

***Rockchip
RK3036
Technical Reference Manual
System and System Control***

**Revision 1.0
Sep. 2015**

Revision History

Date	Revision	Description
2015-9-7	1.0	Update-Part1 System and System Control

Table of Content

Table of Content	3
Figure Index	5
Table Index.....	6
Warranty Disclaimer.....	7
Chapter 1 Introduction	8
1.1 Features	8
1.2 Block Diagram	15
Chapter 2 System Overview	16
2.1 Address Mapping.....	16
2.2 System Boot.....	16
2.3 System Interrupt connection.....	18
2.4 System DMA hardware request connection.....	21
Chapter 3 Clock & Reset Unit (CRU)	22
3.1 Overview	22
3.2 Block Diagram	22
3.3 System Reset Solution	22
3.4 Function Description	23
3.5 PLL Introduction.....	23
3.6 Register Description	24
3.7 Timing Diagram	64
3.8 Application Notes	64
Chapter 4 Embedded Processor (Cortex-A7).....	67
4.1 Overview	67
4.2 Block Diagram	67
4.3 Function Description	67
4.4 Register Description	67
Chapter 5 General Register Files (GRF).....	68
5.1 Overview	68
5.2 Function Description	68
5.3 Register Description	68
Chapter 6 DMC (Dynamic Memory Interface).....	124
6.1 Overview	124
6.2 Block Diagram	125
6.3 Function Description	125
6.4 Register Description	127
6.5 Timing Diagram	170
6.6 Interface Description	171
6.7 Application Notes	171
Chapter 7 Nand Flash Controller (NandC).....	179
7.1 Overview	179
7.2 Block Diagram	180
7.3 Function Description	180
7.4 Register Description	181
7.5 Interface Description	207
7.6 Application Notes	208
Chapter 8 Timer	215
8.1 Overview	215
8.2 Block Diagram	215
8.3 Function Description	215

8.4 Register Description	216
8.5 Application Notes	218
Chapter 9 Generic Interrupt Controller (GIC).....	219
9.1 Overview	219
9.2 Block Diagram	219
Chapter 10 DMA Controller(DMAC).....	220
10.1 Overview.....	220
10.2 Block Diagram	220
10.3 Function Description	221
10.4 Register Description.....	222
10.5 Timing Diagram	238
10.6 Interface Description.....	238
10.7 Application Notes.....	239
Chapter 11 Process-Voltage-Temperature Monitor (PVTM)	245
11.1 Overview.....	245
11.2 Block Diagram	245
11.3 Function Description	245
11.4 Application Notes.....	246
Chapter 12 EFUSE.....	247
12.1 Overview.....	247
12.2 Block Diagram	247
12.3 Function Description	247
12.4 Register Description.....	248
12.5 Timing Diagram	249
12.6 Application Notes.....	250
Chapter 13 WatchDog	251
13.1 Overview.....	251
13.2 Block Diagram	251
13.3 Function Description	251
13.4 Operation	251
13.5 Programming sequence	252
13.6 Register Description.....	252
13.7 Application Notes.....	255
Chapter 14 System Debug	256
14.1 Overview.....	256
14.2 Block Diagram	256
14.3 Function Description	256
14.4 Register Description.....	256
14.5 Interface Description.....	257

Figure Index

Fig. 1-1 RK3036 Block Diagram	15
Fig. 2-1 RK3036 Address Mapping.....	16
Fig. 2-2 RK3036 boot procedure flow.....	18
Fig. 3-1 CRU Architecture	22
Fig. 3-2 Reset Architecture Diagram	22
Fig. 3-3 PLL Block Diagram	24
Fig. 3-4 Chip Power On Reset Timing Diagram	64
Fig. 4-1 MP Subsystem architecture	67
Fig. 7-1 NandC Block Diagram	180
Fig. 7-2 NandC Address Assignment.....	210
Fig. 7-3 NandC Data Format.....	211
Fig. 7-4 NandC LLP Data Format	213
Fig. 8-1 Timer Block Diagram	215
Fig. 8-2 Timer Usage Flow	216
Fig. 8-3 Timing between timer_en and timer_clk.....	218
Fig. 9-1 GIC Block Diagram.....	219
Fig. 10-1 Block diagram of DMAC.....	220
Fig. 10-2 DMAC operation states.....	221
Fig. 10-3 DMAC request and acknowledge timing	238
Fig. 11-1 PVTM Block Diagram.....	245
Fig. 12-1 EFUSE Block Diagram	247
Fig. 12-2 EFUSE Timing Diagram in PGM Mode.....	249
Fig. 12-3 EFUSE Timing Diagram in Read Mode	250
Fig. 13-1 WDT block diagram	251
Fig. 13-2 WDT Operation Flow	252
Fig. 14-1 Debug system structure	256
Fig. 14-2 DAP SWJ interface.....	257
Fig. 14-3 SW-DP acknowledgement timing	257

