	0x0	INT1 The bit is set when the txframecount_gb counter reaches half the maximum value, and also when it reaches the maximum value.
0	RC	0x0	INT0 The bit is set when the txoctetcount_gb counter reaches half the maximum value, and also when it reaches the maximum value.

GMAC_MMC_RX_INT_MSK
Address: Operational Base + offset (0x010c)
MMC Receive Interrupt Mask Register

Bit	Attr	Reset Value	Description
31:22	RO	0x0	reserved
21	RW	0x0	INT21 Setting this bit masks the interrupt when the rxfifooverflow counter reaches half the maximum value, and also when it reaches the maximum value.
20:19	RO	0x0	reserved

Bit	Attr	Reset Value	Description
18	RW	0x0	INT18 Setting this bit masks the interrupt when the rxlengtherror counter reaches half the maximum value, and also when it reaches the maximum value.
17:6	RO	0x0	reserved
5	RW	0x0	INT5 Setting this bit masks the interrupt when the rxcrcerror counter reaches half the maximum value, and also when it reaches the maximum value.
4	RW	0x0	INT4 Setting this bit masks the interrupt when the rxmulticastframes_g counter reaches half the maximum value, and also when it reaches the maximum value.
3	RO	0x0	reserved
2	RW	0x0	INT2 Setting this bit masks the interrupt when the rxoctetcount_g counter reaches half the maximum value, and also when it reaches the maximum value.
1	RW	0x0	INT1 Setting this bit masks the interrupt when the rxoctetcount_gb counter reaches half the maximum value, and also when it reaches the maximum value.
0	RW	0x0	INT0 Setting this bit masks the interrupt when the rxframecount_gb counter reaches half the maximum value, and also when it reaches the maximum value.

GMAC_MMC_TX_INT_MSK
Address: Operational Base + offset (0x0110)
MMC Transmit Interrupt Mask Register

Bit	Attr	Reset Value	Description
31:22	RO	0x0	reserved
21	RW	0x0	INT21 Setting this bit masks the interrupt when the txframecount_g counter reaches half the maximum value, and also when it reaches the maximum value.
20	RW	0x0	INT20 Setting this bit masks the interrupt when the txoctetcount_g counter reaches half the maximum value, and also when it reaches the maximum value.
19	RW	0x0	INT19 Setting this bit masks the interrupt when the txcarriererror counter reaches half the maximum value, and also when it reaches the maximum value.
18:14	RO	0x0	reserved

Bit	Attr	Reset Value	Description
13	RW	0x0	INT13 Setting this bit masks the interrupt when the txunderflowerror counter reaches half the maximum value, and also when it reaches the maximum value.
12:2	RO	0x0	reserved
1	RW	0x0	INT1 Setting this bit masks the interrupt when the txframecount_gb counter reaches half the maximum value, and also when it reaches the maximum value.
0	RW	0x0	INT0 Setting this bit masks the interrupt when the txoctetcount_gb counter reaches half the maximum value, and also when it reaches the maximum value.

GMAC_MMC_TXOCTETCNT_GB
Address: Operational Base + offset (0x0114)
MMC TX OCTET Good and Bad Counter

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	txoctetcount_gb Number of bytes transmitted, exclusive of preamble and retried bytes, in good and bad frames.

GMAC_MMC_TXFRMCNT_GB
Address: Operational Base + offset (0x0118)
MMC TX Frame Good and Bad Counter

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	txframecount_gb Number of good and bad frames transmitted, exclusive of retried frames.

GMAC_MMC_TXUNDFLWERR
Address: Operational Base + offset (0x0148)
MMC TX Underflow Error

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	txunderflowerror Number of frames aborted due to frame underflow error.

GMAC_MMC_TXCARERR
Address: Operational Base + offset (0x0160)
MMC TX Carrier Error

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	txcarriererror Number of frames aborted due to carrier sense error (no carrier or loss of carrier).

GMAC_MMC_TXOCTETCNT_G
Address: Operational Base + offset (0x0164)
MMC TX OCTET Good Counter

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	txoctetcount_g Number of bytes transmitted, exclusive of preamble, in good frames only.

GMAC_MMC_TXFRMCNT_G
Address: Operational Base + offset (0x0168)
MMC TX Frame Good Counter

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	txframecount_g Number of good frames transmitted.

GMAC_MMC_RXFRMCNT_GB
Address: Operational Base + offset (0x0180)
MMC RX Frame Good and Bad Counter

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rxframecount_gb Number of good and bad frames received.

GMAC_MMC_RXOCTETCNT_GB
Address: Operational Base + offset (0x0184)
MMC RX OCTET Good and Bad Counter

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rxoctetcount_gb Number of bytes received, exclusive of preamble, in good and bad frames.

GMAC_MMC_RXOCTETCNT_G
Address: Operational Base + offset (0x0188)
MMC RX OCTET Good Counter

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rxoctetcount_g Number of bytes received, exclusive of preamble, only in good frames.

GMAC_MMC_RXMCFRMCNT_G
Address: Operational Base + offset (0x0190)
MMC RX Mulitcast Frame Good Counter

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rxmulticastframes_g Number of good multicast frames received.

GMAC_MMC_RXCRCERR
Address: Operational Base + offset (0x0194)
MMC RX Carrier

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rxcrcerror Number of frames received with CRC error.

GMAC_MMC_RXLENERR
Address: Operational Base + offset (0x01c8)
MMC RX Length Error

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rxlengtherror Number of frames received with length error (Length type field ≠frame size), for all frames with valid length field.

GMAC_MMC_RXFIFOOVRFLW
Address: Operational Base + offset (0x01d4)
MMC RX FIFO Overflow

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rxfifooverflow Number of missed received frames due to FIFO overflow.

GMAC_MMC_IPC_INT_MSK
Address: Operational Base + offset (0x0200)
MMC Receive Checksum Offload Interrupt Mask Register

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved
29	RW	0x0	INT29 Setting this bit masks the interrupt when the rxicmp_err_octets counter reaches half the maximum value, and also when it reaches the maximum value.
28	RO	0x0	reserved
27	RW	0x0	INT27 Setting this bit masks the interrupt when the rxtcp_err_octets counter reaches half the maximum value, and also when it reaches the maximum value.
26	RO	0x0	reserved
25	RW	0x0	INT25 Setting this bit masks the interrupt when the rxudp_err_octets counter reaches half the maximum value, and also when it reaches the maximum value.
24:23	RO	0x0	reserved
22	RW	0x0	INT22 Setting this bit masks the interrupt when the rxipv6_hdrerr_octets counter reaches half the maximum value, and also when it reaches the maximum value.

Bit	Attr	Reset Value	Description
21:18	RO	0x0	reserved
17	RW	0x0	INT17 Setting this bit masks the interrupt when the rxipv4_hdrerr_octets counter reaches half the maximum value, and also when it reaches the maximum value.
16:14	RO	0x0	reserved
13	RW	0x0	INT13 Setting this bit masks the interrupt when the rxicmp_err_frms counter reaches half the maximum value, and also when it reaches the maximum value.
12	RO	0x0	reserved
11	RW	0x0	INT11 Setting this bit masks the interrupt when the rxtcp_err_frms counter reaches half the maximum value, and also when it reaches the maximum value.
10	RO	0x0	reserved
9	RW	0x0	INT9 Setting this bit masks the interrupt when the rxudp_err_frms counter reaches half the maximum value, and also when it reaches the maximum value.
8:7	RO	0x0	reserved
6	RW	0x0	INT6 Setting this bit masks the interrupt when the rxipv6_hdrerr_frms counter reaches half the maximum value, and also when it reaches the maximum value.
5	RW	0x0	INT5 Setting this bit masks the interrupt when the rxipv6_gd_frms counter reaches half the maximum value, and also when it reaches the maximum value.
4:2	RO	0x0	reserved
1	RW	0x0	INT1 Setting this bit masks the interrupt when the rxipv4_hdrerr_frms counter reaches half the maximum value, and also when it reaches the maximum value.
0	RW	0x0	INT0 Setting this bit masks the interrupt when the rxipv4_gd_frms counter reaches half the maximum value, and also when it reaches the maximum value.

GMAC_MMC_IPC_INTR
Address: Operational Base + offset (0x0208)
MMC Receive Checksum Offload Interrupt Register

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved

Bit	Attr	Reset Value	Description
29	RC	0x0	INT29 The bit is set when the rxicmp_err_octets counter reaches half the maximum value, and also when it reaches the maximum value.
28	RO	0x0	reserved
27	RC	0x0	INT27 The bit is set when the rxtcp_err_octets counter reaches half the maximum value, and also when it reaches the maximum value.
26	RO	0x0	reserved
25	RC	0x0	INT25 The bit is set when the rxudp_err_octets counter reaches half the maximum value, and also when it reaches the maximum value.
24:23	RO	0x0	reserved
22	RC	0x0	INT22 The bit is set when the rxipv6_hdrerr_octets counter reaches half the maximum value, and also when it reaches the maximum value.
21:18	RO	0x0	reserved
17	RC	0x0	INT17 The bit is set when the rxipv4_hdrerr_octets counter reaches half the maximum value, and also when it reaches the maximum value.
16:14	RO	0x0	reserved
13	RC	0x0	INT13 The bit is set when the rxicmp_err_frms counter reaches half the maximum value, and also when it reaches the maximum value.
12	RO	0x0	reserved
11	RC	0x0	INT11 The bit is set when the rxtcp_err_frms counter reaches half the maximum value, and also when it reaches the maximum value.
10	RO	0x0	reserved
9	RC	0x0	INT9 The bit is set when the rxudp_err_frms counter reaches half the maximum value, and also when it reaches the maximum value.
8:7	RO	0x0	reserved
6	RC	0x0	INT6 The bit is set when the rxipv6_hdrerr_frms counter reaches half the maximum value, and also when it reaches the maximum value.
5	RC	0x0	INT5 The bit is set when the rxipv6_gd_frms counter reaches half the maximum value, and also when it reaches the maximum value.
4:2	RO	0x0	reserved

Bit	Attr	Reset Value	Description
1	RC	0x0	INT1 The bit is set when the rxipv4_hdrerr_frms counter reaches half the maximum value, and also when it reaches the maximum value.
0	RC	0x0	INT0 The bit is set when the rxipv4_gd_frms counter reaches half the maximum value, and also when it reaches the maximum value.

GMAC_MMC_RXIPV4GFRM
Address: Operational Base + offset (0x0210)
MMC RX IPV4 Good Frame

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rxipv4_gd_frms Number of good IPv4 datagrams received with the TCP, UDP, or ICMP payload

GMAC_MMC_RXIPV4HDERRFRM
Address: Operational Base + offset (0x0214)
MMC RX IPV4 Head Error Frame

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rxipv4_hdrerr_frms Number of IPv4 datagrams received with header (checksum, length, or version mismatch) errors

GMAC_MMC_RXIPV6GFRM
Address: Operational Base + offset (0x0224)
MMC RX IPV6 Good Frame

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rxipv6_gd_frms Number of good IPv6 datagrams received with TCP, UDP, or ICMP payloads.

GMAC_MMC_RXIPV6HDERRFRM
Address: Operational Base + offset (0x0228)
MMC RX IPV6 Head Error Frame

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rxipv6_hdrerr_frms Number of IPv6 datagrams received with header errors (length or version mismatch).

GMAC_MMC_RXUDPERRFRM
Address: Operational Base + offset (0x0234)
MMC RX UDP Error Frame

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rxudp_err_frms Number of good IP datagrams whose UDP payload has a checksum error.

GMAC_MMC_RXTCPERRFRM
Address: Operational Base + offset (0x023c)
MMC RX TCP Error Frame

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rxtcp_err_frms Number of good IP datagrams whose TCP payload has a checksum error.

GMAC_MMC_RXICMPERRFRM
Address: Operational Base + offset (0x0244)
MMC RX ICMP Error Frame

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rxicmp_err_frms Number of good IP datagrams whose ICMP payload has a checksum error.

GMAC_MMC_RXIPV4HDERROCT
Address: Operational Base + offset (0x0254)
MMC RX OCTET IPV4 Head Error

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rxipv4_hdrerr_octets Number of bytes received in IPv4 datagrams with header errors (checksum, length, version mismatch). The value in the Length field of IPv4 header is used to update this counter.

GMAC_MMC_RXIPV6HDERROCT
Address: Operational Base + offset (0x0268)
MMC RX OCTET IPV6 Head Error

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rxipv6_hdrerr_octets Number of bytes received in IPv6 datagrams with header errors (length, version mismatch). The value in the IPv6 header's Length field is used to update this counter.

GMAC_MMC_RXUDPERROCT
Address: Operational Base + offset (0x0274)
MMC RX OCTET UDP Error

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rxudp_err_octets Number of bytes received in a UDP segment that had checksum errors.

GMAC_MMC_RXTCPERROCT
Address: Operational Base + offset (0x027c)
MMC RX OCTET TCP Error

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rxtcp_err_octets Number of bytes received in a TCP segment with checksum errors.

GMAC_MMC_RXICMPERROCT
Address: Operational Base + offset (0x0284)
MMC RX OCTET ICMP Error

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rxicmp_err_octets Number of bytes received in an ICMP segment with checksum errors.

GMAC_BUS_MODE
Address: Operational Base + offset (0x1000)
Bus Mode Register

Bit	Attr	Reset Value	Description
31:26	RO	0x0	reserved
25	RW	0x0	AAL Address-Aligned Beats When this bit is set high and the FB bit equals 1, the AXI interface generates all bursts aligned to the start address LS bits. If the FB bit equals 0, the first burst (accessing the data buffer's start address) is not aligned, but subsequent bursts are aligned to the address.
24	RW	0x0	PBL_Mode 8xPBL Mode When set high, this bit multiplies the PBL value programmed (bits [22:17] and bits [13:8]) eight times. Thus the DMA will transfer data in to a maximum of 8, 16, 32, 64, 128, and 256 beats depending on the PBL value.
23	RW	0x0	USP Use Separate PBL When set high, it configures the RxDMA to use the value configured in bits [22:17] as PBL while the PBL value in bits [13:8] is applicable to TxDMA operations only. When reset to low, the PBL value in bits [13:8] is applicable for both DMA engines.

