Chapter 3 Clock and reset unit

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 four PLLs
- Support only one crystal
- Flexible selection of clock source
- Supports the respectivegating of all clocks
- Supports the respective software reset of all modules

3.2 Block Diagram

The CRU comprises with:

- Four PLLs
- Register configuration unit
- Clock generate unit
- Reset generate unit

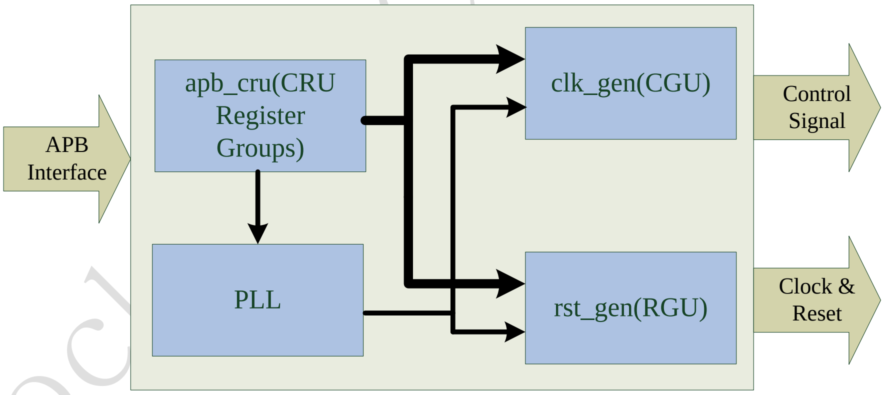

Fig. 3-1 CRU Architecture

3.3 System Clock Solution

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

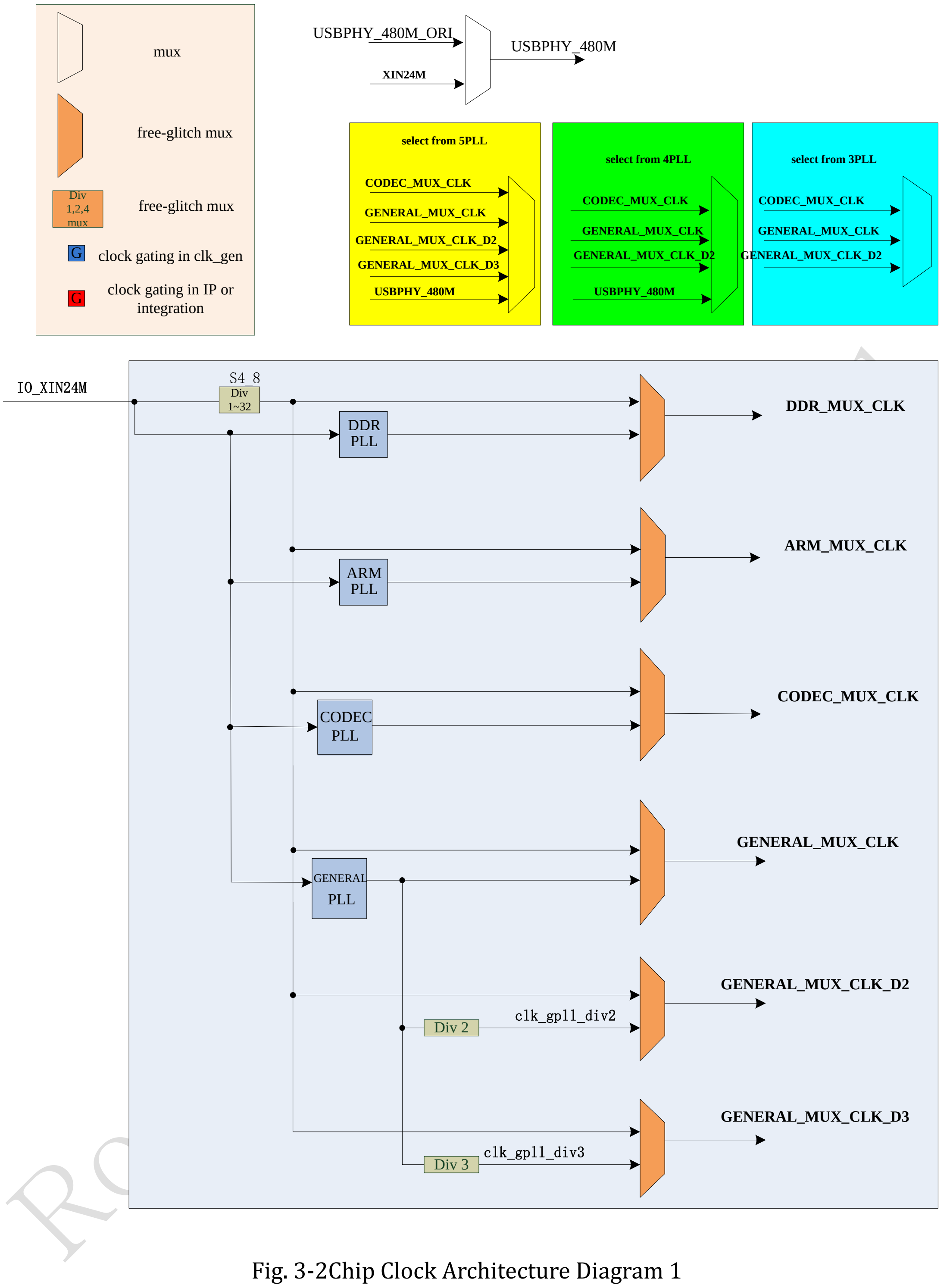

Fig. 3-2Chip Clock Architecture Diagram 1

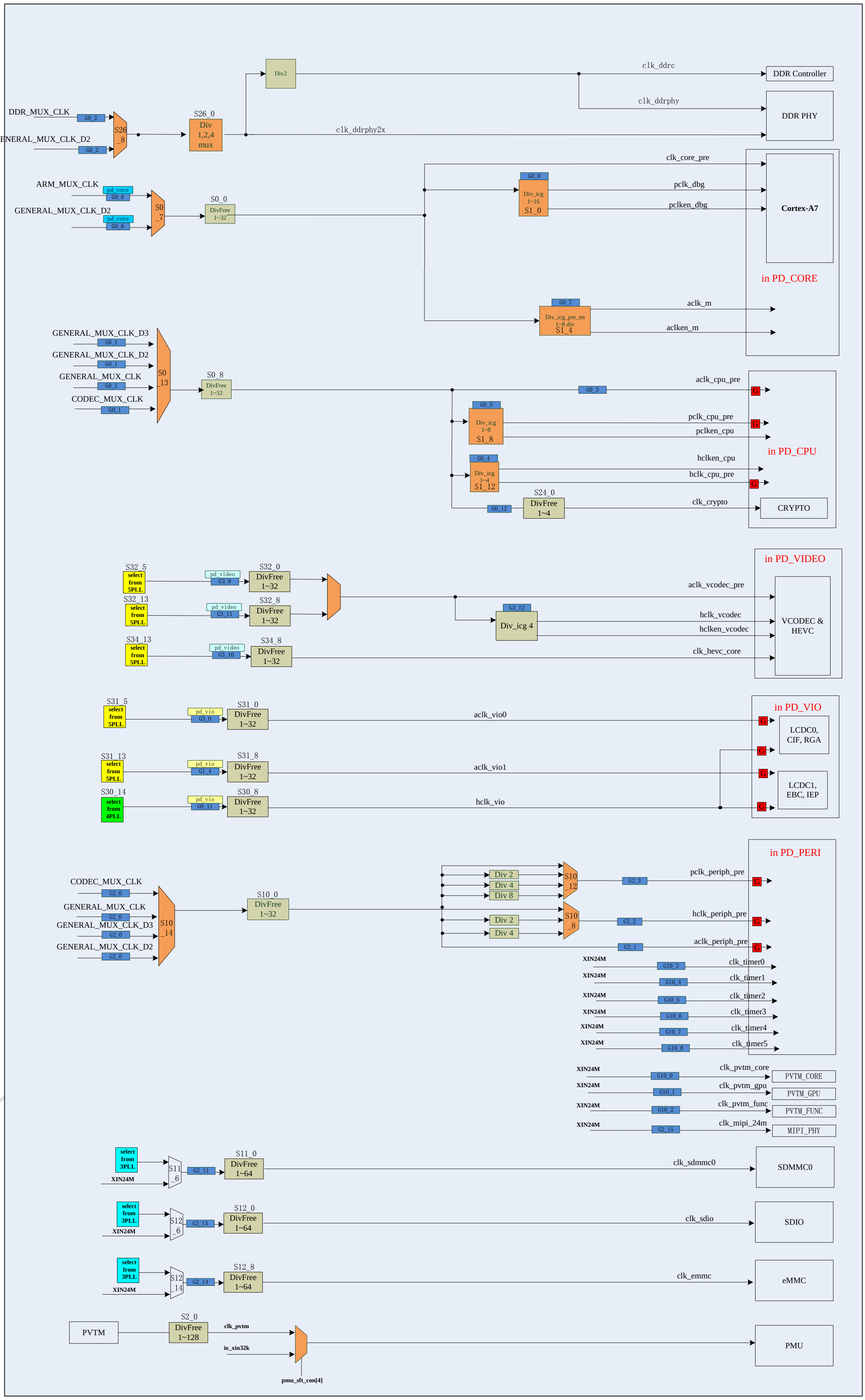

Fig. 3-3Chip Clock Architecture Diagram 2

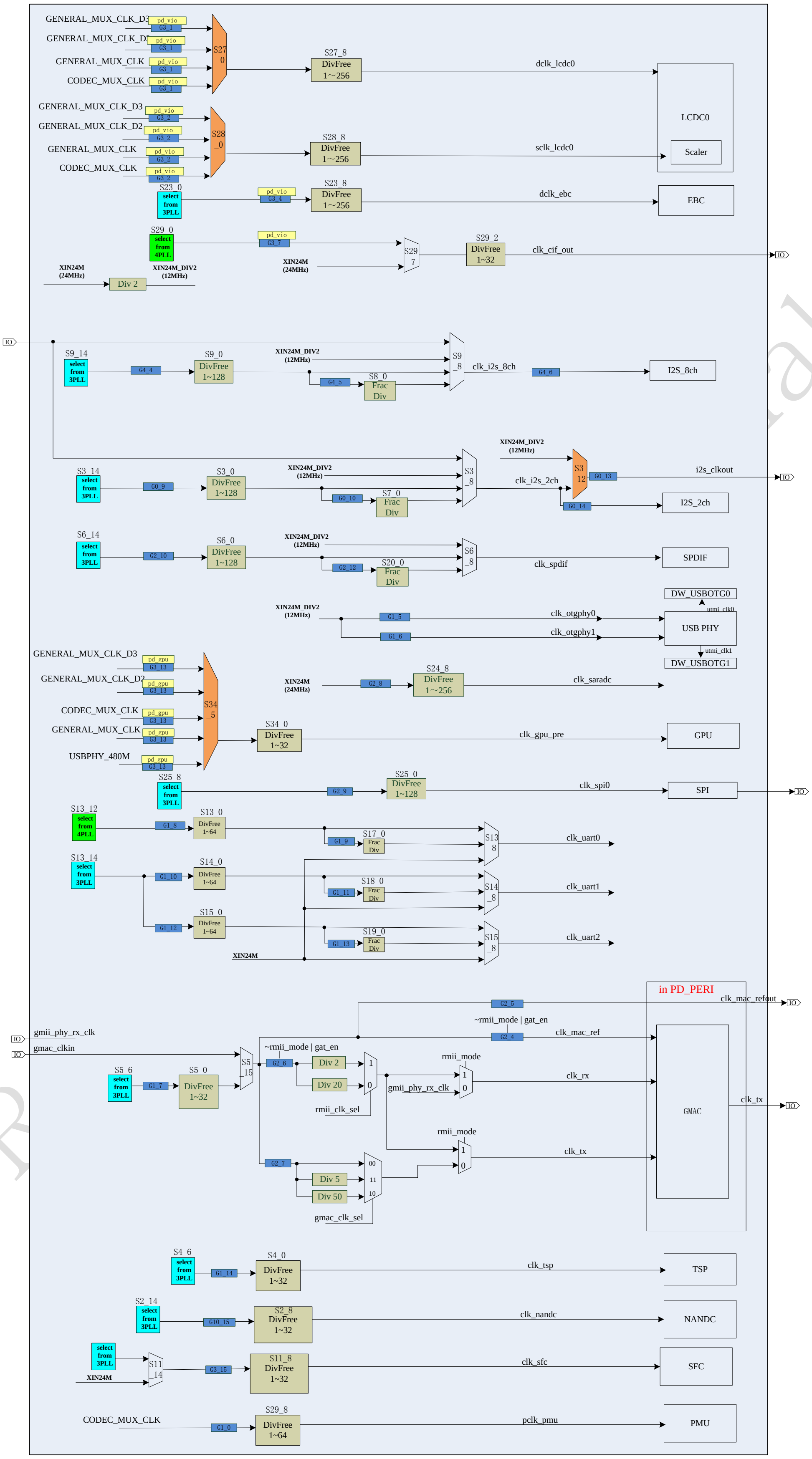

Fig. 3-4Chip Clock Architecture Diagram 3

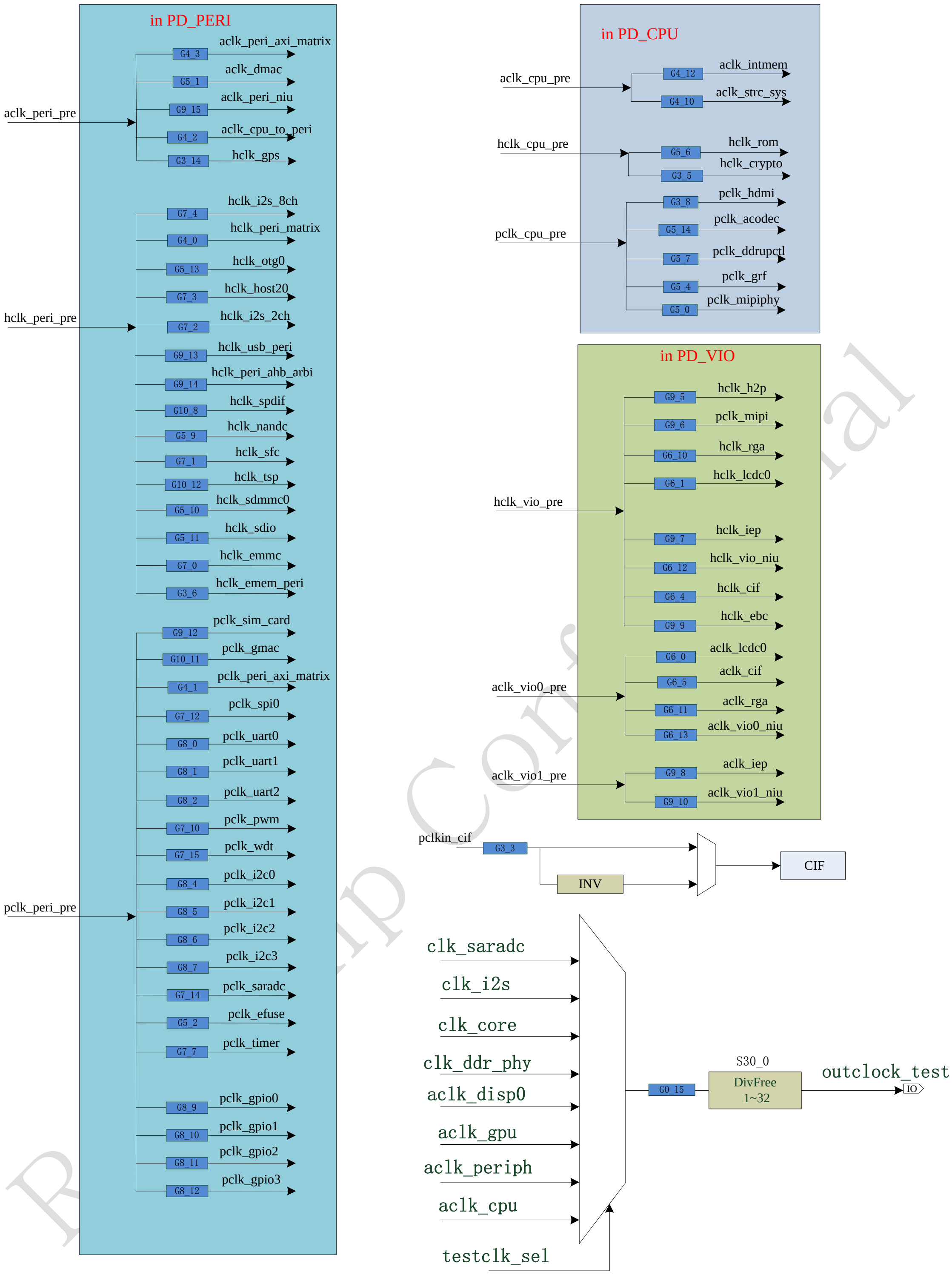

Fig. 3-6 Chip Clock Architecture Diagram 4

Description about input clock

The source of input clock in upper diagrams is listed as following Table.

Table 3-1 Input clock description in clock architecture diagram

Input Clock	Source	IO Name
xin24m	External crystal oscillator (24MHz)	XIN24M

3.4 System Reset Solution

The following diagrams show reset architecture in This device.

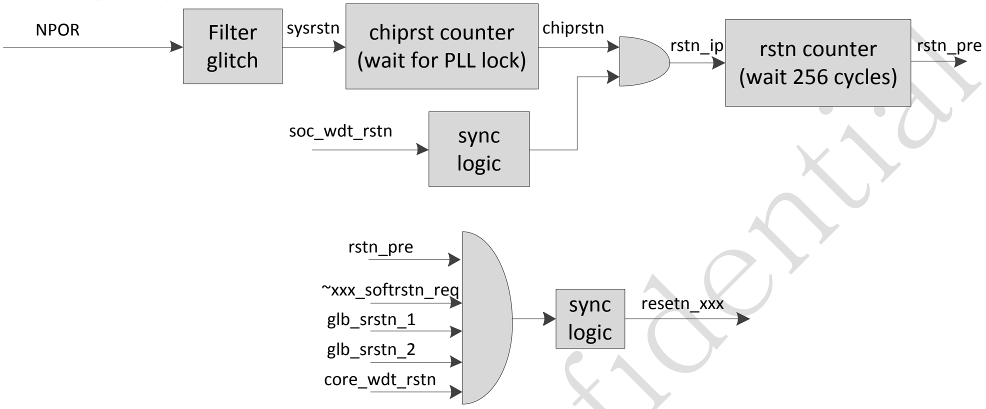

Fig. 3-5Reset Architecture Diagram

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 A9 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 A9 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 four PLLs in the chip: ARM PLL, DDR PLL, CODEC PLL and GENERAL PLL, and it supports only onecrystal oscillator: 24MHz. Each PLL can only receive 24MHz oscillator.