Table Index

Table 2-1 RK3036 Interrupt connection list.....	19
Table 2-2 RK3036 DMAC_PERI Hardware request connection list	21
Table 6-1 meaning of the parameter.....	170
Table 6-2 DDR IO description	171
Table 6-3 DDR PHY TX DLLs Delay Step	175
Table 6-4 DDR PHY RX DQS Delay Step.....	176
Table 6-5 DM/DQ/DQS/CMD Driver output resistance.....	176
Table 6-6 DM, DQ Signal Drive Strength Register	176
Table 6-7 DM/DQ/DQS/CMD Driver output resistance with control bit	176
Table 6-8 DM/DQ/DQS RX ODT resistance with control bit	176
Table 6-9 Low Power DLL Setting	177
Table 6-10 per-bit de-skew tuning resolution	177
Table 7-1 NandC Address Mapping	181
Table 7-2 NandC Interface Description	207
Table 7-3 NandC Interface Connection	208
Table 7-4 NandC Page/Spare size for flash.....	212
Table 10-1 DMAC Request Mapping Table	220
Table 10-2 DMAC boot interface.....	238
Table 10-3 Source size in CCRn.....	242
Table 10-4 DMAC Instruction sets	243
Table 10-5 DMAC instruction encoding	244
Table 11-1 core_pvtm control source and result destination	245
Table 11-2 gpu_pvtm control source and result destination.....	246
Table 11-3 video_pvtm control source and result destination.....	246
Table 12-1 EFUSE Operation Mode	247
Table 12-2 EFUSE Read Mode	248
Table 12-3 EFUSE Timing Requirements in PGM Mode	249
Table 12-4 EFUSE Timing Requirements in Read Mode	250
Table 14-1 SW-DP Interface Description	257

Warranty Disclaimer

Rockchip Electronics Co., Ltd makes no warranty, representation or guarantee (expressed, implied, statutory, or otherwise) by or with respect to anything in this document, and shall not be liable for any implied warranties of non-infringement, merchantability or fitness for a particular purpose or for any indirect, special or consequential damages.

Information furnished is believed to be accurate and reliable. However, Rockchip Electronics Co., Ltd assumes no responsibility for the consequences of use of such information or for any infringement of patents or other rights of third parties that may result from its use.

Rockchip Electronics Co., Ltd's products are not designed, intended, or authorized for using as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Rockchip Electronics Co., Ltd's product could create a situation where personal injury or death may occur, should buyer purchase or use Rockchip Electronics Co., Ltd's products for any such unintended or unauthorized application, buyers shall indemnify and hold Rockchip Electronics Co., Ltd and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, expenses, and reasonable attorney fees arising out of, either directly or indirectly, any claim of personal injury or death that may be associated with such unintended or unauthorized use, even if such claim alleges that Rockchip Electronics Co., Ltd was negligent regarding the design or manufacture of the part.

Copyright and Patent Right

Information in this document is provided solely to enable system and software implementers to use Rockchip Electronics Co., Ltd's products. There are no expressed or implied copyright licenses granted hereunder to design or fabricate any integrated circuits or integrated circuits based on the information in this document.

Rockchip Electronics Co., Ltd does not convey any license under its patent rights nor the rights of others.

All copyright and patent rights referenced in this document belong to their respective owners and shall be subject to corresponding copyright and patent licensing requirements.

Trademarks

Rockchip and Rockchip™ logo and the name of Rockchip Electronics Co., Ltd's products are trademarks of Rockchip Electronics Co., Ltd. and are exclusively owned by Rockchip Electronics Co., Ltd. References to other companies and their products use trademarks owned by the respective companies and are for reference purpose only.

Confidentiality

The information contained herein (including any attachments) is confidential. The recipient hereby acknowledges the confidentiality of this document, and except for the specific purpose, this document shall not be disclosed to any third party.