Bit	Attr	Reset Value	Description
22:17	RW	0x01	RPBL RxDMA PBL These bits indicate the maximum number of beats to be transferred in one RxDMA transaction. This will be the maximum value that is used in a single block Read/Write. The RxDMA will always attempt to burst as specified in RPBL each time it starts a Burst transfer on the host bus. RPBL can be programmed with permissible values of 1, 2, 4, 8, 16, and 32. Any other value will result in undefined behavior. These bits are valid and applicable only when USP is set high.
16	RW	0x0	FB Fixed Burst This bit controls whether the AXI Master interface performs fixed burst transfers or not. When set, the AHB will use only SINGLE, INCR4, INCR8 or INCR16 during start of normal burst transfers. When reset, the AXI will use SINGLE and INCR burst transfer operations.
15:14	RO	0x0	reserved
13:8	RW	0x01	PBL Programmable Burst Length These bits indicate the maximum number of beats to be transferred in one DMA transaction. This will be the maximum value that is used in a single block Read/Write. The DMA will always attempt to burst as specified in PBL each time it starts a Burst transfer on the host bus. PBL can be programmed with permissible values of 1, 2, 4, 8, 16, and 32. Any other value will result in undefined behavior. When USP is set high, this PBL value is applicable for TxDMA transactions only. The PBL values have the following limitations. The maximum number of beats (PBL) possible is limited by the size of the Tx FIFO and Rx FIFO in the MTL layer and the data bus width on the DMA. The FIFO has a constraint that the maximum beat supported is half the depth of the FIFO, except when specified (as given below). For different data bus widths and FIFO sizes, the valid PBL range (including x8 mode) is provided in the following table. If the PBL is common for both transmit and receive DMA, the minimum Rx FIFO and Tx FIFO depths must be considered. Do not program out-of-range PBL values, because the system may not behave properly. For TxFIFO, valid PBL range in full duplex mode and duplex mode is 128 or less. For RxFIFO, valid PBL range in full duplex mode is all.
7	RO	0x0	reserved

Bit	Attr	Reset Value	Description
6:2	RW	0x00	DSL Descriptor Skip Length This bit specifies the number of dword to skip between two unchained descriptors. The address skipping starts from the end of current descriptor to the start of next descriptor. When DSL value equals zero, then the descriptor table is taken as contiguous by the DMA, in Ring mode.
1	RO	0x0	reserved
0	R/WSC	0x1	SWR Software Reset When this bit is set, the MAC DMA Controller resets all GMAC Subsystem internal registers and logic. It is cleared automatically after the reset operation has completed in all of the core clock domains. Read a 0 value in this bit before re-programming any register of the core. Note: The reset operation is completed only when all the resets in all the active clock domains are de-asserted. Hence it is essential that all the PHY inputs clocks (applicable for the selected PHY interface) are present for software reset completion.

GMAC_TX_POLL_DEMAND
Address: Operational Base + offset (0x1004)
Transmit Poll Demand Register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	TPD Transmit Poll Demand When these bits are written with any value, the DMA reads the current descriptor pointed to by Register GMAC_CUR_HOST_TX_DESC. If that descriptor is not available (owned by Host), transmission returns to the Suspend state and DMA Register GMAC_STATUS[2] is asserted. If the descriptor is available, transmission resumes.

GMAC_RX_POLL_DEMAND
Address: Operational Base + offset (0x1008)
Receive Poll Demand Register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	RPD Receive Poll Demand When these bits are written with any value, the DMA reads the current descriptor pointed to by Register GMAC_CUR_HOST_RX_DESC. If that descriptor is not available (owned by Host), reception returns to the Suspended state and Register GMAC_STATUS[7] is not asserted. If the descriptor is available, the Receive DMA returns to active state.

GMAC_RX_DESC_LIST_ADDR
Address: Operational Base + offset (0x100c)
Receive Descriptor List Address Register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	SRL Start of Receive List This field contains the base address of the First Descriptor in the Receive Descriptor list. The LSB bits [1/2/3:0] for 32/64/128-bit bus width) will be ignored and taken as all-zero by the DMA internally. Hence these LSB bits are Read Only.

GMAC_TX_DESC_LIST_ADDR
Address: Operational Base + offset (0x1010)
Transmit Descriptor List Address Register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	STL Start of Transmit List This field contains the base address of the First Descriptor in the Transmit Descriptor list. The LSB bits [1/2/3:0] for 32/64/128-bit bus width) will be ignored and taken as all-zero by the DMA internally. Hence these LSB bits are Read Only.

GMAC_STATUS
Address: Operational Base + offset (0x1014)
Status Register

Bit	Attr	Reset Value	Description
31:29	RO	0x0	reserved
28	RO	0x0	GPI GMAC PMT Interrupt This bit indicates an interrupt event in the GMAC core's PMT module. The software must read the corresponding registers in the GMAC core to get the exact cause of interrupt and clear its source to reset this bit to 1'b0. The interrupt signal from the GMAC subsystem (sbd_intr_o) is high when this bit is high.
27	RO	0x0	GMI GMAC MMC Interrupt This bit reflects an interrupt event in the MMC module of the GMAC core. The software must read the corresponding registers in the GMAC core to get the exact cause of interrupt and clear the source of interrupt to make this bit as 1'b0. The interrupt signal from the GMAC subsystem (sbd_intr_o) is high when this bit is high.

Bit	Attr	Reset Value	Description
26	RO	0x0	GLI GMAC Line interface Interrupt This bit reflects an interrupt event in the GMAC Core's PCS or RGMII interface block. The software must read the corresponding registers in the GMAC core to get the exact cause of interrupt and clear the source of interrupt to make this bit as 1'b0. The interrupt signal from the GMAC subsystem (sbd_intr_o) is high when this bit is high.
25:23	RO	0x0	EB Error Bits These bits indicate the type of error that caused a Bus Error (e.g., error response on the AXI interface). Valid only with Fatal Bus Error bit (Register GMAC_STATUS[13]) set. This field does not generate an interrupt. Bit 23: 1'b1 Error during data transfer by TxDMA 1'b0 Error during data transfer by RxDMA Bit 24: 1'b1 Error during read transfer 1'b0 Error during write transfer Bit 25: 1'b1 Error during descriptor access 1'b0 Error during data buffer access
22:20	RO	0x0	TS Transmit Process State These bits indicate the Transmit DMA FSM state. This field does not generate an interrupt. 3'b000: Stopped; Reset or Stop Transmit Command issued. 3'b001: Running; Fetching Transmit Transfer Descriptor. 3'b010: Running; Waiting for status. 3'b011: Running; Reading Data from host memory buffer and queuing it to transmit buffer (Tx FIFO). 3'b100: TIME_STAMP write state. 3'b101: Reserved for future use. 3'b110: Suspended; Transmit Descriptor Unavailable or Transmit Buffer Underflow. 3'b111: Running; Closing Transmit Descriptor.

Bit	Attr	Reset Value	Description
19:17	RO	0x0	RS Receive Process State These bits indicate the Receive DMA FSM state. This field does not generate an interrupt. 3'b000: Stopped: Reset or Stop Receive Command issued. 3'b001: Running: Fetching Receive Transfer Descriptor. 3'b010: Reserved for future use. 3'b011: Running: Waiting for receive packet. 3'b100: Suspended: Receive Descriptor Unavailable. 3'b101: Running: Closing Receive Descriptor. 3'b110: TIME_STAMP write state. 3'b111: Running: Transferring the receive packet data from receive buffer to host memory.
16	W1C	0x0	NIS Normal Interrupt Summary Normal Interrupt Summary bit value is the logical OR of the following when the corresponding interrupt bits are enabled in Register OP_MODE: Register GMAC_STATUS[0]: Transmit Interrupt Register GMAC_STATUS[2]: Transmit Buffer Unavailable Register GMAC_STATUS[6]: Receive Interrupt Register GMAC_STATUS[14]: Early Receive Interrupt Only unmasked bits affect the Normal Interrupt Summary bit. This is a sticky bit and must be cleared (by writing a 1 to this bit) each time a corresponding bit that causes NIS to be set is cleared.
15	W1C	0x0	AIS Abnormal Interrupt Summary Abnormal Interrupt Summary bit value is the logical OR of the following when the corresponding interrupt bits are enabled in Register OP_MODE: Register GMAC_STATUS[1]: Transmit Process Stopped Register GMAC_STATUS[3]: Transmit Jabber Timeout Register GMAC_STATUS[4]: Receive FIFO Overflow Register GMAC_STATUS[5]: Transmit Underflow Register GMAC_STATUS[7]: Receive Buffer Unavailable Register GMAC_STATUS[8]: Receive Process Stopped Register GMAC_STATUS[9]: Receive Watchdog Timeout Register GMAC_STATUS[10]: Early Transmit Interrupt Register GMAC_STATUS[13]: Fatal Bus Error Only unmasked bits affect the Abnormal Interrupt Summary bit. This is a sticky bit and must be cleared each time a corresponding bit that causes AIS to be set is cleared.

Bit	Attr	Reset Value	Description
14	W1C	0x0	ERI Early Receive Interrupt This bit indicates that the DMA had filled the first data buffer of the packet. Receive Interrupt Register GMAC_STATUS[6] automatically clears this bit.
13	W1C	0x0	FBI Fatal Bus Error Interrupt This bit indicates that a bus error occurred, as detailed in [25:23]. When this bit is set, the corresponding DMA engine disables all its bus accesses.
12:11	RO	0x0	reserved
10	W1C	0x0	ETI Early Transmit Interrupt This bit indicates that the frame to be transmitted was fully transferred to the MTL Transmit FIFO.
9	W1C	0x0	RWT Receive Watchdog Timeout This bit is asserted when a frame with a length greater than 2,048 bytes is received.
8	W1C	0x0	RPS Receive Process Stopped This bit is asserted when the Receive Process enters the Stopped state.
7	W1C	0x0	RU Receive Buffer Unavailable This bit indicates that the Next Descriptor in the Receive List is owned by the host and cannot be acquired by the DMA. Receive Process is suspended. To resume processing Receive descriptors, the host should change the ownership of the descriptor and issue a Receive Poll Demand command. If no Receive Poll Demand is issued, Receive Process resumes when the next recognized incoming frame is received. Register GMAC_STATUS[7] is set only when the previous Receive Descriptor was owned by the DMA.
6	W1C	0x0	RI Receive Interrupt This bit indicates the completion of frame reception. Specific frame status information has been posted in the descriptor. Reception remains in the Running state.
5	W1C	0x0	UNF Transmit Underflow This bit indicates that the Transmit Buffer had an Underflow during frame transmission. Transmission is suspended and an Underflow Error TDES0[1] is set.

Bit	Attr	Reset Value	Description
4	W1C	0x0	OVF Receive Overflow This bit indicates that the Receive Buffer had an Overflow during frame reception. If the partial frame is transferred to application, the overflow status is set in RDES0[11].
3	W1C	0x0	TJT Transmit Jabber Timeout This bit indicates that the Transmit Jabber Timer expired, meaning that the transmitter had been excessively active. The transmission process is aborted and placed in the Stopped state. This causes the Transmit Jabber Timeout TDES0[14] flag to assert.
2	W1C	0x0	TU Transmit Buffer Unavailable This bit indicates that the Next Descriptor in the Transmit List is owned by the host and cannot be acquired by the DMA. Transmission is suspended. Bits[22:20] explain the Transmit Process state transitions. To resume processing transmit descriptors, the host should change the ownership of the bit of the descriptor and then issue a Transmit Poll Demand command.
1	W1C	0x0	TPS Transmit Process Stopped This bit is set when the transmission is stopped.
0	W1C	0x0	TI Transmit Interrupt This bit indicates that frame transmission is finished and TDES1[31] is set in the First Descriptor.

GMAC_OP_MODE
Address: Operational Base + offset (0x1018)
Operation Mode Register

Bit	Attr	Reset Value	Description
31:27	RO	0x0	reserved
26	RW	0x0	DT Disable Dropping of TCP/IP Checksum Error Frames When this bit is set, the core does not drop frames that only have errors detected by the Receive Checksum Offload engine. Such frames do not have any errors (including FCS error) in the Ethernet frame received by the MAC but have errors in the encapsulated payload only. When this bit is reset, all error frames are dropped if the FEF bit is reset.