Four PLLs all can be set to slow mode or deep slow mode, directly output selectable 24MHz or 32.768kHz. 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 multy 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 2.4GHz PLL for all four PLLs.The 2.4GHz PLL is a general purpose, high-performance PLL-based clock generator. ThePLL is a multi-function, general purposefrequency synthesizer. Ultra-wide input and output ranges along with best-in-class jitter performanceallow 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 outputclocks into a low power, low area, widely programmable design, Silicon Creations can greatly simplifyan SoC by enabling a single macro to be used for all clocking applications in the system.

2.4GHz PLL supports the following features:

- Input Frequency Range:1MHz to 800MHz(Integer Mode) and
- 10MHz to 800MHz (Fractional Mode)
- Output Frequency Range: 12MHz to 2.4GHz
- 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 (2.5V) allows for excellent supply rejection in noisy SoC applications.
- Lock Detect Signal indicates when frequency lock has been achieved.

3.6.2 Block diagram

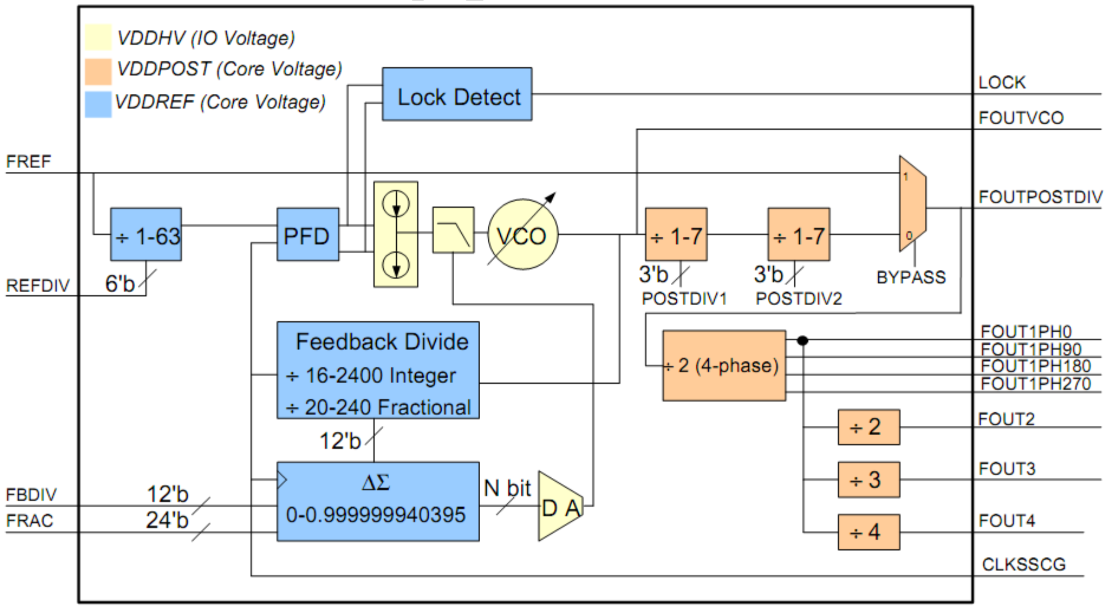

Fig. 3-6PLL Block Diagram

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 / 2^{24})
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

Name	Offset	Size	Reset Value	Description
CRU_APLL_CON0	0x0000	W	0x000010af	ARM PLL control register0
CRU_APLL_CON1	0x0004	W	0x00001046	ARM PLL control register1
CRU_APLL_CON2	0x0008	W	0x00000001	ARM PLL control register2
CRU_DPLL_CON0	0x0010	W	0x00001064	DDR PLL control register0
CRU_DPLL_CON1	0x0014	W	0x00001043	DDR PLL control register1
CRU_DPLL_CON2	0x0018	W	0x00000001	DDR PLL control register2
CRU_CPLL_CON0	0x0020	W	0x0000107d	Codec PLL control register0
CRU_CPLL_CON1	0x0024	W	0x00001046	Codec PLL control register1
CRU_CPLL_CON2	0x0028	W	0x00000001	Codec PLL control register2
CRU_GPLL_CON0	0x0030	W	0x00004063	General PLL control register0
CRU_GPLL_CON1	0x0034	W	0x00001042	General PLL control register1
CRU_GPLL_CON2	0x0038	W	0x00000001	General PLL control register2
CRU_MODE_CON	0x0040	W	0x00000000	System work mode control register