Reverse engineering or disassembly is prohibited.

ROCKCHIP ELECTRONICS CO.,LTD. RESERVES THE RIGHT TO MAKE CHANGES IN ITS PRODUCTS OR PRODUCT SPECIFICATIONS WITH THE INTENT TO IMPROVE FUNCTION OR DESIGN AT ANY TIME AND WITHOUT NOTICE AND IS NOT REQUIRED TO UNDATE THIS DOCUMENTATION TO REFLECT SUCH CHANGES.

Copyright © 2015 Rockchip Electronics Co., Ltd.

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electric or mechanical, by photocopying, recording, or otherwise, without the prior written consent of Rockchip Electronics Co., Ltd.

Chapter 1 Introduction

RK3036 is a low power consumption, high performance processor solution for low-end OTT TV BOX, and other digital multimedia applications, and integrates dual-core Cortex-A7, with separate NEON coprocessor and 128KB L2 Cache.

Many embedded powerful hardware engines provide optimized performance for high-end application. RK3036 supports almost full-format 1080P H.264 decoder and H.265 decoder, high-quality JPEG decoder and special image preprocessor and postprocessor.

Embedded 3D GPU makes RK3036 completely compatible with OpenGL ES1.1 and 2.0, OpenVG1.1 etc.

RK3036 has high-performance external memory interface (DDR3/DDR3L) capable of sustaining demanding memory bandwidths, also provides a complete set of peripheral interface to support very flexible applications.

1.1 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.

1.1.1 Micro Processor

- Dual-core ARM Cortex-A7 MPCore processor, a high-performance, low-power and cached application processor
- Full implementation of the ARM architecture v7-A instruction set, ARM Neon Advanced SIMD (single instruction, multiple data) support for accelerated media and signal processing computation
- Superscalar, variable length, out-of-order pipeline with dynamic branch prediction, 8-stage pipeline
- Include VFP v3 hardware to support single and double-precision add, subtract, divide, multiply and accumulate, and square root operations
- SCU ensures memory coherency between the two CPUs
- Integrated 32KB L1 instruction cache , 32KB L1 data cache with 4-way set associative
- 128KByte unified L2 Cache

1.1.2 Internal Memory

- Internal BootRom
 - Size : 16KB
 - Support system boot from the following device:
 - ◆ 8bits Asynchronous Nand Flash
 - ◆ SPI Nand Flash
 - ◆ SPI Nor Flash
 - ◆ eMMC card
 - ◆ SD card
 - Support system code download by the following interface:
 - ◆ USB OTG interface
- Internal SRAM
 - Size : 8KB

1.1.3 External Memory or Storage device

- Dynamic Memory Interface (DDR3-1066/DDR3L-1066)
 - Compatible with JEDEC standard DDR3/DDR3L SDRAM
 - Support up to 2 ranks (chip selects), maximum 2GB address space
 - data width is 8bits or 16bits, software programmable
 - Four host ports with 64bits/128bits AXI bus interface for system access, AXI bus clock

- is asynchronous with DDR clock
- Programmable timing parameters to support DDR3/DDR3L SDRAM from various vendor
- Advanced command reordering and scheduling to maximize bus utilization
- Low power modes, such as power-down and self-refresh for DDR3 /DDR3L SDRAM
- Programmable output and ODT impedance with dynamic PVT compensation
- Support one low-power work mode: power down DDR PHY and most of DDR IO except two cs and cke output signals, make SDRAM still in self-refresh state to prevent data missing
- Nand Flash Interface
 - Support asynchronous nand flash, each channel 8bits, 1 bank
 - Support configurable interface timing
 - Embedded special DMA interface to do data transfer
- eMMC Interface
 - Compatible with standard iNAND interface
 - Support MMC4.5 protocol
 - Provide eMMC boot sequence to receive boot data from external eMMC device
 - Embedded clock adjustment for high speed transfer
 - Support 8bits data width with signal data rate or dual data rate
- SD/MMC Interface
 - Compatible with SD3.0, MMC4.5
 - Support FIFO over-run and under-run prevention by stopping card clock automatically
 - Embedded clock adjustment for high speed transfer
 - Support 4bits data width signal data rate or dual data rate
- Serial Flash Controller(SFC)
 - One on-chip SFC
 - Support command/address/data x1/x2/x4 bits mode configurable
 - Support 2 chip select
 - Support connected to serial nor flash or serial nand flash