Bit	Attr	Reset Value	Description
25	RW	0x0	RSF Receive Store and Forward When this bit is set, the MTL only reads a frame from the Rx FIFO after the complete frame has been written to it, ignoring RTC bits. When this bit is reset, the Rx FIFO operates in Cut-Through mode, subject to the threshold specified by the RTC bits.
24	RW	0x0	DFF Disable Flushing of Received Frames When this bit is set, the RxDMA does not flush any frames due to the unavailability of receive descriptors/buffers as it does normally when this bit is reset.
23:22	RO	0x0	reserved
21	RW	0x0	TSF Transmit Store and Forward When this bit is set, transmission starts when a full frame resides in the MTL Transmit FIFO. When this bit is set, the TTC values specified in Register GMAC_OP_MODE[16:14] are ignored. This bit should be changed only when transmission is stopped.
20	W1C	0x0	FTF Flush Transmit FIFO When this bit is set, the transmit FIFO controller logic is reset to its default values and thus all data in the Tx FIFO is lost/flushed. This bit is cleared internally when the flushing operation is completed fully. The Operation Mode register should not be written to until this bit is cleared. The data which is already accepted by the MAC transmitter will not be flushed. It will be scheduled for transmission and will result in underflow and runt frame transmission. Note: The flush operation completes only after emptying the TxFIFO of its contents and all the pending Transmit Status of the transmitted frames are accepted by the host. In order to complete this flush operation, the PHY transmit clock (clk_tx_i) is required to be active.
19:17	RO	0x0	reserved

Bit	Attr	Reset Value	Description
16:14	RW	0x0	TTC Transmit Threshold Control These three bits control the threshold level of the MTL Transmit FIFO. Transmission starts when the frame size within the MTL Transmit FIFO is larger than the threshold. In addition, full frames with a length less than the threshold are also transmitted. These bits are used only when the TSF bit (Bit 21) is reset. 3'b000: 64 3'b001: 128 3'b010: 192 3'b011: 256 3'b100: 40 3'b101: 32 3'b110: 24 3'b111: 16
13	RW	0x0	ST Start/Stop Transmission Command When this bit is set, transmission is placed in the Running state, and the DMA checks the Transmit List at the current position for a frame to be transmitted. Descriptor acquisition is attempted either from the current position in the list, which is the Transmit List Base Address set by Register GMAC_TX_DESC_LIST_ADDR, or from the position retained when transmission was stopped previously. If the current descriptor is not owned by the DMA, transmission enters the Suspended state and Transmit Buffer Unavailable (Register GMAC_STATUS[2]) is set. The Start Transmission command is effective only when transmission is stopped. If the command is issued before setting DMA Register TX_DESC_LIST_ADDR, then the DMA behavior is unpredictable. When this bit is reset, the transmission process is placed in the Stopped state after completing the transmission of the current frame. The Next Descriptor position in the Transmit List is saved, and becomes the current position when transmission is restarted. The stop transmission command is effective only the transmission of the current frame is complete or when the transmission is in the Suspended state.
12:11	RW	0x0	RFD Threshold for deactivating flow control (in both HD and FD) These bits control the threshold (Fill-level of Rx FIFO) at which the flow-control is de-asserted after activation. 2'b00: Full minus 1 KB 2'b01: Full minus 2 KB 2'b10: Full minus 3 KB 2'b11: Full minus 4 KB Note that the de-assertion is effective only after flow control is asserted.

Bit	Attr	Reset Value	Description
10:9	RW	0x0	RFA Threshold for activating flow control (in both HD and FD) These bits control the threshold (Fill level of Rx FIFO) at which flow control is activated. 2'b00: Full minus 1 KB 2'b01: Full minus 2 KB 2'b10: Full minus 3 KB 2'b11: Full minus 4 KB Note that the above only applies to Rx FIFOs of 4 KB or more when the EFC bit is set high.
8	RW	0x0	EFC Enable HW flow control When this bit is set, the flow control signal operation based on fill-level of Rx FIFO is enabled. When reset, the flow control operation is disabled.
7	RW	0x0	FEF Forward Error Frames When this bit is reset, the Rx FIFO drops frames with error status (CRC error, collision error, GMII_ER, giant frame, watchdog timeout, overflow). However, if the frame's start byte (write) pointer is already transferred to the read controller side (in Threshold mode), then the frames are not dropped. When FEF is set, all frames except runt error frames are forwarded to the DMA. But when RxFIFO overflows when a partial frame is written, then such frames are dropped even when FEF is set.
6	RW	0x0	FUF Forward Undersized Good Frames When set, the Rx FIFO will forward Undersized frames (frames with no Error and length less than 64 bytes) including pad-bytes and CRC). When reset, the Rx FIFO will drop all frames of less than 64 bytes, unless it is already transferred due to lower value of Receive Threshold (e.g., RTC = 01).
5	RO	0x0	reserved

Bit	Attr	Reset Value	Description
4:3	RW	0x0	RTC Receive Threshold Control These two bits control the threshold level of the MTL Receive FIFO. Transfer (request) to DMA starts when the frame size within the MTL Receive FIFO is larger than the threshold. In addition, full frames with a length less than the threshold are transferred automatically. Note that value of 11 is not applicable if the configured Receive FIFO size is 128 bytes. These bits are valid only when the RSF bit is zero, and are ignored when the RSF bit is set to 1. 2'b00: 64 2'b01: 32 2'b10: 96 2'b11: 128
2	RW	0x0	OSF Operate on Second Frame When this bit is set, this bit instructs the DMA to process a second frame of Transmit data even before status for first frame is obtained.
1	RW	0x0	SR Start/Stop Receive When this bit is set, the Receive process is placed in the Running state. The DMA attempts to acquire the descriptor from the Receive list and processes incoming frames. Descriptor acquisition is attempted from the current position in the list, which is the address set by register GMAC_RX_DESC_LIST_ADDR or the position retained when the Receive process was previously stopped. If no descriptor is owned by the DMA, reception is suspended and Receive Buffer Unavailable (Register GMAC_STATUS[7]) is set. The Start Receive command is effective only when reception has stopped. If the command was issued before setting register GMAC_RX_DESC_LIST_ADDR, DMA behavior is unpredictable. When this bit is cleared, RxDMA operation is stopped after the transfer of the current frame. The next descriptor position in the Receive list is saved and becomes the current position after the Receive process is restarted. The Stop Receive command is effective only when the Receive process is in either the Running (waiting for receive packet) or in the Suspended state.
0	RO	0x0	reserved

GMAC_INT_ENA
Address: Operational Base + offset (0x101c)
Interrupt Enable Register

Bit	Attr	Reset Value	Description
31:17	RO	0x0	reserved
16	RW	0x0	NIE Normal Interrupt Summary Enable When this bit is set, a normal interrupt is enabled. When this bit is reset, a normal interrupt is disabled. This bit enables the following bits: Register GMAC_STATUS[0]: Transmit Interrupt Register GMAC_STATUS[2]: Transmit Buffer Unavailable Register GMAC_STATUS[6]: Receive Interrupt Register GMAC_STATUS[14]: Early Receive Interrupt
15	RW	0x0	AIE Abnormal Interrupt Summary Enable When this bit is set, an Abnormal Interrupt is enabled. When this bit is reset, an Abnormal Interrupt is disabled. This bit enables the following bits Register GMAC_STATUS[1]: Transmit Process Stopped Register GMAC_STATUS[3]: Transmit Jabber Timeout Register GMAC_STATUS[4]: Receive Overflow Register GMAC_STATUS[5]: Transmit Underflow Register GMAC_STATUS[7]: Receive Buffer Unavailable Register GMAC_STATUS[8]: Receive Process Stopped Register GMAC_STATUS[9]: Receive Watchdog Timeout Register GMAC_STATUS[10]: Early Transmit Interrupt Register GMAC_STATUS[13]: Fatal Bus Error
14	RW	0x0	ERE Early Receive Interrupt Enable When this bit is set with Normal Interrupt Summary Enable (BIT 16), Early Receive Interrupt is enabled. When this bit is reset, Early Receive Interrupt is disabled.
13	RW	0x0	FBE Fatal Bus Error Enable When this bit is set with Abnormal Interrupt Summary Enable (BIT 15), the Fatal Bus Error Interrupt is enabled. When this bit is reset, Fatal Bus Error Enable Interrupt is disabled.
12:11	RO	0x0	reserved
10	RW	0x0	ETE Early Transmit Interrupt Enable When this bit is set with an Abnormal Interrupt Summary Enable (BIT 15), Early Transmit Interrupt is enabled. When this bit is reset, Early Transmit Interrupt is disabled.

Bit	Attr	Reset Value	Description
9	RW	0x0	RWE Receive Watchdog Timeout Enable When this bit is set with Abnormal Interrupt Summary Enable (BIT 15), the Receive Watchdog Timeout Interrupt is enabled. When this bit is reset, Receive Watchdog Timeout Interrupt is disabled.
8	RW	0x0	RSE Receive Stopped Enable When this bit is set with Abnormal Interrupt Summary Enable (BIT 15), Receive Stopped Interrupt is enabled. When this bit is reset, Receive Stopped Interrupt is disabled.
7	RW	0x0	RUE Receive Buffer Unavailable Enable When this bit is set with Abnormal Interrupt Summary Enable (BIT 15), Receive Buffer Unavailable Interrupt is enabled. When this bit is reset, the Receive Buffer Unavailable Interrupt is disabled
6	RW	0x0	RIE Receive Interrupt Enable When this bit is set with Normal Interrupt Summary Enable (BIT 16), Receive Interrupt is enabled. When this bit is reset, Receive Interrupt is disabled.
5	RW	0x0	UNE Underflow Interrupt Enable When this bit is set with Abnormal Interrupt Summary Enable (BIT 15), Transmit Underflow Interrupt is enabled. When this bit is reset, Underflow Interrupt is disabled.
4	RW	0x0	OVE Overflow Interrupt Enable When this bit is set with Abnormal Interrupt Summary Enable (BIT 15), Receive Overflow Interrupt is enabled. When this bit is reset, Overflow Interrupt is disabled
3	RW	0x0	TJE Transmit Jabber Timeout Enable When this bit is set with Abnormal Interrupt Summary Enable (BIT 15), Transmit Jabber Timeout Interrupt is enabled. When this bit is reset, Transmit Jabber Timeout Interrupt is disabled.
2	RW	0x0	TUE Transmit Buffer Unavailable Enable When this bit is set with Normal Interrupt Summary Enable (BIT 16), Transmit Buffer Unavailable Interrupt is enabled. When this bit is reset, Transmit Buffer Unavailable Interrupt is disabled.

Bit	Attr	Reset Value	Description
1	RW	0x0	TSE Transmit Stopped Enable When this bit is set with Abnormal Interrupt Summary Enable (BIT 15), Transmission Stopped Interrupt is enabled. When this bit is reset, Transmission Stopped Interrupt is disabled.
0	RW	0x0	TIE Transmit Interrupt Enable When this bit is set with Normal Interrupt Summary Enable (BIT 16), Transmit Interrupt is enabled. When this bit is reset, Transmit Interrupt is disabled.

GMAC_OVERFLOW_CNT
Address: Operational Base + offset (0x1020)
Missed Frame and Buffer Overflow Counter Register

Bit	Attr	Reset Value	Description
31:29	RO	0x0	reserved
28	RC	0x0	FIFO_overflow_bit Overflow bit for FIFO Overflow Counter
27:17	RC	0x000	Frame_miss_number Indicates the number of frames missed by the application This counter is incremented each time the MTL asserts the sideband signal mtl_rxoverflow_o. The counter is cleared when this register is read with mci_be_i[2] at 1'b1.
16	RC	0x0	Miss_frame_overflow_bit Overflow bit for Missed Frame Counter
15:0	RC	0x0000	Frame_miss_number_2 Indicates the number of frames missed by the controller due to the Host Receive Buffer being unavailable. This counter is incremented each time the DMA discards an incoming frame. The counter is cleared when this register is read with mci_be_i[0] at 1'b1.

GMAC_REC_INT_WDT_TIMER
Address: Operational Base + offset (0x1024)
Receive Interrupt Watchdog Timer Register

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved

Bit	Attr	Reset Value	Description
7:0	RW	0x00	RIWT RI Watchdog Timer count Indicates the number of system clock cycles multiplied by 256 for which the watchdog timer is set. The watchdog timer gets triggered with the programmed value after the RxDMA completes the transfer of a frame for which the RI status bit is not set due to the setting in the corresponding descriptor RDES1[31]. When the watch-dog timer runs out, the RI bit is set and the timer is stopped. The watchdog timer is reset when RI bit is set high due to automatic setting of RI as per RDES1[31] of any received frame.

GMAC_AXI_BUS_MODE
Address: Operational Base + offset (0x1028)
AXI Bus Mode Register

Bit	Attr	Reset Value	Description
31	RW	0x0	EN_LPI Enable LPI (Low Power Interface) When set to 1, enable the LPI (Low Power Interface) supported by the GMAC and accepts the LPI request from the AXI System Clock controller. When set to 0, disables the Low Power Mode and always denies the LPI request from the AXI System Clock controller.
30	RW	0x0	UNLCK_ON_MGK_RWK Unlock on Magic Packet or Remote Wake Up When set to 1, enables it to request coming out of Low Power mode only when Magic Packet or Remote Wake Up Packet is received. When set to 0, enables it requests to come out of Low Power mode when any frame is received.
29:22	RO	0x0	reserved
21:20	RW	0x1	WR_OSR_LMT AXI Maximum Write Out Standing Request Limit This value limits the maximum outstanding request on the AXI write interface. Maximum outstanding requests = WR_OSR_LMT+1
19:18	RO	0x0	reserved
17:16	RW	0x1	RD_OSR_LMT AXI Maximum Read Out Standing Request Limit This value limits the maximum outstanding request on the AXI read interface. Maximum outstanding requests = RD_OSR_LMT+1
15:13	RO	0x0	reserved

Bit	Attr	Reset Value	Description
12	RO	0x0	AXI_AAL Address-Aligned Beats This bit is read-only bit and reflects the AAL bit (register GMAC_BUS_MODE[25]). When this bit set to 1, it performs address-aligned burst transfers on both read and write channels.
11:4	RO	0x0	reserved
3	RW	0x0	BLEN16 AXI Burst Length 16 When this bit is set to 1, or when UNDEF is set to 1, it is allowed to select a burst length of 16.
2	RW	0x0	BLEN8 AXI Burst Length 8 When this bit is set to 1, or when UNDEF is set to 1, it is allowed to select a burst length of 8.
1	RW	0x0	BLEN4 AXI Burst Length 4 When this bit is set to 1, or when UNDEF is set to 1, it is allowed to select a burst length of 4.
0	RO	0x1	UNDEF AXI Undefined Burst Length This bit is read-only bit and indicates the complement (invert) value of FB bit in register GMAC_BUS_MODE[16]. When this bit is set to 1, it is allowed to perform any burst length equal to or below the maximum allowed burst length as programmed in bits[7:1]; When this bit is set to 0, it is allowed to perform only fixed burst lengths as indicated by BLEN256/128/64/32/16/8/4, or a burst length of 1.