Name	Offset	Size	Reset Value	Description
CRU_CLKSEL0_CON	0x0044	W	0x00000100	Internal clock select and divide register0
CRU_CLKSEL1_CON	0x0048	W	0x00003113	Internal clock select and divide register1
CRU_CLKSEL2_CON	0x004c	W	0x00000707	Internal clock select and divide register2
CRU_CLKSEL3_CON	0x0050	W	0x0000001f	Internal clock select and divide register3
CRU_CLKSEL4_CON	0x0054	W	0x00000003	Internal clock select and divide register4
CRU_CLKSEL5_CON	0x0058	W	0x00000003	Internal clock select and divide register5
CRU_CLKSEL6_CON	0x005c	W	0x0000021f	Internal clock select and divide register6
CRU_CLKSEL7_CON	0x0060	W	0x0bb8ea60	Internal clock select and divide register7
CRU_CLKSEL8_CON	0x0064	W	0x0bb8ea60	Internal clock select and divide register8
CRU_CLKSEL9_CON	0x0068	W	0x0000001f	Internal clock select and divide register9
CRU_CLKSEL10_CON	0x006c	W	0x0000a100	Internal clock select and divide register10
CRU_CLKSEL11_CON	0x0070	W	0x00000717	Internal clock select and divide register11
CRU_CLKSEL12_CON	0x0074	W	0x00001717	Internal clock select and divide register12
CRU_CLKSEL13_CON	0x0078	W	0x0000121f	Internal clock select and divide register13
CRU_CLKSEL14_CON	0x007c	W	0x0000021f	Internal clock select and divide register14
CRU_CLKSEL15_CON	0x0080	W	0x0000021f	Internal clock select and divide register15
CRU_CLKSEL17_CON	0x0088	W	0x0bb8ea60	Internal clock select and divide register17
CRU_CLKSEL18_CON	0x008c	W	0x0bb8ea60	Internal clock select and divide register18
CRU_CLKSEL19_CON	0x0090	W	0x0bb8ea60	Internal clock select and divide register19
CRU_CLKSEL20_CON	0x0094	W	0x0bb8ea60	Internal clock select and divide register20
CRU_CLKSEL23_CON	0x00a0	W	0x00000100	Internal clock select and divide register23
CRU_CLKSEL24_CON	0x00a4	W	0x00001701	Internal clock select and divide register24
CRU_CLKSEL25_CON	0x00a8	W	0x0000011f	Internal clock select and divide register25
CRU_CLKSEL26_CON	0x00ac	W	0x00000000	Internal clock select and divide register26
CRU_CLKSEL27_CON	0x00b0	W	0x00000301	Internal clock select and divide register27

Name	Offset	Size	Reset Value	Description
CRU_CLKSEL28_CON	0x00b4	W	0x00000301	Internal clock select and divide register28
CRU_CLKSEL29_CON	0x00b8	W	0x00000524	Internal clock select and divide register29
CRU_CLKSEL30_CON	0x00bc	W	0x00000300	Internal clock select and divide register30
CRU_CLKSEL31_CON	0x00c0	W	0x00004040	Internal clock select and divide register31
CRU_CLKSEL32_CON	0x00c4	W	0x00000101	Internal clock select and divide register32
CRU_CLKSEL34_CON	0x00cc	W	0x00004141	Internal clock select and divide register34
CRU_CLKGATE0_CON	0x00d0	W	0x00000000	Internal clock gating control register0
CRU_CLKGATE1_CON	0x00d4	W	0x00000000	Internal clock gating control register1
CRU_CLKGATE2_CON	0x00d8	W	0x00000000	Internal clock gating control register2
CRU_CLKGATE3_CON	0x00dc	W	0x00000000	Internal clock gating control register3
CRU_CLKGATE4_CON	0x00e0	W	0x00000000	Internal clock gating control register0
CRU_CLKGATE5_CON	0x00e4	W	0x00000000	Internal clock gating control register5
CRU_CLKGATE6_CON	0x00e8	W	0x00000000	Internal clock gating control register6
CRU_CLKGATE7_CON	0x00ec	W	0x00000000	Internal clock gating control register7
CRU_CLKGATE8_CON	0x00f0	W	0x00000000	Internal clock gating control register8
CRU_CLKGATE9_CON	0x00f4	W	0x00000000	Internal clock gating control register9
CRU_CLKGATE10_CON	0x00f8	W	0x00000000	Internal clock gating control register10
CRU_GLB_SRST_FST_VALUE	0x0100	W	0x00000000	The first global software reset config value
CRU_GLB_SRST_SND_VALUE	0x0104	W	0x00000000	The second global software reset config value
CRU_SOFTRST0_CON	0x0110	W	0x00000000	Internal software reset control register0
CRU_SOFTRST1_CON	0x0114	W	0x00000000	Internal software reset control register1
CRU_SOFTRST2_CON	0x0118	W	0x00000000	Internal software reset control register2
CRU_SOFTRST3_CON	0x011c	W	0x00000000	Internal software reset control register3
CRU_SOFTRST4_CON	0x0120	W	0x00000000	Internal software reset control register4
CRU_SOFTRST5_CON	0x0124	W	0x00000000	Internal software reset control register5

Name	Offset	Size	Reset Value	Description
CRU_SOFTRST6_CON	0x0128	W	0x00000000	Internal software reset control register6
CRU_SOFTRST7_CON	0x012c	W	0x00000000	Internal software reset control register7
CRU_SOFTRST8_CON	0x0130	W	0x00000000	Internal software reset control register8
CRU_MISC_CON	0x0134	W	0x00008000	SCU control register
CRU_GLB_CNT_TH	0x0140	W	0x3a980064	global reset wait counter threshold
CRU_GLB_RST_ST	0x0150	W	0x00000000	global reset status
CRU_SDMMC_CON0	0x01c0	W	0x00000004	sdmmc control0
CRU_SDMMC_CON1	0x01c4	W	0x00000000	sdmmc control1
CRU_SDIO_CON0	0x01c8	W	0x00000004	sdio0 control0
CRU_SDIO_CON1	0x01cc	W	0x00000000	sdio0 control1
CRU_EMMC_CON0	0x01d8	W	0x00000004	emmc control0
CRU_EMMC_CON1	0x01dc	W	0x00000000	emmc control1
CRU_PLL_PRG_EN	0x01f0	W	0x00000000	PLL program enable

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

3.7.2 Detail Register Description

CRU_APLL_CON0
Address: Operational Base + offset (0x0000)
ARM PLL control register0

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	bit_write_mask bit_write_mask control corresponding (bit_write_mask - 16) configuration bit 0: mask 1: unmask
15	RW	0x0	bp pll bypass
14:12	RW	0x1	postdiv1 PLL factor postdiv1
11:0	RW	0x0af	fbdiv PLL factor fbdiv

CRU_APLL_CON1
Address: Operational Base + offset (0x0004)
ARM PLL control register1

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	bit_write_mask bit_write_mask control corresponding (bit_write_mask - 16) configuration bit 0: mask 1: unmask

Bit	Attr	Reset Value	Description
15	RW	0x0	rstmode PLL Reset select 0 : internal reset 1 : software reset
14	RW	0x0	rst PLL Software Reset 0 : normal 1 : reset
13	RW	0x0	pd PLL software power down，active high
12	RW	0x1	dsmpd when 1, PLL work at interger mode when 0, PLL work at frac mode
11	RO	0x0	reserved
10	RW	0x0	lock PLL lock status
9	RO	0x0	reserved
8:6	RW	0x1	postdiv2 PLL factor postdiv2
5:0	RW	0x06	refdiv PLL factor refdiv

CRU_APLL_CON2
Address: Operational Base + offset (0x0008)
ARM PLL control register2

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27	RW	0x0	fout4phasepd 4 phase clock power down，active high
26	RW	0x0	foutvcopd buffered VCO clock power down，active high
25	RW	0x0	foutpostdivpd post divide power down，active high
24	RW	0x0	dacpd PLL cancellation DAC power down, active high
23:0	RW	0x000001	frac PLL factor frac

CRU_DPLL_CON0
Address: Operational Base + offset (0x0010)
DDR PLL control register0

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	bit_write_mask bit_write_mask control corresponding (bit_write_mask - 16) configuration bit 0: mask 1: unmask
15	RW	0x0	bp pll bypass

Bit	Attr	Reset Value	Description
14:12	RW	0x1	postdiv1 PLL factor postdiv1
11:0	RW	0x064	fbdiv PLL factor fbdiv

CRU_DPLL_CON1
Address: Operational Base + offset (0x0014)
DDR PLL control register1

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	bit_write_mask bit_write_mask control corresponding (bit_write_mask - 16) configuration bit 0: mask 1: unmask
15	RW	0x0	rstmode PLL Reset select 0 : internal reset 1 : software reset
14	RW	0x0	rst PLL Software Reset 0 : normal 1 : reset
13	RW	0x0	pd PLL software power down，active high
12	RW	0x1	dsmpd when 1, PLL work at interger mode when 0, PLL work at frac mode
11	RO	0x0	reserved
10	RW	0x0	lock PLL lock status
9	RO	0x0	reserved
8:6	RW	0x1	postdiv2 PLL factor postdiv2
5:0	RW	0x03	refdiv PLL factor refdiv

CRU_DPLL_CON2
Address: Operational Base + offset (0x0018)
DDR PLL control register2

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27	RW	0x0	fout4phasepd 4 phase clock power down，active high
26	RW	0x0	foutvcopd buffered VCO clock power down，active high
25	RW	0x0	foutpostdivpd post divide power down，active high
24	RW	0x0	dacpd PLL cancellation DAC power down, active high

Bit	Attr	Reset Value	Description
23:0	RW	0x000001	frac PLL factor frac

CRU_CPLL_CON0
Address: Operational Base + offset (0x0020)
Codec PLL control register0

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	bit_write_mask bit_write_mask control corresponding (bit_write_mask - 16) configuration bit 0: mask 1: unmask
15	RW	0x0	bp pll bypass
14:12	RW	0x1	postdiv1 PLL factor postdiv1
11:0	RW	0x07d	fbdiv PLL factor fbdiv

CRU_CPLL_CON1
Address: Operational Base + offset (0x0024)
Codec PLL control register1

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	bit_write_mask bit_write_mask control corresponding (bit_write_mask - 16) configuration bit 0: mask 1: unmask
15	RW	0x0	rstmode PLL Reset select 0 : internal reset 1 : software reset
14	RW	0x0	rst PLL Software Reset 0 : normal 1 : reset
13	RW	0x0	pd PLL software power down，active high
12	RW	0x1	dsmpd when 1, PLL work at interger mode when 0, PLL work at frac mode
11	RO	0x0	reserved
10	RW	0x0	lock PLL lock status
9	RO	0x0	reserved
8:6	RW	0x1	postdiv2 PLL factor postdiv2
5:0	RW	0x06	refdiv PLL factor refdiv

CRU_CPLL_CON2
Address: Operational Base + offset (0x0028)
Codec PLL control register2

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27	RW	0x0	fout4phasepd 4 phase clock power down，active high
26	RW	0x0	foutvcopd buffered VCO clock power down，active high
25	RW	0x0	foutpostdivpd post divide power down，active high
24	RW	0x0	dacpd PLL cancellation DAC power down, active high
23:0	RW	0x000001	frac PLL factor frac

CRU_GPLL_CON0
Address: Operational Base + offset (0x0030)
General PLL control register0

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	bit_write_mask bit_write_mask control corresponding (bit_write_mask - 16) configuration bit 0: mask 1: unmask
15	RW	0x0	bp pll bypass
14:12	RW	0x4	postdiv1 PLL factor postdiv1
11:0	RW	0x063	fbdiv PLL factor fbdiv

CRU_GPLL_CON1
Address: Operational Base + offset (0x0034)
General PLL control register1

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	bit_write_mask bit_write_mask control corresponding (bit_write_mask - 16) configuration bit 0: mask 1: unmask
15	RW	0x0	rstmode PLL Reset select 0 : internal reset 1 : software reset
14	RW	0x0	rst PLL Software Reset 0 : normal 1 : reset

Bit	Attr	Reset Value	Description
13	RW	0x0	pd PLL software power down，active high
12	RW	0x1	dsmpd when 1, PLL work at interger mode when 0, PLL work at frac mode
11	RO	0x0	reserved
10	RW	0x0	lock PLL lock status
9	RO	0x0	reserved
8:6	RW	0x1	postdiv2 PLL factor postdiv2
5:0	RW	0x02	refdiv PLL factor refdiv

CRU_GPLL_CON2
Address: Operational Base + offset (0x0038)
General PLL control register2

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27	RW	0x0	fout4phasepd 4 phase clock power down，active high
26	RW	0x0	foutvcopd buffered VCO clock power down，active high
25	RW	0x0	foutpostdivpd post divide power down，active high
24	RW	0x0	dacpd PLL cancellation DAC power down, active high
23:0	RW	0x000001	frac PLL factor frac

CRU_MODE_CON
Address: Operational Base + offset (0x0040)
System work mode control register

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. 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	gpll_work_mode GENERAL PLL work mode select 1'b0: Slow mode, clock from external 24MHz OSC (default) 1'b1: Normal mode, clock from PLL output
11:9	RO	0x0	reserved

Bit	Attr	Reset Value	Description
8	RW	0x0	cpll_work_mode CODEC PLL work mode select 1'b0: Slow mode, clock from external 24MHz OSC (default) 1'b1: Normal mode, clock from PLL output
7:5	RO	0x0	reserved
4	RW	0x0	dpll_work_mode DDR PLL work mode select 1'b0: Slow mode, clock from external 24MHz OSC (default) 1'b1: Normal mode, clock from PLL output
3:1	RO	0x0	reserved
0	RW	0x0	apll_work_mode ARM PLL work mode select 1'b0: Slow mode, clock from external 24MHz OSC (default) 1'b1: Normal mode, clock from PLL output