1.1.4 System Component

- CRU (clock & reset unit)
 - Support clock gating control for individual components
 - One oscillator with 24MHz clock input and three embedded PLLs: ARM PLL, DDR PLL, GENERAL PLL
 - Support global soft-reset control for whole SOC, also individual soft-reset for every components
- Timer
 - Four on-chip 64bits Timers in SoC with interrupt-based operation
 - Provide two operation modes: free-running and user-defined count
 - Support timer work state checkable
- PWM
 - Four on-chip PWMs with interrupt-based operation
 - Programmable pre-scaled operation to input clock and then further scaled
 - 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
 - 32 bits watchdog counter width

- Counter clock is from APB bus clock
- 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
- Bus Architecture
 - 128bit/64-bit/32-bit multi-layer AXI/AHB/APB composite bus architecture
 - Five embedded AXI interconnect:
 - ◆ BUS interconnect
 - ◆ PERI interconnect
 - ◆ Display interconnect
 - ◆ GPU interconnect
 - ◆ VCODEC interconnect
 - For each interconnect with AXI/AHB/APB composite bus, clocks for AXI/AHB/APB domains are always synchronous, and different integer ratio is supported for them.
 - Flexible different QoS solution to improve the utility of bus bandwidth
- Interrupt Controller
 - Support 3 PPI interrupt source and 128 SPI interrupt sources input from different components
 - Support 16 software-triggered interrupts
 - Input interrupt level is fixed, only high-level sensitive
 - Support different interrupt priority for each interrupt source, and they are always software-programmable
- DMAC
 - Micro-code programming based DMA
 - The specific instruction set provides flexibility for programming DMA transfers
 - Linked list DMA function is supported to complete scatter-gather transfer
 - Support internal instruction cache
 - Embedded DMA manager thread
 - 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
 - dmac configuration:
 - ◆ 8 channels totally
 - ◆ 13 hardware request from peripherals
 - ◆ 2 interrupt output

1.1.5 Video CODEC

- Embedded memory management unit(MMU)
- Video Decoder
 - Real-time video decoder of MPEG-1, MPEG-2, MPEG-4, H.263, H.264, MVC
 - Error detection and concealment support for all video formats
 - Output data format is YUV420 semi-planar, and YUV400(monochrome) is also supported for H.264
 - H.264 up to HP level 4.2 : 1080p@30fps (1920x1088)[®]
 - H.265 up to main 4.1 : 1080p@30fps (1920x1088)
 - MPEG-4 up to ASP level 5 : 1080p@30fps (1920x1088)
 - MPEG-2 up to MP : 1080p@30fps (1920x1088)
 - MPEG-1 up to MP : 1080p@30fps (1920x1088)

- H.263 : 576p@30fps (720x576)
- MVC : 1080p@30fps (1920x1088)
- For H.264, image cropping not supported
- For MPEG-4, GMC(global motion compensation) not supported
- For MPEG-4 SP/H.263, using a modified H.264 in-loop filter to implement deblocking filter in post-processor unit

1.1.6 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[®] is up to 76million pixels per second
 - Embedded memory management unit(MMU)

1.1.7 Image Enhancement

- Image Post-Processor (embedded inside video decoder)
 - Combined with HD video decoder and JPEG decoder, post-processor can read input data directly from decoder output to reduce bus bandwidth
 - Also work as a stand-alone mode, its input data is from image data stored in external memory
 - Input data format:
 - ◆ Any format generated by video decoder in combined mode
 - ◆ YCbCr 4:2:0 semi-planar
 - ◆ YCbCr 4:2:0 planar
 - ◆ YCbYCr 4:2:2
 - ◆ YCrYCb 4:2:2
 - ◆ CbYCrY 4:2:2
 - ◆ CrYCbY 4:2:2
 - Output data format:
 - ◆ YCbCr 4:2:0 semi-planar
 - ◆ YCbYCr 4:2:2
 - ◆ YCrYCb 4:2:2
 - ◆ CbYCrY 4:2:2
 - ◆ CrYCbY 4:2:2
 - ◆ Fully configurable ARGB channel lengths and locations inside 32bits, such as ARGB8888, RGB565, ARGB4444 etc.
 - Input image size:
 - ◆ Combined mode: from 48x48 to 8176x8176 (66.8Mpixels)
 - ◆ Stand-alone mode: width from 48 to 8176, height from 48 to 8176, and maximum size limited to 16.7Mpixels
 - ◆ Step size is 16 pixels
 - Output image size: from 16x16 to 1920x1088 (horizontal step size 8, vertical step size 2)
 - Support image up-scaling:
 - ◆ Bicubic polynomial interpolation with a four-tap horizontal kernel and a two-tap vertical kernel
 - ◆ Arbitrary non-integer scaling ratio separately for both dimensions
 - ◆ Maximum output width is 3x input width
 - ◆ Maximum output height is 3x input height
 - Support image down-scaling:
 - ◆ Arbitrary non-integer scaling ratio separately for both dimensions
 - ◆ Unlimited down-scaling ratio
 - Support YUV to RGB color conversion, compatible with BT.601-5, BT.709 and user