GMAC_AXI_STATUS
Address: Operational Base + offset (0x102c)
AXI Status Register

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1	RO	0x0	RD_CH_STA When high, it indicates that AXI Master's read channel is active and transferring data.
0	RO	0x0	WR_CH_STA When high, it indicates that AXI Master's write channel is active and transferring data.

GMAC_CUR_HOST_TX_DESC
Address: Operational Base + offset (0x1048)
Current Host Transmit Descriptor Register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	HTDAP Host Transmit Descriptor Address Pointer Cleared on Reset. Pointer updated by DMA during operation.

GMAC_CUR_HOST_RX_DESC
Address: Operational Base + offset (0x104c)
Current Host Receive Descriptor Register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	HRDAP Host Receive Descriptor Address Pointer Cleared on Reset. Pointer updated by DMA during operation.

GMAC_CUR_HOST_TX_Buf_ADDR
Address: Operational Base + offset (0x1050)
Current Host Transmit Buffer Address Register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	HTBAP Host Transmit Buffer Address Pointer Cleared on Reset. Pointer updated by DMA during operation.

GMAC_CUR_HOST_RX_BUF_ADDR
Address: Operational Base + offset (0x1054)
Current Host Receive Buffer Adderss Register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	HRBAP Host Receive Buffer Address Pointer Cleared on Reset. Pointer updated by DMA during operation.

26.5 Interface Description

Table 26-1 RMII Interface Description

Module pin	Direction	Pad name	IOMUX setting
RMII interface			
mac_clk	I/O	IO_MACclk_I2C5TRACKPADscl_GMACgpio3b3	GRF_GPIO3B_IOMUX[7:6]=2'b10
mac_txen	O	IO_MACtxen_UART1BBsin_GMACgpio3b4	GRF_GPIO3B_IOMUX[9:8]=2'b10
mac_txd1	O	IO_MACtxd1_SPI0NORCODECtxd_GMACgpio3a5	GRF_GPIO3A_IOMUX[11:10]=2'b10
mac_txd0	O	IO_MACtxd0_SPI0NORCODECrxd_GMACgpio3a4	GRF_GPIO3A_IOMUX[9:8]=2'b10
mac_rxdv	I	IO_MACrxdv_GMACgpio3b1	GRF_GPIO3B_IOMUX[3:2]=2'b10
mac_rxer	I	IO_MACrxer_I2C5TRACKPADsda_GMACgpio3b2	GRF_GPIO3B_IOMUX[5:4]=2'b10
mac_rxd1	I	IO_MACrxd1_SPI0NORCODECcsn0_GMACgpio3a7	GRF_GPIO3A_IOMUX[15:14]=2'b10

mac_rxd0	I	IO_MACrxd0_SPI0NORCODECclk_GMACgpio3a6	GRF_GPIO3A_IOMUX[13:12]=2'b10
Management interface			
mac_mdio	I/O	IO_MACmdio_UART1BBsout_GMACgpio3b5	GRF_GPIO3B_IOMUX[11:10]=2'b10
mac_mdc	O	IO_MACmdc_SPI0NORCODECcsn1_GMACgpio3b0	GRF_GPIO3B_IOMUX[1:0]=2'b10

Table 26-2 RGMII Interface Description

Module pin	Direction	Pad name	IOMUX setting
RGMII/RMII interface			
mac_clk	I/O	IO_MACclk_I2C5TRACKPADscl_GMACgpio3b3	GRF_GPIO3B_IOMUX[7:6]=2'b10
mac_txclk	O	IO_MACtxclk_UART3GPSrtsn_GMACgpio3c1	GRF_GPIO3C_IOMUX[3:2]=2'b10
mac_txen	O	IO_MACtxen_UART1BBsin_GMACgpio3b4	GRF_GPIO3B_IOMUX[9:8]=2'b10
mac_txd3	O	IO_MACtxd3_SPI4EXPtxd_TRACEdata13_GMACgpio3a1	GRF_GPIO3A_IOMUX[3:2]=2'b10
mac_txd2	O	IO_MACtxd2_SPI4EXPrxd_TRACEdata12_GMACgpio3a0	GRF_GPIO3A_IOMUX[1:0]=2'b10
mac_txd1	O	IO_MACtxd1_SPI0NORCODECtxd_GMACgpio3a5	GRF_GPIO3A_IOMUX[11:10]=2'b10
mac_txd0	O	IO_MACtxd0_SPI0NORCODECrxd_GMACgpio3a4	GRF_GPIO3A_IOMUX[9:8]=2'b10
mac_rxclk	I	IO_MACrxclk_UART3GPSsin_GMACgpio3b6	GRF_GPIO3B_IOMUX[13:12]=2'b10
mac_rxdv	I	IO_MACrxdv_GMACgpio3b1	GRF_GPIO3B_IOMUX[3:2]=2'b10
mac_rxd3	I	IO_MACrxd3_SPI4EXPcsn0_TRACEdata15_GMACgpio3a3	GRF_GPIO3A_IOMUX[7:6]=2'b10
mac_rxd2	I	IO_MACrxd2_SPI4EXPclk_TRACEdata14_GMACgpio3a2	GRF_GPIO3A_IOMUX[5:4]=2'b10
mac_rxd1	I	IO_MACrxd1_SPI0NORCODECcsn0_GMACgpio3a7	GRF_GPIO3A_IOMUX[15:14]=2'b10
mac_rxd0	I	IO_MACrxd0_SPI0NORCODECclk_GMACgpio3a6	GRF_GPIO3A_IOMUX[13:12]=2'b10
mac_crs	I	IO_MACcrs_UART3GPSsout_CIFclkoutb_GMACgpio3b7	GRF_GPIO3B_IOMUX[15:14]=2'b10
mac_col	I	IO_MACcol_UART3GPSctsn_SPDIFtxb_GMACgpio3c0	GRF_GPIO3C_IOMUX[1:0]=2'b10
Management interface			
mac_mdio	I/O	IO_MACmdio_UART1BBsout_GMACgpio3b5	GRF_GPIO3B_IOMUX[11:10]=2'b10
mac_mdc	O	IO_MACmdc_SPI0NORCODECcsn1_GMACgpio3b0	GRF_GPIO3B_IOMUX[1:0]=2'b10

Notes: I=input, O=output, I/O=input/output, bidirectional

26.6 Application Notes

26.6.1 Descriptors

The DMA in GMAC can communicate with Host driver through descriptor lists and data buffers. The DMA transfers data frames received by the core to the Receive Buffer in the Host memory, and Transmit data frames from the Transmit Buffer in the Host memory. Descriptors that reside in the Host memory act as pointers to these buffers.

There are two descriptor lists; one for reception, and one for transmission. The base address of each list is written into DMA Registers RX_DESC_LIST_ADDR and TX_DESC_LIST_ADDR, respectively. A descriptor list is forward linked (either implicitly or explicitly). The last descriptor may point back to the first entry to create a ring structure.

Explicit chaining of descriptors is accomplished by setting the second address chained in both Receive and Transmit descriptors (RDES1[24] and TDES1[24]). The descriptor lists resides in the Host physical memory address space. Each descriptor can point to a maximum of two buffers. This enables two buffers to be used, physically addressed, rather than contiguous buffers in memory.
A data buffer resides in the Host physical memory space, and consists of an entire frame or part of a frame, but cannot exceed a single frame. Buffers contain only data, buffer status is maintained in the descriptor. Data chaining refers to frames that span multiple data buffers. However, a single descriptor cannot span multiple frames. The DMA will skip to the next frame buffer when end-of-frame is detected. Data chaining can be enabled or disabled
The descriptor ring and chain structure is shown in following figure.

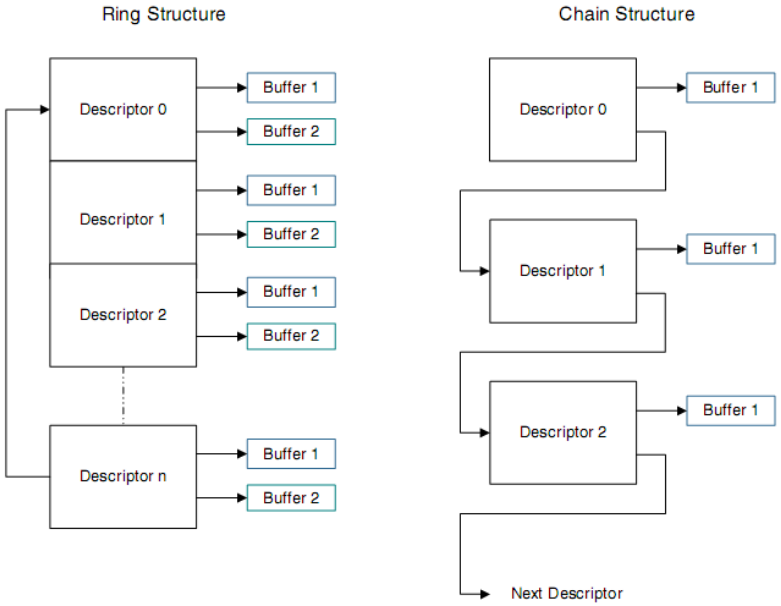


Fig. 26-10 Descriptor Ring and Chain Structure

Each descriptor contains two buffers, two byte-count buffers, and two address pointers, which enable the adapter port to be compatible with various types of memory management schemes. The descriptor addresses must be aligned to the bus width used (Word/Dword/Lword for 32/64/128-bit buses).

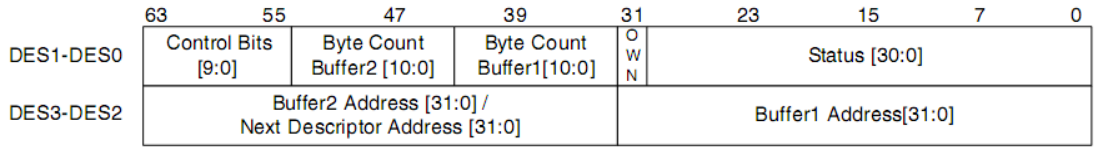


Fig. 26-11 Rx/Tx Descriptors definition

26.6.2 Receive Descriptor

The GMAC Subsystem requires at least two descriptors when receiving a frame. The Receive state machine of the DMAalways attempts to acquire an extra descriptor in anticipation of an incoming frame. (The size of the incoming frame is unknown). Before the RxDMA closes a descriptor, it will attempt to acquire the next descriptor even if no frames are received.
In a single descriptor (receive) system, the subsystem will generate a descriptor error if the receive buffer is unable to accommodate the incoming frame and the next descriptor is not owned by the DMA. Thus, the Host is forced to increase either its descriptor pool or the buffer size. Otherwise, the subsystem starts dropping all incoming frames.

Receive Descriptor 0 (RDES0)

RDES0 contains the received frame status, the frame length, and the descriptor ownership information.

Table 26-3 Receive Descriptor 0

Bit	Description

31	OWN: Own Bit When set, this bit indicates that the descriptor is owned by the DMA of the GMAC Subsystem. When this bit is reset, this bit indicates that the descriptor is owned by the Host. The DMA clears this bit either when it completes the frame reception or when the buffers that are associated with this descriptor are full.
30	AFM: Destination Address Filter Fail When set, this bit indicates a frame that failed in the DA Filter in the GMAC Core.
29:16	FL: Frame Length These bits indicate the byte length of the received frame that was transferred to host memory (including CRC). This field is valid when Last Descriptor (RDES0[8]) is set and either the Descriptor Error (RDES0[14]) or Overflow Error bits are reset. The frame length also includes the two bytes appended to the Ethernet frame when IP checksum calculation (Type 1) is enabled and the received frame is not a MAC control frame. This field is valid when Last Descriptor (RDES0[8]) is set. When the Last Descriptor and Error Summary bits are not set, this field indicates the accumulated number of bytes that have been transferred for the current frame.
15	ES: Error Summary Indicates the logical OR of the following bits: • RDES0[0]: Payload Checksum Error • RDES0[1]: CRC Error • RDES0[3]: Receive Error • RDES0[4]: Watchdog Timeout • RDES0[6]: Late Collision • RDES0[7]: IPC Checksum • RDES0[11]: Overflow Error • RDES0[14]: Descriptor Error This field is valid only when the Last Descriptor (RDES0[8]) is set.
14	DE: Descriptor Error When set, this bit indicates a frame truncation caused by a frame that does not fit within the current descriptor buffers, and that the DMA does not own the Next Descriptor. The frame is truncated. This field is valid only when the Last Descriptor (RDES0[8]) is set
13	SAF: Source Address Filter Fail When set, this bit indicates that the SA field of frame failed the SA Filter in the GMAC Core.
12	LE: Length Error When set, this bit indicates that the actual length of the frame received and that the Length/ Type field does not match. This bit is valid only when the Frame Type (RDES0[5]) bit is reset. Length error status is not valid when CRC error is present.
11	OE: Overflow Error When set, this bit indicates that the received frame was damaged due to buffer overflow.
10	VLAN: VLAN Tag When set, this bit indicates that the frame pointed to by this descriptor is a VLAN frame tagged by the GMAC Core.
9	FS: First Descriptor When set, this bit indicates that this descriptor contains the first buffer of the frame. If the size of the first buffer is 0, the second buffer contains the beginning of the frame. If the size of the second buffer is also 0, the next Descriptor contains the beginning of the frame.
8	LS: Last Descriptor When set, this bit indicates that the buffers pointed to by this descriptor are the last buffers of the frame.
7	IPC Checksum Error/Giant Frame When IP Checksum Engine is enabled, this bit, when set, indicates that the 16-bit