CRU_CLKSEL0_CON
Address: Operational Base + offset (0x0044)
Internal clock select and divide register0

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. 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	0x0	cpu_clk_pll_sel pd_cpu aclk_cpu pll source selection 2'b00: select CODEC PLL 2'b01: select GENERAL PLL 2'b10: select GENERAL PLL DIV2 2'b11: select GENERAL PLL DIV3
12:8	RW	0x01	aclk_cpu_div_con aclk_cpu clock divider frequency aclk_cpu=cpu_clk_src/(aclk_cpu_div_con+1)
7	RW	0x0	core_clk_pll_sel core clock pll source selection 1'b0: select ARM PLL 1'b1: select GENERAL PLL div2
6:5	RO	0x0	reserved
4:0	RW	0x00	a7_core_div_con Control A7 core clock divider frequency clk_core=core_clk_src/(a9_core_div_con+1)

CRU_CLKSEL1_CON
Address: Operational Base + offset (0x0048)
Internal clock select and divide register1

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. 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	cpu_pclk_div_con Control cpu subsystem APB clock divider frequency pclk_cpu=cpu_aclk_src/(cpu_pclk_div_con+1)
11:10	RO	0x0	reserved
9:8	RW	0x1	cpu_hclk_div_con Control cpu subsystem AHB clock divider frequency hclk_cpu=cpu_aclk_src/(cpu_hclk_div_con+1)
7	RO	0x0	reserved
6:4	RW	0x1	core_aclk_div_con Control A9 core axi clock divider frequency aclk_core=core_clk_src/(core_aclk_div_con+1)
3:0	RW	0x3	pclk_dbg_div_con pclk_dbg div control pclk_debg=pclk_dbg_src/(pclk_dbg_div_con+1)

CRU_CLKSEL2_CON
Address: Operational Base + offset (0x004c)
Internal clock select and divide register2

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:14	RW	0x0	nandc_clk_pll_sel nandc pll source selection 2'b00: select codec pll clock 2'b01: select general pll clock 2'b10: select general pll div2 clock
13	RO	0x0	reserved

Bit	Attr	Reset Value	Description
12:8	RW	0x07	nandc_div_con Control NANDC clock divider frequency clk_nandc=nandc_clk_src/(nandc_div_con+1)
7	RO	0x0	reserved
6:0	RW	0x07	pvtm_div_con func pvtm clock divider frequency clk_pvtm=pvtm_clk_src/(pvtm_div_con+1)

CRU_CLKSEL3_CON

Address: Operational Base + offset (0x0050)

Internal clock select and divide register3

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:14	RW	0x0	i2s_2ch_pll_sel Control I2S_2ch PLL source selection 2'b00: select codec pll clock 2'b01: select general pll clock 2'b10: select general pll div2 clock
13	RO	0x0	reserved
12	RW	0x0	i2s_2ch_clkout_sel I2S_2ch output clock selection 1'b0: select cru generated clock 1'b1: select io input clock
11:10	RO	0x0	reserved
9:8	RW	0x0	i2s_2ch_clk_sel Control I2S_2ch clock work frequency selection 2'b00: select divider ouput from pll divider 2'b01: select divider ouput from fraction divider 2'b10: select io input clock 2'b11: select 12MHz from osc input
7	RO	0x0	reserved
6:0	RW	0x1f	i2s_2ch_pll_div_con Control I2S_2ch PLL output divider freuency i2s_div_clk=i2s_div_src/(i2s_pll_div_con+1)

CRU_CLKSEL4_CON

Address: Operational Base + offset (0x0054)

Internal clock select and divide register4

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. 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	clk_24m_div_con Control clk_24m divider freuency 24m_div_clk=clk_24m/(clk_24m_div_con+1)
7:6	RW	0x0	tsp_pll_sel Control TSP clock PLL source selection 2'b00: select codec pll clock 2'b01: select general pll clock 2'b10: select general pll div2 clock
5	RO	0x0	reserved
4:0	RW	0x03	clk_tsp_div_con Control clk_tsp divider freuency tsp_div_clk=tsp_div_src/(clk_tsp_div_con+1)

CRU_CLKSEL5_CON
Address: Operational Base + offset (0x0058)
Internal clock select and divide register5

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	rmii_extclk_sel Control RMII external clock selection 1'b0: select internal divider clock 1'b1: select external input clock
14:8	RO	0x0	reserved
7:6	RW	0x0	mac_pll_sel Control MAC clock PLL source selection 2'b00: select codec pll clock 2'b01: select general pll clock 2'b10: select general pll div2 clock
5	RO	0x0	reserved
4:0	RW	0x03	clk_mac_div_con Control clk_mac divider freuency mac_div_clk=mac_div_src/(clk_mac_div_con+1)

CRU_CLKSEL6_CON
Address: Operational Base + offset (0x005c)
Internal clock select and divide register6

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:14	RW	0x0	spdif_pll_sel Control SPDIF PLL source selection 2'b00: select codec pll clock 2'b01: select general pll clock 2'b10: select general pll div2 clock
13:10	RO	0x0	reserved
9:8	RW	0x2	spdif_clk_sel Control SPDIF clock work frequency selection 2'b00: select divider ouput from pll divider 2'b01: select divider ouput from fraction divider 2'b10: select 12MHz from osc inpu
7	RO	0x0	reserved
6:0	RW	0x1f	spdif_pll_div_con Control SPDIF PLL output divider freuency spdif_div_clk=spdif_div_src/(spdif_pll_div_con+1)

CRU_CLKSEL7_CON
Address: Operational Base + offset (0x0060)
Internal clock select and divide register7

Bit	Attr	Reset Value	Description
31:0	RW	0x0bb8ea60	i2s_2ch_frac_factor Control I2S 2channel fraction divider frequency High 16-bit for numerator Low 16-bit for denominator

CRU_CLKSEL8_CON
Address: Operational Base + offset (0x0064)
Internal clock select and divide register8

Bit	Attr	Reset Value	Description
31:0	RW	0x0bb8ea60	i2s_8ch_frac_factor Control I2S 8channel fraction divider frequency High 16-bit for numerator Low 16-bit for denominator

CRU_CLKSEL9_CON
Address: Operational Base + offset (0x0068)
Internal clock select and divide register9

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:14	RW	0x0	i2s_8ch_pll_sel Control I2S_8ch PLL source selection 2'b00: select codec pll clock 2'b01: select general pll clock 2'b10: select general pll div2 clock
13:10	RO	0x0	reserved
9:8	RW	0x0	i2s_8ch_clk_sel Control I2S_8ch clock work frequency selection 2'b00: select divider ouput from pll divider 2'b01: select divider ouput from fraction divider 2'b10: select io input clock 2'b11: select 12MHz from osc input
7	RO	0x0	reserved
6:0	RW	0x1f	i2s_8ch_pll_div_con Control I2S_8ch PLL output divider freuency i2s_div_clk=i2s_div_src/(i2s_pll_div_con+1)

CRU_CLKSEL10_CON
Address: Operational Base + offset (0x006c)
Internal clock select and divide register10

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:14	RW	0x2	peri_pll_sel Control peripheral clock PLL source selection 2'b00: select general pll clock 2'b01: select codec pll clock 2'b10: select general div2 pll clock 2'b11: select general div3 pll clock

Bit	Attr	Reset Value	Description
13:12	RW	0x2	peri_pclk_div_con Control the divider ratio between aclk_periph and pclk_periph 2'b00: aclk_periph:pclk_periph = 1:1 2'b01: aclk_periph:pclk_periph = 2:1 2'b10: aclk_periph:pclk_periph = 4:1 2'b11: aclk_periph:pclk_periph = 8:1
11:10	RO	0x0	reserved
9:8	RW	0x1	peri_hclk_div_con Control the divider ratio between aclk_periph and hclk_periph 2'b00: aclk_periph:hclk_periph = 1:1 2'b01: aclk_periph:hclk_periph = 2:1 2'b10: aclk_periph:hclk_periph = 4:1
7:5	RO	0x0	reserved
4:0	RW	0x00	peri_aclk_div_con Control periphral clock divider frequency aclk_periph=periph_clk_src/(peri_aclk_div_con+1)

CRU_CLKSEL11_CON
Address: Operational Base + offset (0x0070)
Internal clock select and divide register11

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:14	RW	0x0	sfc_clk_pll_sel sfc pll source selection 2'b00: select codec pll clock 2'b01: select general pll clock 2'b10: select general pll div2 clock 2'b11: select 24M clock
13	RO	0x0	reserved
12:8	RW	0x00	sfc_div_con Control SFC clock divider frequency clk_sfc=sfc_clk_src/(sfc_div_con+1)
7:6	RW	0x0	mmc0_pll_sel Control mmc clock PLL source selection 2'b00: select codec pll clock 2'b01: select general pll clock 2'b10: select general pll div2 clock 2'b11: select 24M clock

Bit	Attr	Reset Value	Description
5:0	RW	0x17	mmc0_div_con Control SDMMC0 divider frequency clk_sdmmc0=general_pll_clk/(mmc0_div_con+1)

CRU_CLKSEL12_CON
Address: Operational Base + offset (0x0074)
Internal clock select and divide register12

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:14	RW	0x0	emmc_pll_sel Control emmc clock PLL source selection 2'b00: select codec pll clock 2'b01: select general pll clock 2'b10: select general pll div2 clock 2'b11: select 24M clock
13:8	RW	0x17	emmc_div_con Control EMMC divider frequency clk_emmc=general_pll_clk/(emmc_div_con+1)
7:6	RW	0x0	sdio_pll_sel Control sdio clock PLL source selection 2'b00: select codec pll clock 2'b01: select general pll clock 2'b10: select general pll div2 clock 2'b11: select 24M clock
5:0	RW	0x17	sdio_div_con Control SDIO divider frequency clk_sdio=general_pll_clk/(sdio_div_con+1)

CRU_CLKSEL13_CON
Address: Operational Base + offset (0x0078)
Internal clock select and divide register13

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. 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	uart12_pll_sel Control UART1 and UART2 clock PLL source selection 2'b00: select codec pll clock 2'b01: select general pll clock 2'b10: select general pll div2 clock
13:12	RW	0x1	uart0_pll_sel Control UART0 clock PLL source selection 2'b00: select codec pll clock 2'b01: select general pll clock 2'b10: select general pll div2 clock 2'b11: select USBPHY 480M clock
11:10	RO	0x0	reserved
9:8	RW	0x2	uart0_clk_sel Control UART0 clock work frequency selection 2'b00: select divider ouput from pll divider 2'b01: select divider ouput from fraction divider 2'b10: select 24MHz from osc inpu
7	RO	0x0	reserved
6:0	RW	0x1f	uart0_div_con Control UART0 divider frequency clk_uart0=uart_clk_src/(uart0_div_con+1)

CRU_CLKSEL14_CON

Address: Operational Base + offset (0x007c)

Internal clock select and divide register14

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. 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	uart1_clk_sel Control UART1 clock work frequency selection 2'b00: select divider ouput from pll divider 2'b01: select divider ouput from fraction divider 2'b10: select 24MHz from osc inpu
7	RO	0x0	reserved
6:0	RW	0x1f	uart1_div_con Control UART1 divider frequency clk_uart1=uart_clk_src/(uart1_div_con+1)

CRU_CLKSEL15_CON

Address: Operational Base + offset (0x0080)

Internal clock select and divide register15

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. 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	uart2_clk_sel Control UART2 clock work frequency selection 2'b00: select divider ouput from pll divider 2'b01: select divider ouput from fraction divider 2'b10: select 24MHz from osc inpu
7	RO	0x0	reserved
6:0	RW	0x1f	uart2_div_con Control UART2 divider frequency clk_uart2=uart_clk_src/(uart2_div_con+1)

CRU_CLKSEL17_CON

Address: Operational Base + offset (0x0088)

Internal clock select and divide register17

Bit	Attr	Reset Value	Description
31:0	RW	0x0bb8ea60	uart0_frac_factor Control UART0 fraction divider frequency High 16-bit for numerator Low 16-bit for denominator