- definable conversion coefficient
- Support dithering (Allegro dithering algorithm) for 4/5/6bit RGB channel precision
- Support RGB image contrast/brightness/color saturation adjustment
- Support image cropping & digital zoom only for JPEG or stand-alone mode
- Support picture in picture
- Support image rotation (horizontal flip, vertical flip, rotation 90, 180 or 270 degrees)

1.1.8 Graphics Engine

- 3D Graphics Engine(GPU)
 - High performance OpenGL ES1.1 and 2.0, OpenVG1.1 etc.
 - Embedded 1 shader core with shared hierarchical tiler
 - Separate vertex(geometry) and fragment(pixel) processing for maximum parallel throughput
 - Provide MMU and L2 Cache with 64KB size

1.1.9 Video OUT

- Display Interface
 - Display process
 - ◆ Background layer
 - programmable 24-bit color
 - ◆ Win0 (Video) layer
 - RGB888, ARGB888, RGB565, YCbCr422, YCbCr420, YCbCr444
 - Support virtual display
 - 1/8 to 8 scaling-down and scaling-up engine
 - ✧ Scale up using bicubic or bilinear
 - ✧ Scale down using bilinear
 - ✧ 1 Bicubic table: catrom
 - ✧ coord 8bit, coe 8bit signed
 - ◆ Win1 (UI) layer:
 - RGB888, ARGB888, RGB565
 - Support virtual display
 - ◆ Hardware cursor:
 - 8bpp
 - Support two sizes: 32x32,64x64
 - ◆ Overlay:
 - Win0/Win1 256 level alpha blending (support pre-multiplied alpha)
 - Win0/Win1 overlay position exchangeable
 - Win0/Win1 Transparency color key
 - Win0/Win1 global/per-pixel alpha
 - HWC 256 level alpha blending
 - HWC global/per-pixel alpha
 - Others
 - ◆ Max output resolution: 1920x1080
 - ◆ Max scanning timing: 4096x4096
 - ◆ YcbCr2RGB(rec601-mpeg/rec601-jpeg/rec709)and RGB2YcbCr
 - ◆ Support BCSH function
 - ◆ QoS request signals
 - ◆ Bus address alignment
 - ◆ Embedded memory management unit(MMU)
 - HDMI TX
 - HDMI version 1.4a, HDCP revision 1.2 and DVI version 1.0 compliant transmitter
 - Supports DTV from 480i to 1080i/p HD resolution
 - Supports 3D function defined in HDMI 1.4 spec
 - Supports data rate up to 3.4Gbps over a Single channel HDMI
 - TMDS Tx Drivers with programmable output swing, register values and pre-emphasis

- Digital video interface supports a pixel size of 24 color depth in RGB
 - S/PDIF output supports PCM, Dolby Digital, DTS digital audio transmission (32-192kHz Fs) using IEC60958 and IEC61937
 - Multiphase 4MHz fixed bandwidth PLL with low jitter
 - HDCP encryption and decryption engine contains all the necessary logic to encrypt the incoming audio and video data
 - Support HDMI LipSync if needed
 - Lower power operation with optimal power management feature
 - The EDID and CEC function are also supported
 - Optional Monitor Detection supported through Hot Plug
 - Support 8-channel I2S and 8-channel SPDIF input
- Video CVBS output
 - 10-bit Resolution
 - PAL/NTSC encoding
 - Programmable luma filter coefficient
 - Programmable luma/chroma delay
 - Programmable brightness/contrast