	IPv4 Header checksum calculated by the core did not match the received checksum bytes. The Error Summary bit[15] is NOT set when this bit is set in this mode.
6	LC: Late Collision When set, this bit indicates that a late collision has occurred while receiving the frame in Half-Duplex mode.
5	FT: Frame Type When set, this bit indicates that the Receive Frame is an Ethernet-type frame (the LT field is greater than or equal to 16'h0600). When this bit is reset, it indicates that the received frame is an IEEE802.3 frame. This bit is not valid for Runt frames less than 14 bytes.
4	RWT: Receive Watchdog Timeout When set, this bit indicates that the Receive Watchdog Timer has expired while receiving the current frame and the current frame is truncated after the Watchdog Timeout.
3	RE: Receive Error When set, this bit indicates that the gmii_rxer_i signal is asserted while gmii_rxdv_i is asserted during frame reception. This error also includes carrier extension error in GMII and Half-duplex mode. Error can be of less/no extension, or error (rxd≠ 0f) during extension.
2	DE: Dribble Bit Error When set, this bit indicates that the received frame has a non-integer multiple of bytes (odd nibbles). This bit is valid only in MII Mode.
1	CE: CRC Error When set, this bit indicates that a Cyclic Redundancy Check (CRC) Error occurred on the received frame. This field is valid only when the Last Descriptor (RDES0[8]) is set.
0	Rx MAC Address/Payload Checksum Error When set, this bit indicates that the Rx MAC Address registers value (1 to 15) matched the frame's DA field. When reset, this bit indicates that the Rx MAC Address Register 0 value matched the DA field. If Full Checksum Offload Engine is enabled, this bit, when set, indicates the TCP, UDP, or ICMP checksum the core calculated does not match the received encapsulated TCP, UDP, or ICMP segment's Checksum field. This bit is also set when the received number of payload bytes does not match the value indicated in the Length field of the encapsulated IPv4 or IPv6 datagram in the received Ethernet frame.

Receive Descriptor 1 (RDES1)

RDES1 contains the buffer sizes and other bits that control the descriptor chain/ring.

Table 26-4 Receive Descriptor 1

Bit	Description
31	Disable Interrupt on Completion When set, this bit will prevent the setting of the RI (CSR5[6]) bit of the GMAC_STATUS Register for the received frame that ends in the buffer pointed to by this descriptor. This, in turn, will disable the assertion of the interrupt to Host due to RI for that frame.
30:26	Reserved.
25	RER: Receive End of Ring When set, this bit indicates that the descriptor list reached its final descriptor. The DMA returns to the base address of the list, creating a Descriptor Ring.
24	RCH: Second Address Chained When set, this bit indicates that the second address in the descriptor is the Next Descriptor address rather than the second buffer address. When RDES1[24] is set, RBS2 (RDES1[21-11]) is a "don't care" value. RDES1[25] takes precedence over RDES1[24].

23:22	Reserved.
21:11	RBS2: Receive Buffer 2 Size These bits indicate the second data buffer size in bytes. The buffer size must be a multiple of 8 depending upon the bus widths (64), even if the value of RDES3 (buffer2 address pointer) is not aligned to bus width. In the case where the buffer size is not a multiple of 8, the resulting behavior is undefined. This field is not valid if RDES1[24] is set.
10:0	RBS1: Receive Buffer 1 Size Indicates the first data buffer size in bytes. The buffer size must be a multiple of 8 depending upon the bus widths (64), even if the value of RDES2 (buffer1 address pointer) is not aligned. In the case where the buffer size is not a multiple of 8, the resulting behavior is undefined. If this field is 0, the DMA ignores this buffer and uses Buffer 2 or next descriptor depending on the value of RCH (Bit 24).

Receive Descriptor 2 (RDES2)
RDES2 contains the address pointer to the first data buffer in the descriptor.

Table 26-5 Receive Descriptor 2

Bit	Description
31:0	Buffer 1 Address Pointer These bits indicate the physical address of Buffer 1. There are no limitations on the buffer address alignment except for the following condition: The DMA uses the configured value for its address generation when the RDES2 value is used to store the start of frame. Note that the DMA performs a write operation with the RDES2[2:0] bits as 0 during the transfer of the start of frame but the frame data is shifted as per the actual Buffer address pointer. The DMA ignores RDES2[2:0] (corresponding to bus width of 64) if the address pointer is to a buffer where the middle or last part of the frame is stored.

Receive Descriptor 3 (RDES3)
RDES3 contains the address pointer either to the second data buffer in the descriptor or to the next descriptor.

Table 26-6 Receive Descriptor 3

Bit	Description
31:0	Buffer 2 Address Pointer (Next Descriptor Address) These bits indicate the physical address of Buffer 2 when a descriptor ring structure is used. If the Second Address Chained (RDES1[24]) bit is set, this address contains the pointer to the physical memory where the Next Descriptor is present. If RDES1[24] is set, the buffer (Next Descriptor) address pointer must be bus width-aligned (RDES3[2:0] = 0, corresponding to a bus width of 64. LSBs are ignored internally.) However, when RDES1[24] is reset, there are no limitations on the RDES3 value, except for the following condition: The DMA uses the configured value for its buffer address generation when the RDES3 value is used to store the start of frame. The DMA ignores RDES3[2:0] (corresponding to a bus width of 64) if the address pointer is to a buffer where the middle or last part of the frame is stored.

26.6.3 Transmit Descriptor

The descriptor addresses must be aligned to the bus width used (64). Each descriptor is provided with two buffers, two byte-count buffers, and two address pointers, which enable the adapter port to be compatible with various types of memory-management schemes.

Transmit Descriptor 0 (TDES0)
TDES0 contains the transmitted frame status and the descriptor ownership information.

Table 26-7 Transmit Descriptor 0

Bit	Description
31	OWN: Own Bit When set, this bit indicates that the descriptor is owned by the DMA. When this bit is reset, this bit indicates that the descriptor is owned by the Host. The DMA clears this bit either when it completes the frame transmission or when the buffers allocated in the descriptor are empty. The ownership bit of the First Descriptor of the frame should be set after all subsequent descriptors belonging to the same frame have been set. This avoids a possible race condition between fetching a descriptor and the driver setting an ownership bit.
30:17	Reserved.
16	IHE: IP Header Error When set, this bit indicates that the Checksum Offload engine detected an IP header error and consequently did not modify the transmitted frame for any checksum insertion.
15	ES: Error Summary Indicates the logical OR of the following bits: • TDES0[14]: Jabber Timeout • TDES0[13]: Frame Flush • TDES0[11]: Loss of Carrier • TDES0[10]: No Carrier • TDES0[9]: Late Collision • TDES0[8]: Excessive Collision • TDES0[2]: Excessive Deferral • TDES0[1]: Underflow Error
14	JT: Jabber Timeout When set, this bit indicates the GMAC transmitter has experienced a jabber time-out.
13	FF: Frame Flushed When set, this bit indicates that the DMA/MTL flushed the frame due to a SW flush command given by the CPU.
12	PCE: Payload Checksum Error This bit, when set, indicates that the Checksum Offload engine had a failure and did not insert any checksum into the encapsulated TCP, UDP, or ICMP payload. This failure can be either due to insufficient bytes, as indicated by the IP Header's Payload Length field, or the MTL starting to forward the frame to the MAC transmitter in Store-and-Forward mode without the checksum having been calculated yet. This second error condition only occurs when the Transmit FIFO depth is less than the length of the Ethernet frame being transmitted: to avoid deadlock, the MTL starts forwarding the frame when the FIFO is full, even in Store-and-Forward mode.
11	LC: Loss of Carrier When set, this bit indicates that Loss of Carrier occurred during frame transmission. This is valid only for the frames transmitted without collision and when the GMAC operates in Half-Duplex Mode.
10	NC: No Carrier When set, this bit indicates that the carrier sense signal form the PHY was not asserted during transmission.
9	LC: Late Collision When set, this bit indicates that frame transmission was aborted due to a collision occurring after the collision window (64 byte times including Preamble in RMII Mode and 512 byte times including Preamble and Carrier Extension in RGMII Mode). Not valid if Underflow Error is set.
8	EC: Excessive Collision When set, this bit indicates that the transmission was aborted after 16 successive collisions while attempting to transmit the current frame. If the DR (Disable

Bit	Description
	Retry) bit in the GMAC Configuration Register is set, this bit is set after the first collision and the transmission of the frame is aborted.
7	VF: VLAN Frame When set, this bit indicates that the transmitted frame was a VLAN-type frame.
6:3	CC: Collision Count This 4-bit counter value indicates the number of collisions occurring before the frame was transmitted. The count is not valid when the Excessive Collisions bit (TDES0[8]) is set.
2	ED: Excessive Deferral When set, this bit indicates that the transmission has ended because of excessive deferral of over 24,288 bit times (155,680 bits times in 1000-Mbps mode) if the Deferral Check (DC) bit is set high in the GMAC Control Register.
1	UF: Underflow Error When set, this bit indicates that the GMAC aborted the frame because data arrived late from the Host memory. Underflow Error indicates that the DMA encountered an empty Transmit Buffer while transmitting the frame. The transmission process enters the suspended state and sets both Transmit Underflow (Register GMAC_STATUS[5]) and Transmit Interrupt (Register GMAC_STATUS [0]).
0	DB: Deferred Bit When set, this bit indicates that the GMAC defers before transmission because of the presence of carrier. This bit is valid only in Half-Duplex mode.

Transmit Descriptor 1 (TDES1)

TDES1 contains the buffer sizes and other bits which control the descriptor chain/ring and the frame being transferred.

Table 26-8 Transmit Descriptor 1

Bit	Description
31	IC: Interrupt on Completion When set, this bit sets Transmit Interrupt (Register 5[0]) after the present frame has been transmitted.
30	LS: Last Segment When set, this bit indicates that the buffer contains the last segment of the frame.
29	FS: First Segment When set, this bit indicates that the buffer contains the first segment of a frame.
28:27	CIC: Checksum Insertion Control These bits control the insertion of checksums in Ethernet frames that encapsulate TCP, UDP, or ICMP over IPv4 or IPv6 as described below. • 2'b00: Do nothing. Checksum Engine is bypassed • 2'b01: Insert IPv4 header checksum. Use this value to insert IPv4 header checksum when the frame encapsulates an IPv4 datagram. • 2'b10: Insert TCP/UDP/ICMP checksum. The checksum is calculated over the TCP, UDP, or ICMP segment only and the TCP, UDP, or ICMP pseudo-header checksum is assumed to be present in the corresponding input frame's Checksum field. An IPv4 header checksum is also inserted if the encapsulated datagram conforms to IPv4. • 2'b11: Insert a TCP/UDP/ICMP checksum that is fully calculated in this engine. In other words, the TCP, UDP, or ICMP pseudo-header is included in the checksum calculation, and the input frame's corresponding Checksum field has an all-zero value. An IPv4 Header checksum is also inserted if the encapsulated datagram conforms to IPv4. The Checksum engine detects whether the TCP, UDP, or ICMP segment is encapsulated in IPv4 or IPv6 and processes its data accordingly.

Bit	Description
26	DC: Disable CRC When set, the GMAC does not append the Cyclic Redundancy Check (CRC) to the end of the transmitted frame. This is valid only when the first segment (TDES1[29]).
25	TER: Transmit End of Ring When set, this bit indicates that the descriptor list reached its final descriptor. The returns to the base address of the list, creating a descriptor ring.
24	TCH: Second Address Chained When set, this bit indicates that the second address in the descriptor is the Next Descriptor address rather than the second buffer address. When TDES1[24] is set, TBS2 (TDES1[21–11]) are "don't care" values. TDES1[25] takes precedence over TDES1[24].
23	DP: Disable Padding When set, the GMAC does not automatically add padding to a frame shorter than 64 bytes. When this bit is reset, the DMA automatically adds padding and CRC to a frame shorter than 64 bytes and the CRC field is added despite the state of the DC (TDES1[26]) bit. This is valid only when the first segment (TDES1[29]) is set.
22	Reserved.
21:11	TBS2: Transmit Buffer 2 Size These bits indicate the Second Data Buffer in bytes. This field is not valid if TDES1[24] is set.
10:0	TBS1: Transmit Buffer 1 Size These bits indicate the First Data Buffer byte size. If this field is 0, the DMA ignores this buffer and uses Buffer 2 or next descriptor depending on the value of TCH (Bit 24).

Transmit Descriptor 2 (TDES2)

TDES2 contains the address pointer to the first buffer of the descriptor.

Table 26-9 Transmit Descriptor 2

Bit	Description
31:0	Buffer 1 Address Pointer These bits indicate the physical address of Buffer 1. There is no limitation on the buffer address alignment.

Transmit Descriptor 3 (TDES3)

TDES3 contains the address pointer either to the second buffer of the descriptor or the next descriptor.

Table 26-10 Transmit Descriptor 3

Bit	Description
31:0	Buffer 2 Address Pointer (Next Descriptor Address) Indicates the physical address of Buffer 2 when a descriptor ring structure is used. If the Second Address Chained (TDES1[24]) bit is set, this address contains the pointer to the physical memory where the Next Descriptor is present. The buffer address pointer must be aligned to the bus width only when TDES1[24] is set. (LSBs are ignored internally.)

26.6.4 Programming Guide

DMA Initialization – Descriptors

The following operations must be performed to initialize the DMA.