CRU_CLKSEL18_CON

Address: Operational Base + offset (0x008c)

Internal clock select and divide register18

Bit	Attr	Reset Value	Description
31:0	RW	0x0bb8ea60	uart1_frac_factor Control UART1 fraction divider frequency High 16-bit for numerator Low 16-bit for denominator

CRU_CLKSEL19_CON

Address: Operational Base + offset (0x0090)

Internal clock select and divide register19

Bit	Attr	Reset Value	Description
31:0	RW	0x0bb8ea60	uart2_frac_factor Control UART2 fraction divider frequency High 16-bit for numerator Low 16-bit for denominator

CRU_CLKSEL20_CON

Address: Operational Base + offset (0x0094)
Internal clock select and divide register20

Bit	Attr	Reset Value	Description
31:0	RW	0x0bb8ea60	spdif_frac_factor Control SPDIF fraction divider frequency High 16-bit for numerator Low 16-bit for denominator

CRU_CLKSEL23_CON
Address: Operational Base + offset (0x00a0)
Internal clock select and divide register23

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:8	RW	0x01	ebc_dclk_div_con Control EBC clock divider frequency dclk_ebc=ebc_dclk_src/(ebc_dclk_div_con+1)
7:2	RO	0x0	reserved
1:0	RW	0x0	ebc_dclk_pll_sel Control EBC display clock PLL source selection 2'b00: select codec pll clock 2'b01: select general pll clock 2'b10: select general pll div2 clock

CRU_CLKSEL24_CON
Address: Operational Base + offset (0x00a4)
Internal clock select and divide register24

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:8	RW	0x17	saradc_div_con Control SARADC clock divider frequency clk_saradc=24MHz/(saradc_div_con+1)
7:2	RO	0x0	reserved
1:0	RW	0x1	crypto_div_con crypto clock divider frequency clk=clk/(div_con+1)

CRU_CLKSEL25_CON
Address: Operational Base + offset (0x00a8)

Internal clock select and divide register25

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. 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	0x1	spi_clk_pll_sel SPI clock pll source selection 2'b00: select codec pll clock 2'b01: select general pll clock 2'b10: select general pll div2 clock
7	RO	0x0	reserved
6:0	RW	0x1f	spi0_div_con Control SPI0 clock divider frequency clk_spi0=general_pll_clk/(spi0_div_con+1)

CRU_CLKSEL26_CON

Address: Operational Base + offset (0x00ac)

Internal clock select and divide register26

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. 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	ddr_clk_pll_sel DDR clock pll source selection 1'b0: select DDR PLL 1'b1: select GENERAL PLL div2
7:2	RO	0x0	reserved
1:0	RW	0x0	ddr_div_sel Control DDR divider frequency 2'b00: clk_ddr_src:clk_ddrphy = 1:1 2'b01: clk_ddr_src:clk_ddrphy = 2:1 2'b10: clk_ddr_src:clk_ddrphy = 4:1

CRU_CLKSEL27_CON

Address: Operational Base + offset (0x00b0)

Internal clock select and divide register27

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. 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	0x03	lcdc0_dclk_div_con Control LCDC clock divider frequency dclk_lcdc0=lcdc_dclk_src/(lcdc0_dclk_div_con+1)
7:2	RO	0x0	reserved
1:0	RW	0x1	lcdc0_dclk_pll_sel Control LCDC display clock PLL source selection 2'b00: select codec pll clock 2'b01: select general pll clock 2'b10: select general pll div2 clock 2'b11: select general pll div3 clock

CRU_CLKSEL28_CON

Address: Operational Base + offset (0x00b4)

Internal clock select and divide register28

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:8	RW	0x03	lcdc_sclk_div_con Control LCDC display clock divider frequency sclk_lcdc=lcdc_sclk_src/(lcdc_sclk_div_con+1)
7:2	RO	0x0	reserved
1:0	RW	0x1	lcdc_sclk_pll_sel Control LCDC display clock PLL source selection 2'b00: select codec pll clock 2'b01: select general pll clock 2'b10: select general pll div2 clock 2'b11: select general pll div3 clock

CRU_CLKSEL29_CON

Address: Operational Base + offset (0x00b8)

Internal clock select and divide register29

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. 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	RO	0x0	reserved
13:8	RW	0x05	pmu_div_con Control PMU clock divider frequency clk_pmu=pmu_clk_src/(pmu_div_con+1)
7	RW	0x0	cif0_clk_sel Control CIF0 clock selection 1'b0: select from PLL divider output 1'b1: select from 24MHz osc
6:2	RW	0x09	cif0_div_con Control CIF0 clock divider frequency clk_cif0=cif0_clk_src/(cif0_div_con+1)
1:0	RW	0x0	cif_pll_sel Control CIF clock PLL source selection 2'b00: select codec pll clock 2'b01: select general pll clock 2'b10: select general pll div2 clock 2'b11: usbphy480M

CRU_CLKSEL30_CON
Address: Operational Base + offset (0x00bc)
Internal clock select and divide register30

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:14	RW	0x0	vio_hclk_pll_sel Control VIO AHB clock PLL source selection 2'b00: select codec pll clock 2'b01: select general pll clock 2'b10: select general pll div2 clock 2'b11: select usbphy 480m clock
13	RO	0x0	reserved
12:8	RW	0x03	vio_hclk_div_con Control VIO AHB clock divider frequency hclk_vio=vio_hclk_src/(vio_hclk_div_con+1)
7	RW	0x0	cif0_clkin_inv_sel CIF0 input clock inverter selection 1'b0: select not invert 1'b1: select invert
6:5	RO	0x0	reserved
4:0	RW	0x00	testout_div_con testout clock divider frequency clk_testout=testout_clk_src/(testout_div_con+1)

CRU_CLKSEL31_CON
Address: Operational Base + offset (0x00c0)
Internal clock select and divide register31

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:13	RW	0x2	lcdc1_aclk_pll_sel Control LCDC1 AXI clock PLL source selection 3'b000: select codec pll clock 3'b001: select general pll clock 3'b010: select general pll div2 clock 3'b011: select general pll div3 clock 3'b100: select USBPHY 480M clock
12:8	RW	0x00	lcdc1_aclk_div_con Control LCDC1 AXI clock divider frequency aclk_lcdc1=lcdc1_aclk_src/(lcdc1_aclk_div_con+1)
7:5	RW	0x2	lcdc0_aclk_pll_sel Control LCDC0 AXI clock PLL source selection 3'b000: select codec pll clock 3'b001: select general pll clock 3'b010: select general pll div2 clock 3'b011: select general pll div3 clock 3'b100: select USBPHY 480M clock
4:0	RW	0x00	lcdc0_aclk_div_con Control LCDC0 AXI clock divider frequency aclk_lcdc0=lcdc0_aclk_src/(lcdc0_aclk_div_con+1)

CRU_CLKSEL32_CON
Address: Operational Base + offset (0x00c4)
Internal clock select and divide register32

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:13	RW	0x0	vdpu_aclk_pll_sel Control VDPU AXI clock PLL source selection 3'b000: select codec pll clock 3'b001: select general pll clock 3'b010: select general pll div2 clock 3'b011: select general pll div3 clock 3'b100: select USBPHY 480M clock

Bit	Attr	Reset Value	Description
12:8	RW	0x01	vdpu_aclk_div_con Control VDPU AXI clock divider frequency aclk_vdpu=vdpu_aclk_src/(vdpu_aclk_div_con+1)
7:5	RW	0x0	vepu_aclk_pll_sel Control VEPU AXI clock PLL source selection 3'b000: select codec pll clock 3'b001: select general pll clock 3'b010: select general pll div2 clock 3'b011: select general pll div3 clock 3'b100: select USBPHY 480M clock
4:0	RW	0x01	vepu_aclk_div_con Control VEPU AXI clock divider frequency aclk_vepu=vepu_aclk_src/(vepu_aclk_div_con+1)

CRU_CLKSEL34_CON
Address: Operational Base + offset (0x00cc)
Internal clock select and divide register34

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:13	RW	0x2	hevc_core_clk_pll_sel HEVC CORE clock PLL source selection 3'b000: select codec pll clock 3'b001: select general pll clock 3'b010: select general pll div2 clock 3'b011: select general pll div3 clock 3'b100: select USBPHY 480M clock
12:8	RW	0x01	hevc_core_clk_div_con HEVC CORE clock divider frequency clk_hevc_core=hevccore_clk_src/(hevc_core_clk_div_con+1)
7:5	RW	0x2	gpu_aclk_pll_sel Control GPU AXI clock PLL source selection 3'b000: select codec pll clock 3'b001: select general pll clock 3'b010: select general pll div2 clock 3'b011: select general pll div3 clock 3'b100: select USBPHY 480M clock

Bit	Attr	Reset Value	Description
4:0	RW	0x01	gpu_aclk_div_con Control GPU AXI clock divider frequency aclk_gpu=gpu_aclk_src/(gpu_aclk_div_con+1)

CRU_CLKGATE0_CON
Address: Operational Base + offset (0x00d0)
Internal clock gating control register0

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	testclk_gate_en Test output clock disable When HIGH, disable clock
14	RW	0x0	clk_i2s_2ch_gate_en I2S_2ch clock disable. When HIGH, disable clock
13	RW	0x0	clk_i2s_2ch_out_gate_en I2S_2ch output clock disable. When HIGH, disable clock
12	RW	0x0	clk_crypto_gate_en crypto clock disable. When HIGH, disable clock
11	RW	0x0	hclk_disp_gate_en display AHB clock disable. When HIGH, disable clock
10	RW	0x0	clk_i2s_2ch_frac_src_gate_en I2S_2ch fraction divider source clock disable. When HIGH, disable clock
9	RW	0x0	clk_i2s_2ch_src_gate_en I2S_2ch source clock disable. When HIGH, disable clock
8	RO	0x0	reserved
7	RW	0x0	aclk_core_gate_en ARM core axi clock(aclk_core) disable. When HIGH, disable clock
6	RW	0x0	core_src_clk_gate_en CORE source clock path clock disable. When HIGH, disable clock
5	RW	0x0	pclk_cpu_gate_en CPU system APB clock(pclk_cpu_pre) disable. When HIGH, disable clock
4	RW	0x0	hclk_cpu_gate_en CPU system AHB clock(hclk_cpu_pre) disable. When HIGH, disable clock

Bit	Attr	Reset Value	Description
3	RW	0x0	aclk_cpu_gate_en CPU system AXI clock(aclk_cpu_pre) disable. When HIGH, disable clock
2	RW	0x0	clk_ddrphy_src_gate_en DDR PHY clock(clk_ddrphy) disable. When HIGH, disable clock
1	RW	0x0	cpu_src_clk_gate_en CPU clock source clock disable. When HIGH, disable clock
0	RW	0x0	clk_core_periph_gate_en ARM core peripheral clock(clk_core_periph) disable. When HIGH, disable clock

CRU_CLKGATE1_CON
Address: Operational Base + offset (0x00d4)
Internal clock gating control register1

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. 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_tsp_gate_en clk_tsp clock disable. When HIGH, disable clock
13	RW	0x0	clk_uart2_frac_src_gate_en UART2 fraction divider source clock disable. When HIGH, disable clock
12	RW	0x0	clk_uart2_src_gate_en UART2 source clock disable. When HIGH, disable clock
11	RW	0x0	clk_uart1_frac_src_gate_en UART1 fraction divider source clock disable. When HIGH, disable clock
10	RW	0x0	clk_uart1_src_gate_en UART1 source clock disable. When HIGH, disable clock
9	RW	0x0	clk_uart0_frac_src_gate_en UART0 fraction divider source clock disable. When HIGH, disable clock
8	RW	0x0	clk_uart0_src_gate_en UART0 source clock disable. When HIGH, disable clock
7	RW	0x0	clk_mac_src_gate_en clk_mac source clock disable. When HIGH, disable clock

Bit	Attr	Reset Value	Description
6	RW	0x0	clk_otgphy1_gate_en OTGPHY1 clock(clk_otgphy1) disable. When HIGH, disable clock
5	RW	0x0	clk_otgphy0_gate_en OTGPHY0 clock(clk_otgphy0) disable. When HIGH, disable clock
4	RW	0x0	aclk_vio1_src_gate_en aclk_vio1_src clock disable. When HIGH, disable clock
3	RW	0x0	clk_jtag_gate_en JTAG clock disable. When HIGH, disable clock
2:1	RO	0x0	reserved
0	RW	0x0	pclk_pmu_src_gate_en pclk_pmu src clock disable. When HIGH, disable clock