1.1.10 Audio Interface

- I2S/PCM with 8ch
 - Up to 8 channels (4xTX, 1xRX)
 - 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
 - Up to 8 channels
 - 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 bits audio data transfer in linear PCM mode
 - Support non-linear PCM transfer
- Audio DAC
 - 24 bit audio DAC and line out buffer
 - Support Mono channel performance
 - Digital interpolation and decimation filter integrated
 - Sampling rate of 8kHz/12kHz/16kHz/24kHz/32kHz/44.1KHz/48KHz/96KHz
 - Optional fractional PLL available that support 6MHz to 20MHz clock input to any clock

1.1.11 Connectivity

- SDIO interface
 - Embedded one SDIO interface
 - Compatible with SDIO 3.0 protocol
 - 4bits data bus widths
- EMAC 10/100M Ethernet Controller
 - IEEE802.3u compliant Ethernet Media Access Controller(MAC)
 - Support only RMII(Reduced MII) mode
 - 10Mbps and 100Mbps compatible
 - Automatic retry and automatic collision frame deletion
 - Full duplex support with flow-control

- Address filtering(broadcast, multicast, logical, physical)
- SPI Controller
 - One on-chip SPI controller
 - Support serial-master and serial-slave mode, software-configurable
 - DMA-based or interrupt-based operation
 - Support 2 chip-selects output in serial-master mode
- UART Controller
 - Three on-chip UART controller
 - DMA-based or interrupt-based operation
 - For all UART, UART0 is 64Bytes FIFOs, and the others are with 32Bytes FIFOs
 - Support 5bit,6bit,7bit,8bit 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
- I2C controller
 - Three on-chip I2C controller
 - Multi-master I2C operation
 - Support 7bits and 10bits address mode
 - Software programmable clock frequency and transfer rate up to 400Kbit/s in the fast mode
 - Serial 8bits oriented and bidirectional data transfers can be made at up to 100Kbit/s in the standard mode
- GPIO
 - Totally 70 GPIOs
 - All of GPIOs can be used to generate interrupt to Cortex-A7
 - The pull direction(pullup or pulldown) for all of GPIOs are software-programmable
 - All of GPIOs are always in input direction in default after power-on-reset
 - The drive strength is not software-programmable, determined by the GPIO type
- USB Host2.0
 - Compatible with USB Host2.0 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
- USB OTG2.0
 - Compatible with USB OTG2.0 specification
 - Supports high-speed(480Mbps), full-speed(12Mbps) and low-speed (1.5Mbps) mode
 - Support up to 9 device mode endpoints in addition to control endpoint 0
 - Support up to 6 device mode IN endpoints including control endpoint 0
 - Endpoints 1/3/5/7 can be used only as data IN endpoint
 - Endpoints 2/4/6 can be used only as data OUT endpoint
 - Endpoints 8/9 can be used as data OUT and IN endpoint
 - Provides 9 host mode channels

1.1.12 Others

- eFuse
 - One 256bits high-density electrical Fuse, organized as 32x8bits
 - Program 1-bit each time in program mode
 - Read 8-bit each time in read mode
 - Three operation modes: program mode, read mode, inactive mode

- Package Type
 - eLQFP176 (body: 20mm x 20mm; pin pitch: 0.4mm)
 - BGA236(body: 12mm x 12mm; pin pitch: 0.65mm)

Notes:

- ① Actual maximum frame rate will depend on the clock frequency and system bus performance
- ④ Actual maximum data rate will depend on the clock frequency and JPEG compression rate

1.2 Block Diagram

The following diagram shows the basic block diagram for RK3036.