1. Provide a software reset. This will reset all of the GMAC internal registers and logic. (GMAC_OP_MODE[0]).
2. Wait for the completion of the reset process (poll GMAC_OP_MODE[0], which is only cleared after the reset operation is completed).
3. Program the following fields to initialize the Bus Mode Register by setting values in

register GMAC_BUS_MODE
  a. Mixed Burst and AAL
  b. Fixed burst or undefined burst
  c. Burst length values and burst mode values.
  d. Descriptor Length (only valid if Ring Mode is used)
  e. Tx and Rx DMA Arbitration scheme
4. Program the AXI Interface options in the register GMAC_BUS_MODE
  a. If fixed burst-length is enabled, then select the maximum burst-length possible on the AXI bus (Bits[7:1])
5. A proper descriptor chain for transmit and receive must be created. It should also ensure that the receive descriptors are owned by DMA (bit 31 of descriptor should be set). When OSF mode is used, at least two descriptors are required.
6. Software should create three or more different transmit or receive descriptors in the chain before reusing any of the descriptors.
7. Initialize receive and transmit descriptor list address with the base address of transmit and receive descriptor (register GMAC_RX_DESC_LIST_ADDR and GMAC_TX_DESC_LIST_ADDR).
8. Program the following fields to initialize the mode of operation by setting values in register GMAC_OP_MODE
  a. Receive and Transmit Store And Forward
  b. Receive and Transmit Threshold Control (RTC and TTC)
  c. Hardware Flow Control enable
  d. Flow Control Activation and De-activation thresholds for MTL Receive and Transmit FIFO (RFA and RFD)
  e. Error Frame and undersized good frame forwarding enable
  f. OSF Mode
9. Clear the interrupt requests, by writing to those bits of the status register (interrupt bits only) which are set. For example, by writing 1 into bit 16 - normal interrupt summary will clear this bit (register GMAC_STATUS).
10. Enable the interrupts by programming the interrupt enable register GMAC_INT_ENA.
11. Start the Receive and Transmit DMA by setting SR (bit 1) and ST (bit 13) of the control register GMAC_OP_MODE.

MAC Initialization

The following MAC Initialization operations can be performed after the DMA initialization sequence. If the MAC Initialization is done before the DMA is set-up, then enable the MAC receiver (last step below) only after the DMA is active. Otherwise, received frames will fill the RxFIFO and overflow.
1. Program the register GMAC_GMII_ADDR for controlling the management cycles for external PHY, for example, Physical Layer Address PA (bits 15-11). Also set bit 0 (GMII Busy) for writing into PHY and reading from PHY.
2. Read the 16-bit data of (GMAC_GMII_DATA) from the PHY for link up, speed of operation, and mode of operation, by specifying the appropriate address value in registerGMAC_GMII_ADDR (bits 15-11).
3. Provide the MAC address registers (GMAC_MAC_ADDR0_HI and GMAC_MAC_ADDR0_LO).
4. If Hash filtering is enabled in your configuration, program the Hash filter register (GMAC_HASH_TAB_HI and GMAC_HASH_TAB_LO).
5. Program the following fields to set the appropriate filters for the incoming frames in register GMAC_MAC_FRM_FILT
  a. Receive All
  b. Promiscuous mode
  c. Hash or Perfect Filter
  d. Unicast, Multicast, broad cast and control frames filter settings etc.
6. Program the following fields for proper flow control in register GMAC_FLOW_CTRL.
  a. Pause time and other pause frame control bits
  b. Receive and Transmit Flow control bits

c. Flow Control Busy/Backpressure Activate

7. Program the Interrupt Mask register bits, as required, and if applicable, for your configuration.
8. Program the appropriate fields in register GMAC_MAC_CONF for example, Inter-frame gap while transmission, jabber disable, etc. Based on the Auto-negotiation you can set the Duplex mode (bit 11), port select (bit 15), etc.
9. Set the bits Transmit enable (TE bit-3) and Receive Enable (RE bit-2) in register GMAC_MAC_CONF.

Normal Receive and Transmit Operation

For normal operation, the following steps can be followed.

- For normal transmit and receive interrupts, read the interrupt status. Then poll the descriptors, reading the status of the descriptor owned by the Host (either transmit or receive).
- On completion of the above step, set appropriate values for the descriptors, ensuring that transmit and receive descriptors are owned by the DMA to resume the transmission and reception of data.
- If the descriptors were not owned by the DMA (or no descriptor is available), the DMA will go into SUSPEND state. The transmission or reception can be resumed by freeing the descriptors and issuing a poll demand by writing 0 into the Tx/Rx poll demand register (GMAC_TX_POLL_DEMAND and GMAC_RX_POLL_DEMAND).
- The values of the current host transmitter or receiver descriptor address pointer can be read for the debug process (GMAC_CUR_HOST_TX_DESC and GMAC_CUR_HOST_RX_DESC).
- The values of the current host transmit buffer address pointer and receive buffer address pointer can be read for the debug process (GMAC_CUR_HOST_TX_Buf_ADDR and GMAC_CUR_HOST_RX_BUF_ADDR).

Stop and Start Operation

When the transmission is required to be paused for some time then the following steps can be followed.

1. Disable the Transmit DMA (if applicable), by clearing ST (bit 13) of the control register GMAC_OP_MODE.
2. Wait for any previous frame transmissions to complete. This can be checked by reading the appropriate bits of MAC Debug register.
3. Disable the MAC transmitter and MAC receiver by clearing the bits Transmit enable (TE bit-3) and Receive Enable (RE bit-2) in register GMAC_MAC_CONF.
4. Disable the Receive DMA (if applicable), after making sure the data in the RX FIFO is transferred to the system memory (by reading the register GMAC_DEBUG).
5. Make sure both the TX FIFO and RX FIFO are empty.
6. To re-start the operation, start the DMAs first, before enabling the MAC Transmitter and Receiver.

26.6.5 Clock Architecture

In RMII mode, reference clock and TX/RX clock can be from CRU or external OSC as following figure.

The mux select rmii_speed is GRF_SOC_CON1[11].

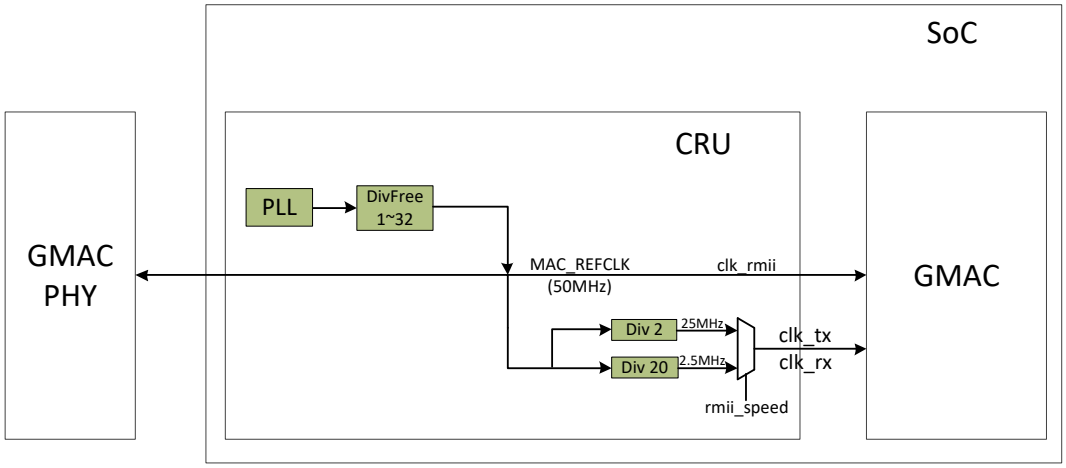


Fig. 26-12 RMII clock architecture when clock source from CRU

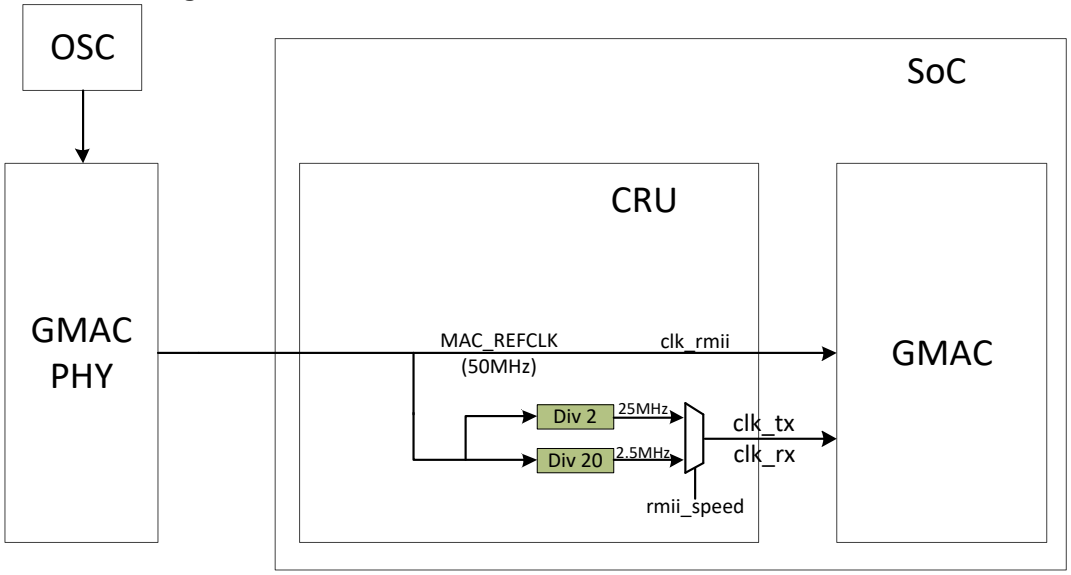


Fig. 26-13 RMII clock architecture when clock source from external OSC

In RGMII mode, clock architecture only supports that TX clock source is from CRU as following figure.

In order to dynamically adjust the timing between TX/RX clocks with data, deleyline is integrated in TX and RX clock path. Register GRF_SOC_CON3[15:14] can enable the deleylines, and GRF_SOC_CON3[13:0] is used to determine the delay length. There are 100 deley elements in each delayline.

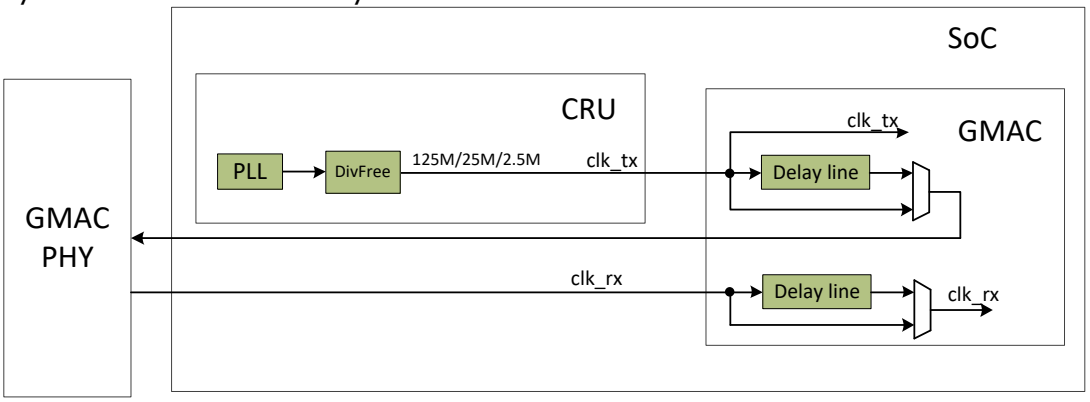


Fig. 26-14 RGMII clock architecture when clock source from CRU

26.6.6 Remote Wake-Up Frame Filter Register

The register wkupfmfilter_reg, address (028H), loads the Wake-up Frame Filter register. To load values in a Wake-up Frame Filter register, the entire register (wkupfmfilter_reg) must be written. The wkupfmfilter_reg register is loaded by sequentially loading the eight register values in address (028) for wkupfmfilter_reg0, wkupfmfilter_reg1, ...,

wkupfmfilter_reg7, respectively. Wkupfmfilter_reg is read in the same way.
The internal counter to access the appropriate wkupfmfilter_reg is incremented when lane3 (or lane 0 in big-endian) is accessed by the CPU. This should be kept in mind if you are accessing these registers in byte or half-word mode.

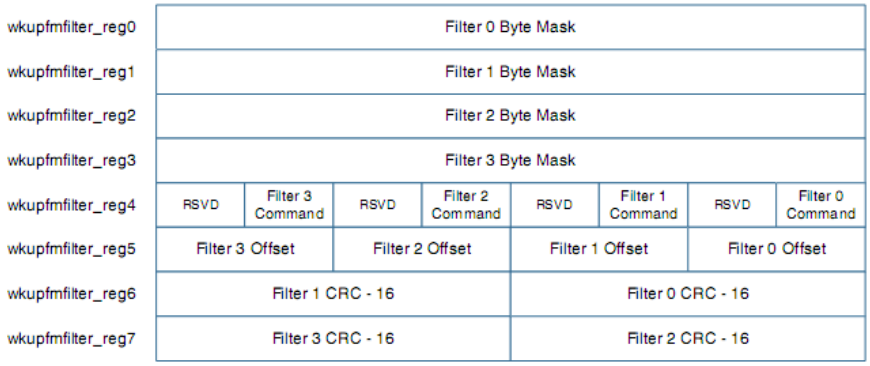


Fig. 26-15 Wake-Up Frame Filter Register

Filter i Byte Mask

This register defines which bytes of the frame are examined by filter i (0, 1, 2, and 3) in order to determine whether or not the frame is a wake-up frame. The MSB (thirty-first bit) must be zero. Bit j [30:0] is the Byte Mask. If bit j (byte number) of the Byte Mask is set, then Filter i Offset + j of the incoming frame is processed by the CRC block; otherwise Filter i Offset + j is ignored.