CRU_CLKGATE2_CON
Address: Operational Base + offset (0x00d8)
Internal clock gating control register2

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	clk_mipiphy_24m_gate_en MIPI PHY 24M clock disable. When HIGH, disable clock
14	RW	0x0	clk_emmc_src_gate_en EMMC source clock disable. When HIGH, disable clock
13	RW	0x0	clk_sdio_src_gate_en SDIO source clock disable. When HIGH, disable clock
12	RW	0x0	clk_spdif_frac_src_gate_en SPDIF fraction divider source clock disable. When HIGH, disable clock
11	RW	0x0	clk_mmc0_src_gate_en SDMMC0 source clock disable. When HIGH, disable clock
10	RW	0x0	clk_spdif_src_gate_en SPDIF source clock disable. When HIGH, disable clock
9	RW	0x0	clk_spi0_src_gate_en SPI0 source clock disable. When HIGH, disable clock
8	RW	0x0	clk_saradc_src_gate_en SARADC source clock disable. When HIGH, disable clock

Bit	Attr	Reset Value	Description
7	RW	0x0	clk_mac_tx_gate_en clk_mac_tx clock disable. When HIGH, disable clock
6	RW	0x0	clk_mac_rx_gate_en clk_mac_rx clock disable. When HIGH, disable clock
5	RW	0x0	clk_mac_refout_gate_en clk_mac_refout clock disable. When HIGH, disable clock
4	RW	0x0	clk_mac_ref_gate_en clk_mac_ref clock disable. When HIGH, disable clock
3	RW	0x0	pclk_periph_gate_en PERIPH system APB clock(pclk_periph) disable. When HIGH, disable clock
2	RW	0x0	hclk_periph_gate_en PERIPH system AHB clock(hclk_periph) disable. When HIGH, disable clock
1	RW	0x0	aclk_periph_gate_en PERIPH system AXI clock(aclk_periph) disable. When HIGH, disable clock
0	RW	0x0	clk_periph_src_gate_en PERIPH system source clock disable. When HIGH, disable clock

CRU_CLKGATE3_CON
Address: Operational Base + offset (0x00dc)
Internal clock gating control register3

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	clk_sfc_gate_en sfc clock disable. When HIGH, disable clock
14	RW	0x0	hclk_gps_gate_en GPS AHB bus source axi clock disable. When HIGH, disable clock
13	RW	0x0	aclk_gpu_src_gate_en GPU AXI source clock disable. When HIGH, disable clock
12	RW	0x0	hclk_vdpu_gate_en VDPU AHB source clock disable. When HIGH, disable clock

Bit	Attr	Reset Value	Description
11	RW	0x0	aclk_vdpu_src_gate_en VDPU AXI source clock disable. When HIGH, disable clock
10	RW	0x0	clk_hevc_core_src_gate_en HEVC CORE clk source clock(clk_cif0_out) disable. When HIGH, disable clock
9	RW	0x0	aclk_vepu_src_gate_en VEPU AXI source clock disable. When HIGH, disable clock
8	RW	0x0	pclk_hdmi_gate_en HDMI APB bus clock disable. When HIGH, disable clock
7	RW	0x0	clk_cif_out_src_gate_en CIF out clk source clock(clk_cif0_out) disable. When HIGH, disable clock
6	RW	0x0	hclk_emem_peri_gate_en hclk_emem_peri souce clock disable. When HIGH, disable clock
5	RW	0x0	hclk_crypto_gate_en hclk_crypto clock disable. When HIGH, disable clock
4	RW	0x0	dclk_ebc_src_gate_en EBC DCLK souce clock disable. When HIGH, disable clock
3	RW	0x0	pclkin_cif_gate_en CIF pix input clk source clock disable. When HIGH, disable clock
2	RW	0x0	sclk_lcdc0_src_gate_en LCDC0 SCLK souce clock disable. When HIGH, disable clock
1	RW	0x0	dclk_lcdc0_src_gate_en LCDC0 DCLK souce clock disable. When HIGH, disable clock
0	RW	0x0	aclk_vio0_src_gate_en VIO0 AXI source clock disable. When HIGH, disable clock

CRU_CLKGATE4_CON

Address: Operational Base + offset (0x00e0)

Internal clock gating control register0

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. 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	RW	0x0	aclk_intmem_gate_en Internal memory AXI clock(aclk_intmem) disable. When HIGH, disable clock
11	RO	0x0	reserved
10	RW	0x0	aclk_strc_sys_gate_en CPU Structure system AXI clock disable. When HIGH, disable clock
9:7	RO	0x0	reserved
6	RW	0x0	clk_i2s_8ch_gate_en I2S_8ch clock disable. When HIGH, disable clock
5	RW	0x0	clk_i2s_8ch_frac_src_gate_en I2S_8ch fraction divider source clock disable. When HIGH, disable clock
4	RW	0x0	clk_i2s_8ch_src_gate_en I2S_8ch source clock disable. When HIGH, disable clock
3	RW	0x0	aclk_peri_axi_matrix_gate_en PERIPH matrix CPU AXI clock(aclk_peri_axi_matrix) disable. When HIGH, disable clock
2	RW	0x0	aclk_cpu_peri_gate_en PERIPH CPU AXI clock(aclk_cpu_peri) disable. When HIGH, disable clock
1	RW	0x0	pclk_peri_axi_matrix_gate_en PERIPH matrix CPU APB clock(pclk_peri_axi_matrix) disable. When HIGH, disable clock
0	RW	0x0	hclk_peri_axi_matrix_gate_en PERIPH matrix CPU AHB clock(hclk_peri_axi_matrix) disable. When HIGH, disable clock

CRU_CLKGATE5_CON
Address: Operational Base + offset (0x00e4)
Internal clock gating control register5

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. 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	pclk_acodec_gate_en audio codec APB clock disable. When HIGH, disable clock
13	RW	0x0	hclk_otg0_gate_en USB OTG PHY0 AHB clock disable. When HIGH, disable clock

Bit	Attr	Reset Value	Description
12	RO	0x0	reserved
11	RW	0x0	hclk_sdio_gate_en SDIO AHB clock disable. When HIGH, disable clock
10	RW	0x0	hclk_sdmmc0_gate_en SDMMC0 AHB clock disable When HIGH, disable clock
9	RW	0x0	hclk_nandc_gate_en NANDC AHB clock disable When HIGH, disable clock
8	RO	0x0	reserved
7	RW	0x0	pclk_ddrupctl_gate_en DDR uPCTL APB clock disable. When HIGH, disable clock
6	RW	0x0	hclk_rom_gate_en ROM AHB clock disable. When HIGH, disable clock
5	RO	0x0	reserved
4	RW	0x0	pclk_grf_gate_en GRF APB clock disable. When HIGH, disable clock
3	RO	0x0	reserved
2	RW	0x0	pclk_efuse_gate_en EFUSE APB clock disable. When HIGH, disable clock
1	RW	0x0	aclk_dmac_gate_en DMAC AXI clock disable. When HIGH, disable clock
0	RW	0x0	pclk_mipiphy_gate_en mipiphy apb clock disable. When HIGH, disable clock

CRU_CLKGATE6_CON
Address: Operational Base + offset (0x00e8)
Internal clock gating control register6

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. 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_vio0_gate_en VIO0 AXI clock disable. When HIGH, disable clock
12	RW	0x0	hclk_vio_bus_gate_en VIO AHB bus clock disable. When HIGH, disable clock

Bit	Attr	Reset Value	Description
11	RW	0x0	aclk_rga_gate_en RGA AXI clock disable. When HIGH, disable clock
10	RW	0x0	hclk_rga_gate_en RGA AHB clock disable. When HIGH, disable clock
9:6	RO	0x0	reserved
5	RW	0x0	aclk_cif_gate_en CIF AXI clock disable. When HIGH, disable clock
4	RW	0x0	hclk_cif_gate_en CIF AHB clock disable. When HIGH, disable clock
3:2	RO	0x0	reserved
1	RW	0x0	hclk_lcdc0_gate_en LCDC0 AHB clock disable. When HIGH, disable clock
0	RW	0x0	aclk_lcdc0_gate_en LCDC0 AXI clock disable. When HIGH, disable clock

CRU_CLKGATE7_CON
Address: Operational Base + offset (0x00ec)
Internal clock gating control register7

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. 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_gate_en WDT APB clock disable. When HIGH, disable clock
14	RW	0x0	pclk_saradc_gate_en SARADC APB clock disable. When HIGH, disable clock
13	RO	0x0	reserved
12	RW	0x0	pclk_spi0_gate_en SPI0 APB clock disable. When HIGH, disable clock
11	RO	0x0	reserved
10	RW	0x0	pclk_pwm01_gate_en PWM0 and PWM1 APB clock disable. When HIGH, disable clock
9:8	RO	0x0	reserved
7	RW	0x0	pclk_timer_gate_en TIMER APB clock disable. When HIGH, disable clock
6:5	RO	0x0	reserved

Bit	Attr	Reset Value	Description
4	RW	0x0	hclk_i2s_8ch_gate_en I2S_8ch AHB clock disable. When HIGH, disable clock
3	RW	0x0	hclk_host_gate_en USB HOST PHY AHB clock disable. When HIGH, disable clock
2	RW	0x0	hclk_i2s_2ch_gate_en I2S_2ch AHB clock disable. When HIGH, disable clock
1	RW	0x0	hclk_sfc_gate_en SFC AHB clock disable. When HIGH, disable clock
0	RW	0x0	hclk_emmc_gate_en EMMC AHB clock disable. When HIGH, disable clock

CRU_CLKGATE8_CON
Address: Operational Base + offset (0x00f0)
Internal clock gating control register8

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. 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_gpio3_gate_en GPIO3 APB clock disable. When HIGH, disable clock
11	RW	0x0	pclk_gpio2_gate_en GPIO2 APB clock disable. When HIGH, disable clock
10	RW	0x0	pclk_gpio1_gate_en GPIO1 APB clock disable. When HIGH, disable clock
9	RW	0x0	pclk_gpio0_gate_en GPIO0 APB clock disable. When HIGH, disable clock
8	RO	0x0	reserved
7	RW	0x0	pclk_i2c3_gate_en I2C3 APB clock disable. When HIGH, disable clock
6	RW	0x0	pclk_i2c2_gate_en I2C2 APB clock disable. When HIGH, disable clock
5	RW	0x0	pclk_i2c1_gate_en I2C1 APB clock disable. When HIGH, disable clock

Bit	Attr	Reset Value	Description
4	RW	0x0	pclk_i2c0_gate_en I2C0 APB clock disable. When HIGH, disable clock
3	RO	0x0	reserved
2	RW	0x0	pclk_uart2_gate_en UART2 APB clock disable. When HIGH, disable clock
1	RW	0x0	pclk_uart1_gate_en UART1 APB clock disable. When HIGH, disable clock
0	RW	0x0	pclk_uart0_gate_en UART0 APB clock disable. When HIGH, disable clock

CRU_CLKGATE9_CON
Address: Operational Base + offset (0x00f4)
Internal clock gating control register9

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	aclk_peri_niu_clock_en periph NIU AXI clock disable. When HIGH, disable clock
14	RW	0x0	hclk_peri_arbi_clock_en periph arbitor AHB clock disable. When HIGH, disable clock
13	RW	0x0	hclk_usb_peri_clock_en USB peri AHB clock disable. When HIGH, disable clock
12	RW	0x0	pclk_sim_clock_en SIM card APB clock disable. When HIGH, disable clock
11	RO	0x0	reserved
10	RW	0x0	aclk_vio1_clock_en VIO1 AXI clock disable. When HIGH, disable clock
9	RW	0x0	hclk_ebc_clock_en EBC AHB clock disable. When HIGH, disable clock
8	RW	0x0	aclk_iep_clock_en IEP AXI clock disable. When HIGH, disable clock
7	RW	0x0	hclk_iep_clock_en IEP AHB clock disable. When HIGH, disable clock

Bit	Attr	Reset Value	Description
6	RW	0x0	pclk_vio_mipi_gate_en pd_vio mipi controller clock disable. When HIGH, disable clock
5	RW	0x0	hclk_vio_h2p_gate_en pd_vio AHB h2p bridge clock disable. When HIGH, disable clock
4	RW	0x0	clk_l2c_gate_en L2C clock disable. When HIGH, disable clock
3	RW	0x0	pclk_pmu_noc_clock_en PD_PMU NOC APB clock disable. When HIGH, disable clock
2	RW	0x0	pclk_pmu_clock_en PMU APB clock disable. When HIGH, disable clock
1	RW	0x0	pclk_dbg_clock_en Debug APB clock disable. When HIGH, disable clock
0	RW	0x0	clk_core_dbg_gate_en Debug core clock disable. When HIGH, disable clock

CRU_CLKGATE10_CON
Address: Operational Base + offset (0x00f8)
Internal clock gating control register10

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	clk_nandc_gate_en Nandc clock disable. When HIGH, disable clock
14	RW	0x0	clk_old_usb_host_gate_en old usb host clock disable. When HIGH, disable clock
13	RW	0x0	clkin0_tsp_gate_en TSP IO clkin0 clock disable. When HIGH, disable clock
12	RW	0x0	hclk_tsp_gate_en TSP AHB clock disable. When HIGH, disable clock
11	RW	0x0	pclk_gmac_gate_en GMAC APB clock disable. When HIGH, disable clock
10	RW	0x0	aclk_gmac_gate_en GMAC AXI clock disable. When HIGH, disable clock

Bit	Attr	Reset Value	Description
9	RW	0x0	hclk_spdif_8ch_gate_en spdif_8ch AHB clock disable. When HIGH, disable clock
8	RW	0x0	clk_timer5_gate_en Timer5 clock disable. When HIGH, disable clock
7	RW	0x0	clk_timer4_gate_en Timer4 clock disable. When HIGH, disable clock
6	RW	0x0	clk_timer3_gate_en Timer3 clock disable. When HIGH, disable clock
5	RW	0x0	clk_timer2_gate_en Timer2 clock disable. When HIGH, disable clock
4	RW	0x0	clk_timer1_gate_en Timer1 clock disable. When HIGH, disable clock
3	RW	0x0	clk_timer0_gate_en Timer0 clock disable. When HIGH, disable clock
2	RW	0x0	func_pvtm_gate_en func PVTM clock disable. When HIGH, disable clock
1	RW	0x0	gpu_pvtm_gate_en pd_gpu PVTM clock disable. When HIGH, disable clock
0	RW	0x0	core_pvtm_gate_en pd_core PVTM clock disable. When HIGH, disable clock

CRU_GLB_SRST_FST_VALUE

Address: Operational Base + offset (0x0100)

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 (0x0104)

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_SOFTRST0_CON
Address: Operational Base + offset (0x0110)
Internal software reset control register0

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	l2c_srstn_req L2C software reset request. When HIGH, reset relative logic
14	RW	0x0	strc_sys_asrstn_req Structrue system AXI software reset request. When HIGH, reset relative logic
13	R/WSC	0x0	aclk_core_srstn_req core AXI clock software reset request. When HIGH, reset relative logic
12	RW	0x0	core_top_dbg_srstn_req CPU top debug software reset request. When HIGH, reset relative logic
11	RW	0x0	core3_dbg_srstn_req core3 CPU debug software reset request. When HIGH, reset relative logic
10	RW	0x0	core2_dbg_srstn_req core2 CPU debug software reset request. When HIGH, reset relative logic
9	RW	0x0	core1_dbg_srstn_req core1 CPU debug software reset request. When HIGH, reset relative logic
8	RW	0x0	core0_dbg_srstn_req core0 CPU debug software reset request. When HIGH, reset relative logic
7	R/WSC	0x0	core3_srstn_req core3 CPU software reset request. When HIGH, reset relative logic
6	R/WSC	0x0	core2_srstn_req core2 CPU software reset request. When HIGH, reset relative logic
5	R/WSC	0x0	core1_srstn_req core1 CPU software reset request. When HIGH, reset relative logic
4	R/WSC	0x0	core0_srstn_req core0 CPU software reset request. When HIGH, reset relative logic
3	R/WSC	0x0	core3_posrstn_req core3 CPU PO software reset request. When HIGH, reset relative logic

Bit	Attr	Reset Value	Description
2	R/WSC	0x0	core2_posrstn_req core2 CPU PO software reset request. When HIGH, reset relative logic
1	R/WSC	0x0	core1_posrstn_req core1 CPU PO software reset request. When HIGH, reset relative logic
0	R/WSC	0x0	core0_posrstn_req core0 CPU PO software reset request. When HIGH, reset relative logic

CRU_SOFTRST1_CON
Address: Operational Base + offset (0x0114)
Internal software reset control register1

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	acodec_psrstn_req audio codec software reset request. When HIGH, reset relative logic
14	RW	0x0	efuse_psrstn_req EFUSE APB software reset request. When HIGH, reset relative logic
13	RW	0x0	core_pvtm_srstn_req CORE PVTM software reset request. When HIGH, reset relative logic
12	RO	0x0	reserved
11	RW	0x0	func_pvtm_srstn_req func PVTM software reset request. When HIGH, reset relative logic
10	RW	0x0	gpu_pvtm_srstn_req GPU PVTM software reset request. When HIGH, reset relative logic
9	RW	0x0	i2s_8ch_srstn_req I2S 8channel software reset request. When HIGH, reset relative logic
8	RW	0x0	i2s_2ch_srstn_req I2S 2channel software reset request. When HIGH, reset relative logic
7	RW	0x0	peri_niu_srstn_req periph_niu software reset request. When HIGH, reset relative logic
6	RW	0x0	rom_srstn_req ROM software reset request. When HIGH, reset relative logic
5	RW	0x0	intmem_srstn_req Internal memory software reset request. When HIGH, reset relative logic

Bit	Attr	Reset Value	Description
4	RW	0x0	spdif_srstn_req SPDIF software reset request. When HIGH, reset relative logic
3	RW	0x0	ahb2apb_hsrstn_req AHB2APB software reset request. When HIGH, reset relative logic
2	RW	0x0	cpusys_hsrstn_req CPU AHB software reset request. When HIGH, reset relative logic
1:0	RO	0x0	reserved

CRU_SOFTRST2_CON
Address: Operational Base + offset (0x0118)
Internal software reset control register2

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	sfc_srstn_req SFC software reset request. When HIGH, reset relative logic
14	RW	0x0	i2c3_srstn_req I2C3 software reset request. When HIGH, reset relative logic
13	RW	0x0	i2c2_srstn_req I2C2 software reset request. When HIGH, reset relative logic
12	RW	0x0	i2c1_srstn_req I2C1 software reset request. When HIGH, reset relative logic
11	RW	0x0	i2c0_srstn_req I2C0 software reset request. When HIGH, reset relative logic
10	RO	0x0	reserved
9	RW	0x0	uart2_srstn_req UART2 software reset request. When HIGH, reset relative logic
8	RW	0x0	uart1_srstn_req UART1 software reset request. When HIGH, reset relative logic
7	RW	0x0	uart0_srstn_req UART0 software reset request. When HIGH, reset relative logic
6:5	RO	0x0	reserved
4	RW	0x0	mipiphy_psrstn_req mipiphy apb bus software reset request. When HIGH, reset relative logic

Bit	Attr	Reset Value	Description
3	RW	0x0	gpio3_srstn_req GPIO3 software reset request. When HIGH, reset relative logic
2	RW	0x0	gpio2_srstn_req GPIO2 software reset request. When HIGH, reset relative logic
1	RW	0x0	gpio1_srstn_req GPIO1 software reset request. When HIGH, reset relative logic
0	RW	0x0	gpio0_srstn_req GPIO0 software reset request. When HIGH, reset relative logic

CRU_SOFTRST3_CON
Address: Operational Base + offset (0x011c)
Internal software reset control register3

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	usb_peri_srstn_req USB PERIPH software reset request. When HIGH, reset relative logic
14	RW	0x0	emem_peri_srstn_req EMEM PERIPH software reset request. When HIGH, reset relative logic
13	RW	0x0	cpu_peri_srstn_req CPU PERIPH software reset request. When HIGH, reset relative logic
12	RW	0x0	smart_card_srstn_req smart_card software reset request. When HIGH, reset relative logic
11	RW	0x0	periphsys_psrstn_req PERIPH APB software reset request. When HIGH, reset relative logic
10	RW	0x0	periphsys_hsrstn_req PERIPH AHB software reset request. When HIGH, reset relative logic
9	RW	0x0	periphsys_asrstn_req PERIPH AXI software reset request. When HIGH, reset relative logic
8	RW	0x0	gmac_srstn_req GMAC software reset request. When HIGH, reset relative logic
7	RW	0x0	grf_srstn_req GRF software reset request. When HIGH, reset relative logic
6	RO	0x0	reserved

Bit	Attr	Reset Value	Description
5	RW	0x0	crypto_srstn_req crypto software reset request. When HIGH, reset relative logic
4	RW	0x0	dap_sys_srstn_req DAP system software reset request. When HIGH, reset relative logic
3	RW	0x0	dap_srstn_req DAP software reset request. When HIGH, reset relative logic
2	RW	0x0	dap_po_srstn_req DAP power software reset request. When HIGH, reset relative logic
1	RO	0x0	reserved
0	RW	0x0	pwm0_srstn_req PWM0 software reset request. When HIGH, reset relative logic

CRU_SOFTRST4_CON
Address: Operational Base + offset (0x0120)
Internal software reset control register4

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	ddrmsch_srstn_req DDR memory scheduler software reset request. When HIGH, reset relative logic
14:11	RO	0x0	reserved
10	RW	0x0	otgc1_srstn_req OTG controller1 software reset request. When HIGH, reset relative logic Host Controller utmi_clk domain reset
9	RO	0x0	reserved
8	RW	0x0	usbotg1_srstn_req USBOTG1 software reset request. When HIGH, reset relative logic Host Controller hclk domain reset.
7	RW	0x0	otgc0_srstn_req OTG controller0 software reset request. When HIGH, reset relative logic. OTG Controller utmi_clk domain reset.
6	RO	0x0	reserved
5	RW	0x0	usbotg0_srstn_req USBOTG0 software reset request. When HIGH, reset relative logic OTG Controller hclk domain reset.

Bit	Attr	Reset Value	Description
4	RW	0x0	nandc_srstn_req NANDC software reset request. When HIGH, reset relative logic
3	RW	0x0	gps_srstn_req GPS software reset request. When HIGH, reset relative logic
2:1	RO	0x0	reserved
0	RW	0x0	dma2_srstn_req DMA2 software reset request. When HIGH, reset relative logic

CRU_SOFTRST5_CON
Address: Operational Base + offset (0x0124)
Internal software reset control register5

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. 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	usbhost0_ehci_srstn_req usbhost0_ehci clock input0 domain software reset request. When HIGH, reset relative logic
13	RW	0x0	tsp_clkin0_srstn_req TSP clock input0 domain software reset request. When HIGH, reset relative logic
12	RW	0x0	tsp_srstn_req TSP software reset request. When HIGH, reset relative logic
11	RW	0x0	ddrctrl_psrstn_req DDR controller APB software reset request. When HIGH, reset relative logic
10	RW	0x0	ddrctrl_srstn_req DDR controller software reset request. When HIGH, reset relative logic
9	RW	0x0	ddrphy_psrstn_req DDR PHY APB software reset request. When HIGH, reset relative logic
8	RW	0x0	ddrphy_srstn_req DDR PHY software reset request. When HIGH, reset relative logic
7	RW	0x0	saradc_srstn_req SARADC software reset request. When HIGH, reset relative logic
6	RW	0x0	wdt_srstn_req WDT software reset request. When HIGH, reset relative logic

Bit	Attr	Reset Value	Description
5	RO	0x0	reserved
4	RW	0x0	spi0_srstn_req SPI0 software reset request. When HIGH, reset relative logic
3	RW	0x0	emmc_srstn_req EMMC software reset request. When HIGH, reset relative logic
2	RW	0x0	sdio_srstn_req SDIO software reset request. When HIGH, reset relative logic
1	RW	0x0	mmc0_srstn_req SDMMC0 software reset request. When HIGH, reset relative logic
0	RO	0x0	reserved

CRU_SOFTRST6_CON
Address: Operational Base + offset (0x0128)
Internal software reset control register6

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	pmu_srstn_req PMU software reset request. When HIGH, reset relative logic
14	RW	0x0	cif0_srstn_req CIF0 software reset request. When HIGH, reset relative logic
13	RW	0x0	rga_hsrstn_req RGA AHB software reset request. When HIGH, reset relative logic
12	RW	0x0	rga_asrstn_req RGA AXI software reset request. When HIGH, reset relative logic
11	RW	0x0	iep_hsrstn_req IEP AHB software reset request. When HIGH, reset relative logic
10	RW	0x0	iep_asrstn_req IEP AXI software reset request. When HIGH, reset relative logic
9	RW	0x0	usbpor_srst_req USBPHY POR software reset request. When HIGH, reset relative logic. USB phy analog domain reset, including both OTG and HOST phy .