Filter i Command

This 4-bit command controls the filter i operation. Bit 3 specifies the address type, defining the pattern's destination address type. When the bit is set, the pattern applies to only multicast frames; when the bit is reset, the pattern applies only to unicast frame. Bit 2 and Bit 1 are reserved. Bit 0 is the enable for filter i; if Bit 0 is not set, filter i is disabled.

Filter i Offset

This register defines the offset (within the frame) from which the frames are examined by filter i. This 8-bit pattern-offset is the offset for the filter i first byte to examined. The minimum allowed is 12, which refers to the 13th byte of the frame (offset value 0 refers to the first byte of the frame).

Filter i CRC-16

This register contains the CRC_16 value calculated from the pattern, as well as the byte mask programmed to the wake-up filter register block.

26.6.7 System Consideration During Power-Down

GMAC neither gates nor stops clocks when Power-Down mode is enabled. Power saving by clock gating must be done outside the core by the CRU. The receive data path must be clocked with clk_rx_i during Power-Down mode, because it is involved in magic packet/wake-on-LAN frame detection. However, the transmit path and the APB path clocks can be gated off during Power-Down mode.
The PMT interrupt is asserted when a valid wake-up frame is received. This interrupt is generated in the clk_rx domain.
The recommended power-down and wake-up sequence is as follows.
1. Disable the Transmit DMA (if applicable) and wait for any previous frame transmissions to complete. These transmissions can be detected when Transmit Interrupt (TI - Register GMAC_STATUS[0]) is received.
2. Disable the MAC transmitter and MAC receiver by clearing the appropriate bits in the MAC Configuration register.
3. Wait until the Receive DMA empties all the frames from the Rx FIFO (a software timer may be required).
4. Enable Power-Down mode by appropriately configuring the PMT registers.
5. Enable the MAC Receiver and enter Power-Down mode.
6. Gate the APB and transmit clock inputs to the core (and other relevant clocks in the system) to reduce power and enter Sleep mode.
7. On receiving a valid wake-up frame, the GMAC asserts the PMT interrupt signal and exits

Power-Down mode.
8. On receiving the interrupt, the system must enable the APB and transmit clock inputs to the core.
9. Read the register GMAC_PMT_CTRL_STA to clear the interrupt, then enable the other modules in the system and resume normal operation.

26.6.8 GRF Register Summary

GRF Register	Register Description
GRF_MAC_CON1[6:4]	PHY interface select 3'b001: RGMII 3'b100: RMII All others: Reserved
GRF_MAC_CON1[3]	GMAC transmit flow control When set high, instructs the GMAC to transmit PAUSE Control frames in Full-duplex mode. In Half-duplex mode, the GMAC enables the Back-pressure function until this signal is made low again
GRF_MAC_CON1[2]	GMACspeed 1'b1: 100-Mbps 1'b0: 10-Mbps
GRF_MAC_CON1[7]	RMII clock selection 1'b1: 25MHz 1'b0: 2.5MHz
GRF_MAC_CON1[9:8]	RGMII clock selection 2'b00: 125MHz 2'b11: 25MHz 2'b10: 2.5MHz
GRF_MAC_CON1[10]	RMII mode selection 1'b1: RMII mode 1’b0: Reserved
GRF_MAC_CON0[6:0]	RGMII TX clock delayline value
GRF_MAC_CON0[13:7]	RGMII RX clock delayline value
GRF_MAC_CON1[0]	RGMII TX clock delayline enable 1'b1: enable 1'b0: disable
GRF_MAC_CON1[1]	RGMII RX clock delayline enable 1'b1: enable 1'b0: disable

Chapter 27 SARADC

27.1 Overview

The ADC is a 6-channel signal-ended 10-bit Successive Approximation Register (SAR) A/D Converter. It uses the supply and ground as it reference which avoid use of any external reference. It converts the analog input signal into 10-bit binary digital codes at maximum conversion rate of 1MSPS with 13MHz A/D converter clock.

27.2 Block Diagram

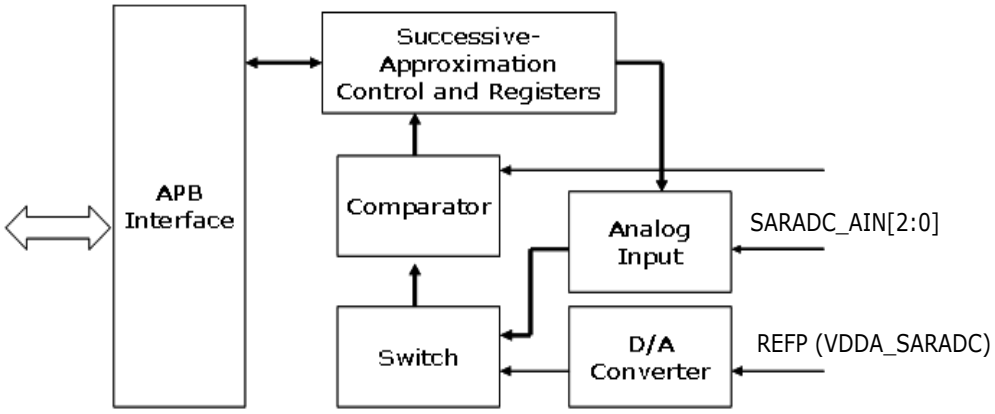


Fig. 27-1 RK3399Pro SAR-ADC block diagram

Successive-Approximate Register and Control Logic Block

This block is exploited to realize binary search algorithm, storing the intermediate result and generate control signal for analog block.

Comparator Block

This block compares the analog input SARADC_AIN[2:0] with the voltage generated from D/A Converter, and output the comparison result to SAR and Control Logic Block for binary search. Three level amplifiers are employed in this comparator to provide enough gain.

27.3 Function Description

27.3.1 APB Interface

In RK3399Pro, SAR-ADC works at single-sample operation mode.

This mode is useful to sample an analog input when there is a gap between two samples to be converted. In this mode START is asserted only on the rising edge of CLKIN where conversion is needed. At the end of every conversion EOC signal is made high and valid output data is available at the rising edge of EOC. The detailed timing diagram will be shown in the following.

27.4 Register description

27.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
SARADC_DATA	0x0000	W	0x00000000	This register contains the data after A/D Conversion.
SARADC_STAS	0x0004	W	0x00000000	The status register of A/D Converter.
SARADC_CTRL	0x0008	W	0x00000000	The control register of A/D Converter.
SARADC_DLY_PU_SOC	0x000c	W	0x00000000	delay between power up and start command

*Notes:Size:**B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access*

27.4.2 Detail Register Description

SARADC_DATA

Address: Operational Base + offset (0x0000)

This register contains the data after A/D Conversion.

Bit	Attr	Reset Value	Description
31:10	RO	0x0	reserved
9:0	RO	0x000	adc_data A/D value of the last conversion (DOUT[9:0]).

SARADC_STAS

Address: Operational Base + offset (0x0004)

The status register of A/D Converter.

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RO	0x0	adc_status ADC status (EOC) 0: ADC stop 1: Conversion in progress

SARADC_CTRL

Address: Operational Base + offset (0x0008)

The control register of A/D Converter.

Bit	Attr	Reset Value	Description
31:7	RO	0x0	reserved
6	RW	0x0	int_status Interrupt status. This bit will be set to 1 when end-of-conversion. Set 0 to clear the interrupt.
5	RW	0x0	int_en Interrupt enable. 0: Disable 1: Enable
4	RO	0x0	reserved
3	RW	0x0	adc_power_ctrl ADC power down control bit 0: ADC power down; 1: ADC power up and reset. start signal will be asserted (DLY_PU_SOC + 2) sclk clock period later after power up

Bit	Attr	Reset Value	Description
2:0	RW	0x0	adc_input_src_sel ADC input source selection(CH_SEL[2:0]). 000 : Input source 0 (SARADC_AIN[0]) 001 : Input source 1 (SARADC_AIN[1]) 010 : Input source 2 (SARADC_AIN[2]) 011 : Input source 3 (SARADC_AIN[3]) 100 : Input source 4 (SARADC_AIN[4]) 101 : Input source 5 (SARADC_AIN[5]) Others : Reserved

SARADC_DLY_PU_SOC
Address: Operational Base + offset (0x000c)
delay between power up and start command

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5:0	RW	0x00	DLY_PU_SOC delay between power up and start command The start signal will be asserted (DLY_PU_SOC + 2) sclk clock period later after power up

27.5 Timing Diagram

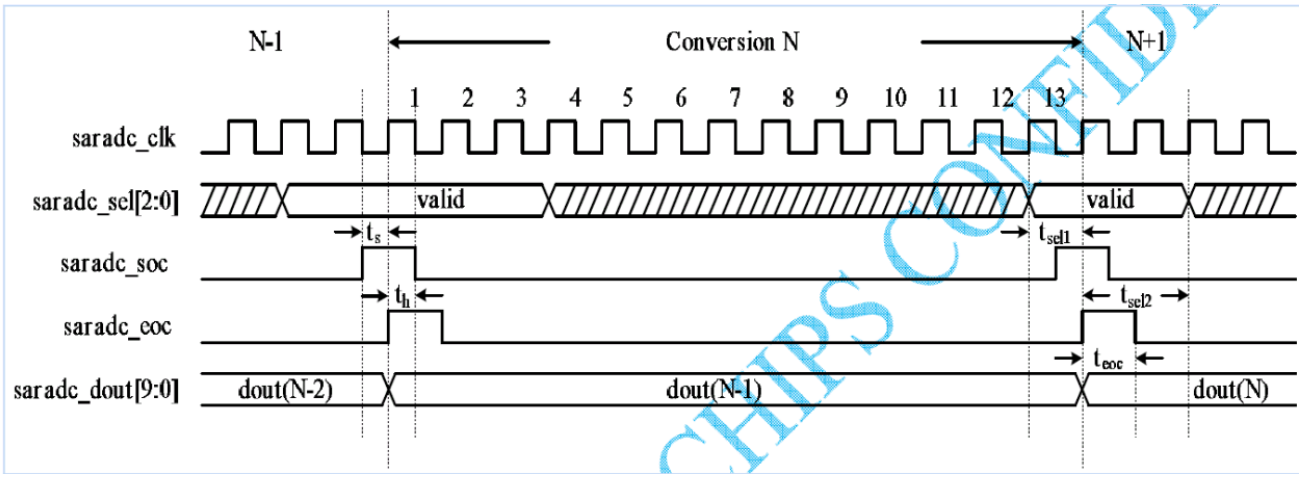


Fig. 27-2 SAR-ADC timing diagram in single-sample conversion mode

The following table has shows the detailed value for timing parameters in the above diagram.

Parameter	Symbol	Condition	Min	Typ	Max	Unit
Operating Condition						
Analog Supply	AVDD		1.62	1.8	1.98	V
Digital Supply	VDD		0.81	0.90	0.99	V
Junction Temperature	T_J		-40		125	℃
SARADC Performance						
Resolution				10		bit
Effective Number of Bit	ENOB			9		bit
Differential Nonlinearity	DNL		-1		+1	LSB
Integral Nonlinearity	INL		-2		+2	LSB
Input Voltage Range	V_{IN}		0		1	AVDD
Input Capacitance	C_{IN}			10		pF
Sampling Rate	f_S				1	MS/s
Spurious Free Dynamic Range	SFDR	f_S=1MS/s f_{OUT}=1.17KHz		61		dB
Signal to Noise and Harmonic Ratio	SNDR			56		dB
Timing Characteristic						
Clock Frequency	f_{CLK}				13	MHz
Clock Period	t_{CLK}		75			ns
Clock Duty Cycle			45		55	%
Conversion Time			13			t_{CLK}
Setup Time of soc signal	t_s			0.5		t_{CLK}
Hold Time of soc signal	t_h			0.5		t_{CLK}
Time Interval between Transition of sel[2:0] and Rising Edge of 1st clock	t_{sel1}		1			t_{CLK}
Time Interval between Transition of sel[2:0] and Rising Edge of 1st clock	t_{sel2}		2			t_{CLK}
High Level Time of eoc signal	t_{eoc}		1			t_{CLK}
Power Consumption						
Analog Supply Current	I_{AVDD}	f_S=1MS/s		450		uA
		Power Down		1		uA
Digital Supply Current	I_{VDD}	f_S=1MS/s		50		uA
		Power Down		1		uA

Fig. 27-3 RK3399Pro SAR-ADC timing parameters list

27.6 Application Notes

Steps of adc conversion:

- Write SARADC_CTRL[3] as 0 to power down adc converter.
- Write SARADC_CTRL[2:0] as n to select adc channel(n).
- Write SARADC_CTRL[5] as 1 to enable adc interrupt.
- Write SARADC_CTRL[3] as 1 to power up adc converter.
- Wait for adc interrupt or poll SARADC_STAS register to assert whether the conversion is completed
- Read the conversion result from SARADC_DATA[9:0]
- Note: The A/D converter was designed to operate at maximum 1MHZ.

Chapter 28 Graphics Process Unit (GPU)

28.1 Overview

GPU provides a complete graphics acceleration platform based on open standards, With support for 2D graphics, 3D graphics, and GPGPU computing
The GPU supports the following graphics standards:

- OpenGL ES 3.0
- OpenGL ES 2.0
- OpenGL ES 1.1
- OpenCL 1.2
- OpenCL 1.1
- OpenCL 1.0
- DirectX 11.1
- DirectX 9

The GPU consists of:

- 4 Shader Cores
- 1 256KB Level2 Cache Memory Subsystem
- 1 Memory Management Unit
- 1 Job Manager
- 1 Hierarchical Tiler
- 1 Core Group

The GPU contains a 32-bit APB bus and a 128-bit AXI bus. CPU configures GPU through APB bus, GPU read and write data through AXI bus.
The GPU includes the following features:

- A rich API feature set with high-performance support for both shader-based and fixed-function graphics APIs.
- Anti-aliasing capabilities.
- An effective core for General Purpose computing on GPU (GPGPU) applications.
- High memory bandwidth and low power consumption for 3D graphics content.
- Scalability for products from smart phones to high-end mobile computing.
- Performance leading 3D graphics.
- Image quality using double-precision FP64.
- Standard bus interfaces.
- Latency tolerance.
- Compressed texture formats.
- Frame buffer compression.
- 10-bit and 16-bit YUV input and output formats.