Bit	Attr	Reset Value	Description
8	RW	0x0	utmi1_srst_req UTMI1 software reset request. When HIGH, reset relative logic HOST phy digital domain reset. It should last at least 10 utmi_clk_1 cycles.
7	RW	0x0	utmi0_srstn_req UTMI0 software reset request. When HIGH, reset relative logic OTG phy digital domain reset. It should last at least 10 utmi_clk_0 cycles.
6	RW	0x0	lcdc0_dsrstn_req LCDC0 DCLK software reset request. When HIGH, reset relative logic
5	RW	0x0	lcdc0_hsrstn_req LCDC0 AHB software reset request. When HIGH, reset relative logic
4	RW	0x0	lcdc0_asrstn_req LCDC0 AXI software reset request. When HIGH, reset relative logic
3	RW	0x0	vio_bus_hsrstn_req VIO bus AHB software reset request. When HIGH, reset relative logic
2	RW	0x0	vio0_asrstn_req VIO 0 NIU AXI software reset request. When HIGH, reset relative logic
1	RW	0x0	vio_arbi_hsrstn_req VIO arbitor AHB software reset request. When HIGH, reset relative logic
0	RW	0x0	hdmi_psrstn_req HDMI PCLK software reset request. When HIGH, reset relative logic

CRU_SOFTRST7_CON
Address: Operational Base + offset (0x012c)
Internal software reset control register7

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. 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	ebc_hsrstn_req EBC AHB software reset request. When HIGH, reset relative logic
11	RW	0x0	ebc_asrstn_req EBC AXI software reset request. When HIGH, reset relative logic

Bit	Attr	Reset Value	Description
10	RW	0x0	gpu_niu_asrstn_req GPU NIU AXI software reset request. When HIGH, reset relative logic
9	RO	0x0	reserved
8	RW	0x0	gpu_srstn_req GPU core software reset request. When HIGH, reset relative logic
7	RW	0x0	lcdc0_ssrstn_req LCDC0 SCLK software reset request. When HIGH, reset relative logic
6	RO	0x0	reserved
5	RW	0x0	pmu_niu_psrstn_req PD_PMU NIU APB software reset request. When HIGH, reset relative logic
4	RW	0x0	vcodec_niu_asrstn_req VCODEC NIU AXI software reset request. When HIGH, reset relative logic
3	RW	0x0	hevc_core_srstn_req HEVC CORE software reset request. When HIGH, reset relative logic
2	RW	0x0	vio1_asrstn_req VIO second AXI software reset request. When HIGH, reset relative logic
1	RW	0x0	vcodec_hsrstn_req VCODEC AHB software reset request. When HIGH, reset relative logic
0	RW	0x0	vcodec_asrstn_req VCODEC AXI software reset request. When HIGH, reset relative logic

CRU_SOFTRST8_CON
Address: Operational Base + offset (0x0130)
Internal software reset control register8

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. 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	vio_mipi_dsi_srstn_req pd_vio mipi_dsi software reset request. When HIGH, reset relative logic
8	RW	0x0	vio_h2p_srstn_req pd_vio h2p bridge software reset request. When HIGH, reset relative logic
7	RW	0x0	timer5_srstn_req Timer5 software reset request. When HIGH, reset relative logic

Bit	Attr	Reset Value	Description
6	RW	0x0	timer4_srstn_req Timer4 software reset request. When HIGH, reset relative logic
5	RW	0x0	timer3_srstn_req Timer3 software reset request. When HIGH, reset relative logic
4	RW	0x0	timer2_srstn_req Timer2 software reset request. When HIGH, reset relative logic
3	RW	0x0	timer1_srstn_req Timer1 software reset request. When HIGH, reset relative logic
2	RW	0x0	timer0_srstn_req Timer0 software reset request. When HIGH, reset relative logic
1	RW	0x0	dbg_psrstn_req DEBUG APB software reset request. When HIGH, reset relative logic
0	RW	0x0	core_dbg_srstn_req CORE DEBUG software reset request. When HIGH, reset relative logic

CRU_MISC_CON

Address: Operational Base + offset (0x0134)

SCU control register

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x1	usb480_src_sel USB480 and 24M selection 1'b0: select 24M 1'b1: select 480M
14:11	RO	0x0	reserved
10:8	RW	0x0	testclk_sel Output clock selection for test 3'b000: clk_pvtm 3'b001: sclk_lcdc 3'b010: clk_core 3'b011: clk_ddrphy 3'b100: aclk_vio0 3'b101: aclk_gpu 3'b110: aclk_peri 3'b111: aclk_bus
7:1	RO	0x0	reserved

Bit	Attr	Reset Value	Description
0	RW	0x0	core0_porst_wdt_sel Select reset watchdog when A9 core 0 power on reset 1'b0: not reset watchdog 1'b1: reset watchdog

CRU_GLB_CNT_TH
Address: Operational Base + offset (0x0140)
global reset wait counter threshold

Bit	Attr	Reset Value	Description
31:16	RW	0x3a98	pll_lock_period PLL lock period
15	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
14	RO	0x0	reserved
13:12	RW	0x0	pmu_glb_srst_ctrl watch dog trigger 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
11:10	RO	0x0	reserved
9:0	RW	0x064	glb_rst_cnt_th Global soft reset counter threshold

CRU_GLB_RST_ST
Address: Operational Base + offset (0x0150)
global reset status

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

Bit	Attr	Reset Value	Description
0	W1C	0x0	fst_glb_rst_st first global rst flag 1'b0: last hot reset is not first global rst 1'b1: last hot reset is first global rst

CRU_SDMMC_CON0
Address: Operational Base + offset (0x01c0)
sdmmc control0

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. 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	WO	0x0	sdmmc_drv_sel sdmmc drive select sdmmc drive select
10:3	WO	0x00	sdmmc_drv_delaynum sdmmc drive delay number sdmmc drive delay number
2:1	WO	0x2	sdmmc_drv_degree sdmmc drive degree sdmmc drive degree
0	WO	0x0	sdmmc_init_state sdmmc initial state sdmmc initial state

CRU_SDMMC_CON1
Address: Operational Base + offset (0x01c4)
sdmmc control1

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. 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	WO	0x0	sdmmc_sample_sel sdmmc sample select sdmmc sample select
9:2	WO	0x00	sdmmc_sample_delaynum sdmmc sample delay number sdmmc sample delay number
1:0	WO	0x0	sdmmc_sample_degree sdmmc sample degree sdmmc sample degree

CRU_SDIO_CON0
Address: Operational Base + offset (0x01c8)
sdio0 control0

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. 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	WO	0x0	sdio0_drv_sel sdio0 drive select sdio0 drive select
10:3	WO	0x00	sdio0_drv_delaynum sdio0 drive delay number sdio0 drive delay number
2:1	WO	0x2	sdio0_drv_degree sdio0 drive degree sdio0 drive degree
0	WO	0x0	sdio0_init_state sdio0 initial state sdio0 initial state

CRU_SDIO_CON1
Address: Operational Base + offset (0x01cc)
sdio0 control1

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. 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	WO	0x0	sdio0_sample_sel sdio0 sample select sdio0 sample select
9:2	WO	0x00	sdio0_sample_delaynum sdio0 sample delay number sdio0 sample delay number
1:0	WO	0x0	sdio0_sample_degree sdio0 sample degree sdio0 sample degree

CRU_EMMC_CON0
Address: Operational Base + offset (0x01d8)
emmc control0

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. 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	WO	0x0	emmc_drv_sel emmc drive select emmc drive select
10:3	WO	0x00	emmc_drv_delaynum emmc drive delay number emmc drive delay number
2:1	WO	0x2	emmc_drv_degree emmc drive degree emmc drive degree
0	WO	0x0	emmc_init_state emmc initial state emmc initial state

CRU_EMMC_CON1
Address: Operational Base + offset (0x01dc)
emmc control1

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. 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	WO	0x0	emmc_sample_sel emmc sample select emmc sample select
9:2	WO	0x00	emmc_sample_delaynum emmc sample delay number emmc sample delay number
1:0	WO	0x0	emmc_sample_degree emmc sample degree emmc sample degree

CRU_PLL_PRG_EN
Address: Operational Base + offset (0x01f0)
PLL program enable

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:0	WO	0x0000	pll_prg_en pll program enable when pll_prg_en is 16'h5a5a, all the pll_con can be programed

3.8 Timing Diagram

Power on reset timing is shown as follow:

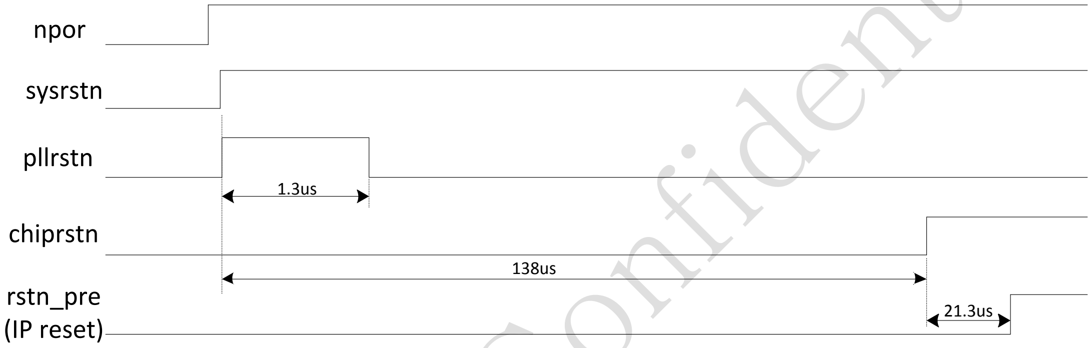

Fig. 3-7Chip Power On Reset Timing Diagram

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 1500 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 2.4GHz for all four PLLs (ARM PLL, DDR PLL, CODEC PLL and GENERAL PLL).

A. PLL output frequency configuration

FBDIV,POSTDIV1, BYPASS can be configured by programming CRU_APLL_CON0, CRU_DPLL_CON0, CRU_CPLL_CON0 and CRU_GPLL_CON0.

DSMPD, REFDIV, POSTDIV2 can be configured by programming CRU_APLL_CON1, CRU_DPLL_CON1, CRU_CPLL_CON1 and CRU_GPLL_CON1.

FRAC can be configured by programming CRU_APLL_CON2, CRU_DPLL_CON2, CRU_CPLL_CON2 and CRU_GPLL_CON2.

(1) 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 = 175
POSTDIV1=1
POSTDIV2=1

And then
FOUTVCO = FREF / REFDIV * FBDIV = 24/6*175=700
FOUTPOSTDIV = FOUTVCO / POSTDIV1 / POSTDIV2=700/1/1=700

(2) If DSMPD = 0 (DSM is enabled, "fractional mode")
FOUTVCO = FREF / REFDIV * (FBDIV + FRAC / 2^{24})
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 / 2^{24}) = 24/1*(40+24’hf5c28f /2^{24})= 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 Rangein Integer Mode):1MHz to 800MHz
FREF(Input Frequency Rangein Fractional Mode):10MHz to 800MHz
FREF/REFDIV(The divided reference frequency): 1 to 50MHz
FOUTVCO: 600NHz to 2.4GHz

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_APLL_CON1[10], CRU_DPLL_CON1[10],CRU_CPLL_CON1[10],CRU_GPLL_CON1[10] 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 1500 REF_CLK.

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 2-5 s. 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 256 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 256*REFDIV/FREF.

3.9.3 Fractional divider usage

To get specific frequency, clocks of I2S, SPDIF, UART, HSADC 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 and HSADC.

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.

After global reset, the reset trigger source can be check in CRU_GLB_RST_ST.