28.2 Block Diagram

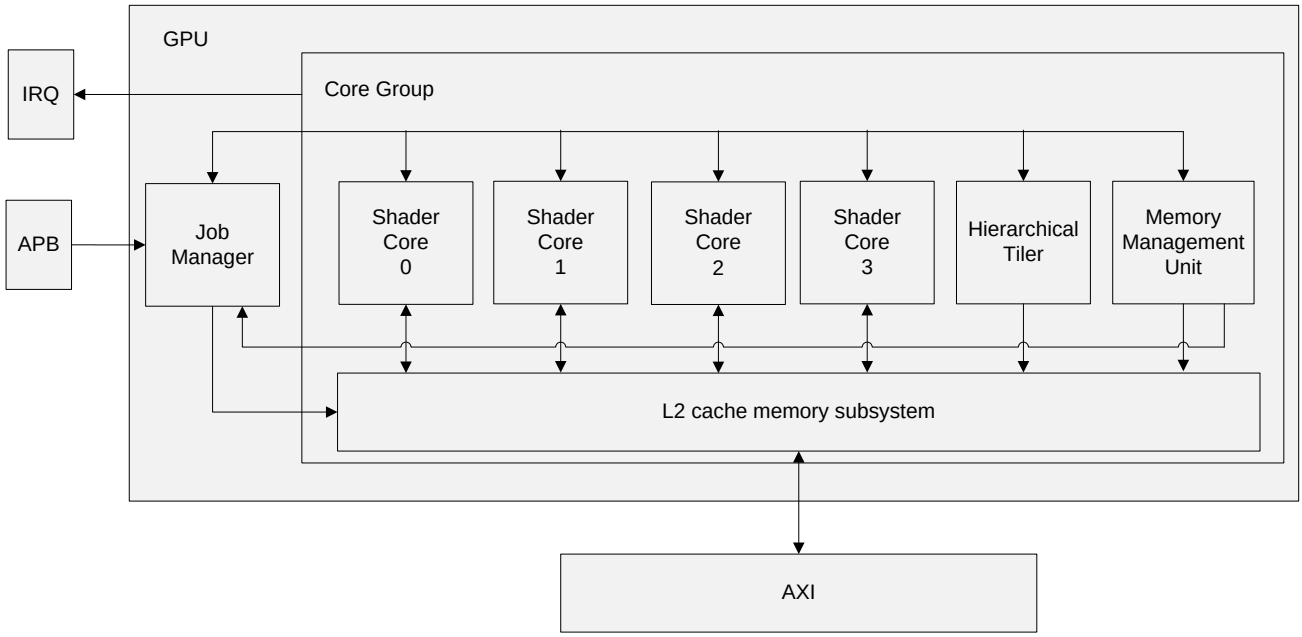


Fig. 28-1 GPU block diagram

• When the application software uses the graphics device driver to send a graphics job to the GPU. The job manager receives the job, interprets it, and then sends a series of graphics tasks to the functional units within the GPU.
• The shader cores are the main processing engines of the GPU. The shader cores carry out all the rendering and computation operations.
• The tiler creates lists of all the objects in a scene, so that the shader cores process the objects efficiently.
• The memory management unit performs virtual address to physical address translation of external AXI interface accesses.
• The level 2 cache memory subsystem provides additional caching for all internal master blocks.
• The interrupts are generated for job handling, memory management, and events not tied to a specific shader core.

28.3 Function Description

When the application software, running on the application processor, schedules a job to be sent to the GPU, the following actions take place:
1. The application processor graphics device driver takes graphics data from a user application, packages it, and sends it as a series of graphics jobs to the job manager.
2. The graphics device driver places the description and data to be used for the graphics jobs into the shared memory in the form of job descriptors stored in defined data structures.
3. The graphics device driver also sets up the high-level configuration of the GPU using configuration registers that communicate with the job manager using the register interface.
4. The job manager reads the descriptions of the graphics jobs from shared memory.
5. The job manager converts the graphics jobs into multiple small GPU tasks that are then distributed to the GPU modules where they are processed.
6. When the tasks complete, the results of the graphics jobs are placed back into shared memory.
7. The application processor is notified that the jobs are complete.
When a graphics job starts, it proceeds to completion without having to refer back to the application processor for more information. When the job is complete it can, if necessary, start the next graphics job without further interaction with the application processor.
There are three top level interrupts raised by the GPU:
• GPU interrupts Exceptions that are not associated with specific jobs.

• Job interrupts Signals the completion or failure of a job running on the GPU.
• MMU interrupts Exceptions caused by memory management.
The application processor interprets the interrupts generated by the GPU . The software queries the states of the registers in the job manager to determine what must be done to handle the interrupt. The job manager is not required to wait for the result of the interrupt, it can be starting on the next set of jobs to be executed.

28.4 Timing Diagram

The GPU only has a clock input, which is called aclk_gpu. aclk_gpu is generated from the CRU module as shows below

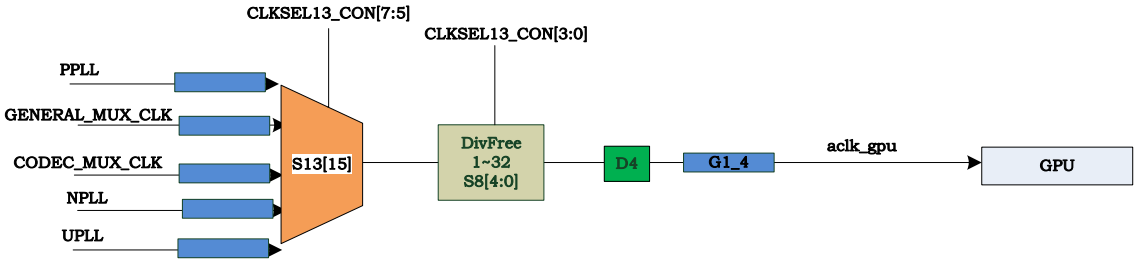


Fig. 28-2 ACLK_GPU generate block diagram

We can configure CPLL, GPLL, NPLL, PPLL,UPLL and CRU register CRU_CLKSEL13_CON to control the gpu_aclk frequency.

28.5 Register Description

The GPU base addressis 0xff9a_0000.

Chapter 29 Neural Process Unit (NPU)

29.1 Overview

NPU is the process unit which is dedicated to neural network. It is designed to accelerate the neural network arithmetic in field of AI (artificial intelligence) such as machine vision and natural language processing. The variety of applications for AI is expanding, and currently provides functionality in a variety of areas, including face tracking as well as gesture and body tracking, image classification, video surveillance, automatic speech recognition (ASR) and advanced driver assistance systems (ADAS).

NPU supports the following features:

- Host interface
  - 32bit AHB interface used for configuration only support single
  - 128bit AXI interface used to fetch data from memory
- Neural Network
  - Support integer 8, integer 16, float 16 convolution operation
  - 1920 MAD (multiply-add units) per cycle (int 8)
  - 192 MAD per cycle (int 16)
  - 64 MAD per cycle (float 16)
  - Support Liner, MIMO, Fully Connected, Fully Convolution
  - Unlimited network size (bound by system resource)
  - Inference Engine : TensorFlow backend, OpenCl, OpenVX, Android NN backend
  - Support network sparse coefficient decompression
  - Support Max, average pooling
  - Max pooling support 2x2, 3x3, stride <= min(input width, input height)
  - Local average pooling size <= 11x11
  - Support unpooling
  - Support batch normalize, l2 normalize, l2 normalize scale, local response normalize
  - Support region proposal
  - Support permute, reshape, concat, depth to space, space to depth, flatten, reorg, squeeze and split
  - Support priorbox layer
  - Support Non-max Suppression
  - Support ROI pooling
  - Convolution size NXN , N <=11*stride, stride <= min(input width, input height)
  - Support dilate convolution, N <=11*stride, stride <= min(input width, input height), dilation <1024
  - Support de-convolution, N <=11*stride, stride <= min(input width, input height)
  - Support Elementwise addition, div, floor, max, mul, scale, sub
  - Support elu, leaky_relu, prelu, relu, relu1, relu6, sigmod, softmax, tanh
  - Support LSTM, RNN
  - Support channel shuffle
  - Support dequantize, dropout.
  - Include embedded lookup table
  - Support hashtable lookup
  - Support lsh projection
  - Support svdf
  - Support reserve

29.2 Block Diagram

NPU comprises with:

- HIF: Host Interface
- PM : Power Management
- NN Engine : Neural Network Engine
- VPU : Vector Procession Unit

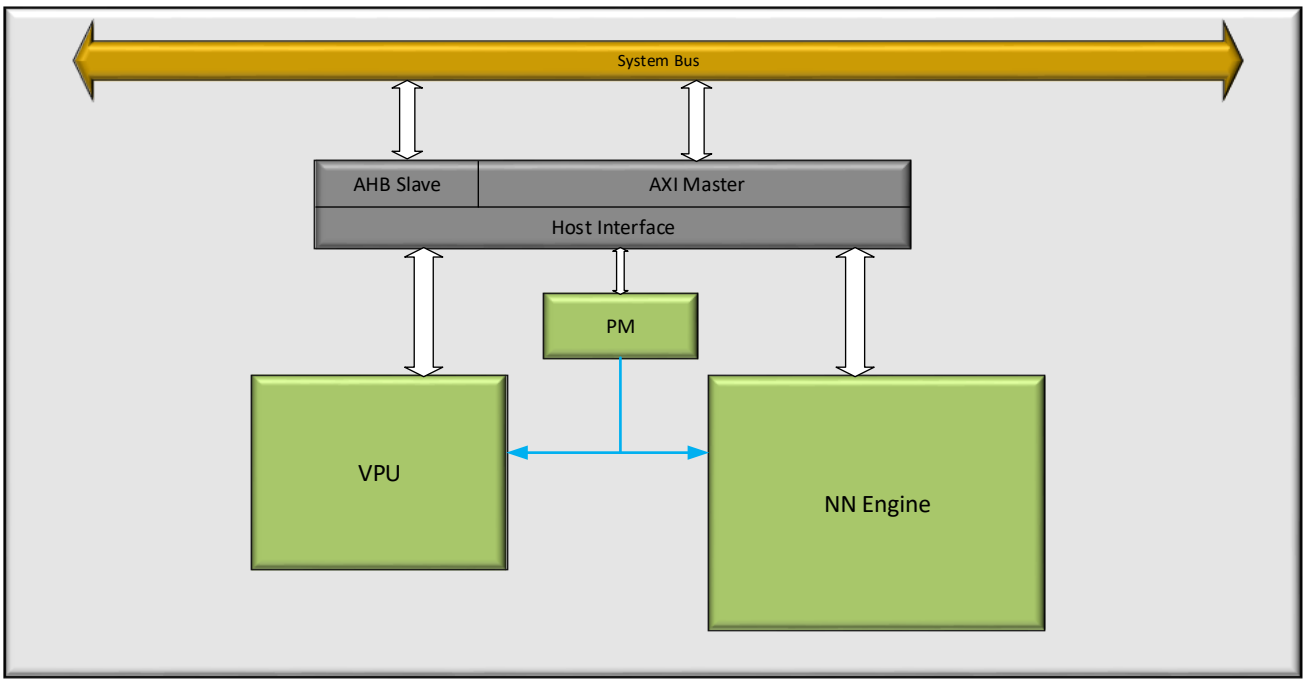


Fig. 29-1 NPU Block Diagram

29.3 Function Description

29.3.1 Host Interface

Allows the NPU to communicate with external memory and the CPU through the AXI or AHB bus. In this block data crosses clock domain boundaries. It includes a Front End (FE) to insert high level primitives and commands into the calculation pipeline. There is an AHB slave and AXI master in Host inter face unit. The AXI master interface is used to fetch data from memory that is attached to the Soc AXI interconnect. The AHB slave interface is used to access the graphics registers for configuration, debug and test.

29.3.2 Power Management

Power management in NPU is used to provide top level controls for clock, reset and power management. AHB clock, AXI clock, clock for VPU and NN Engine are all asynchronization, and clock pins are connected to Power Management. The reset of all models are also controlled by Power Management. Global clock gating can be controlled by Power Management to reduce power.

29.3.3 Neural Network Engine

As the unit name, NN Engine is the main process unit for Neural Network arithmetic. This unit Provides parallel convolution MAC for recognition functions and int8, int16 and fp16 are supported. Active functions and pooling such as leaky_relu, relu, relu1, relu6, sigmod, tanh are also processed in NN Engine. So NN Engine is mainly serve for convolution neural network and fully connected network.

29.3.4 Vector Processing Unit

Vector Processing Unit can be the supplement for NN Engine. The programmable SIMD processor unit is included which perform as a Compute Unit for OpenCL. VPU provides advanced image processing functions. For example, in one cycle, VPU can perform one MUL/ADD instruction or a dot product of two 16-component values. Most element wise operations and matrix operations are processed in VPU.

29.4 Register Description

29.4.1 Internal Address Mapping

Slave address can be divided into different length for different usage, which is shown as follows.

Table 29-1 NPU Address Mapping

Base Address[12:8]	Device	Address Length	Offset Address Range
0xffbc0000	NPU	512K BYTE	0x00000 ~ 0x7ffff