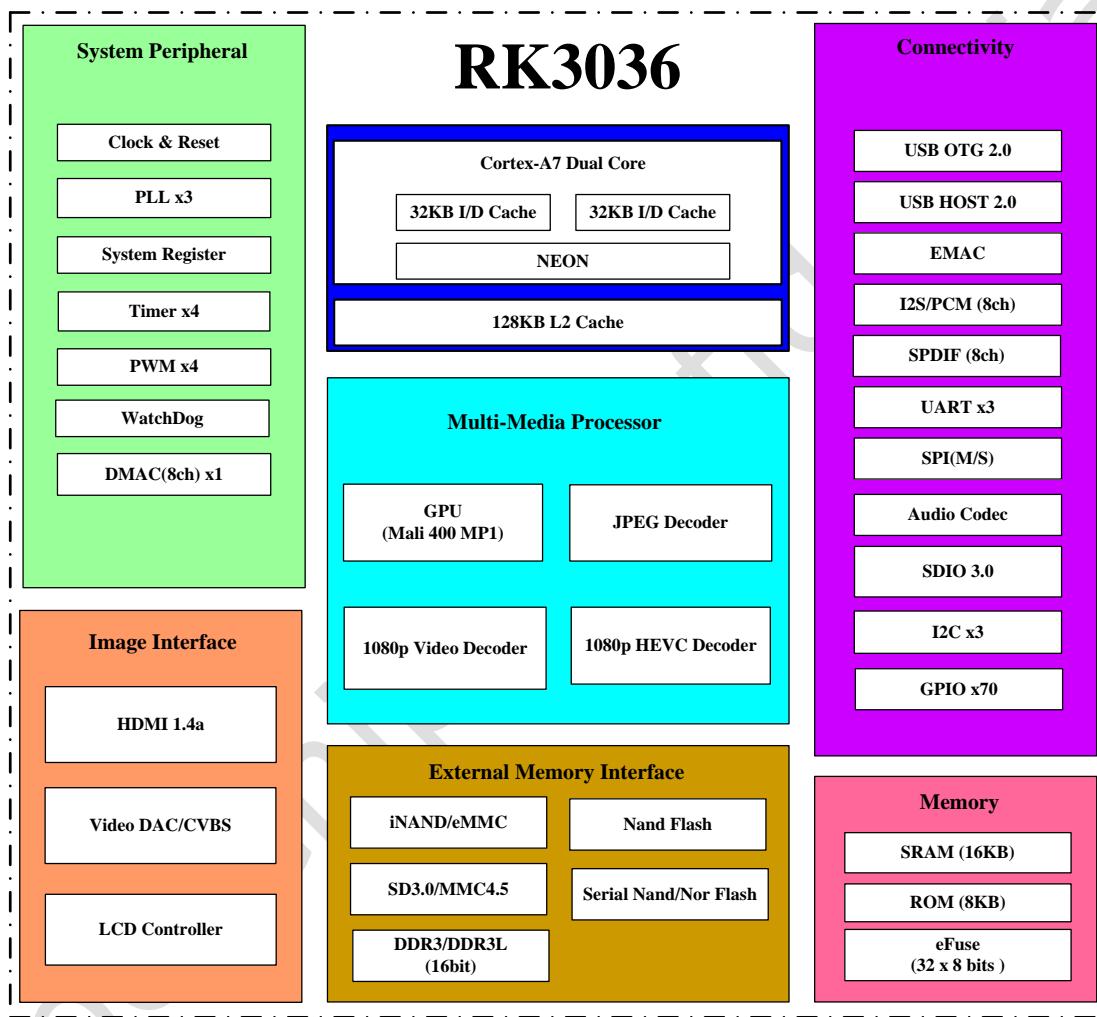


Fig. 1-1 RK3036 Block Diagram

Chapter 2 System Overview

2.1 Address Mapping

RK3036 supports to boot from internal bootrom, which supports remap function by software programming. Remap is controlled by GRF_SOC_CON0[12]. When remap is set to 1, the bootrom is mapped to address 0x10100000 and internal memory is mapped to address 0x0.

Addr	IP	Addr	IP	Addr	IP	Addr	IP
1013e000	Reserved	20038000		10504000		20094000	
1013d000	Reserved 4K	20034000	HDMI-ANA 16K	10500000	NANDC 16K	20090000	eFuse 16K
1013c000	Reserved 8k	20030000	ACODEC-ANA 16K	10400000	Reserved 1024K	2008C000	16K
10138000	GIC 16K	20020000	DBG 64K	10300000	PERI BUS 1024k	20088000	reserved 16K
10130000	Reserved 32K	20010000	Reserved 64K	1023C000	AHB ARB1 784K	20084000	GPIO2 16K
10128000	CPU BUS 32K	2000a000	DDR_PHY 24K	10234000	AHB ARBO 32K	20080000	GPIO1 16K
10124000	Reserved 16K	20008000	GRF 8K	10224000	Reserved 64K	2007C000	GPIO 16K
10120000	Reserved 16k	20004000	DDR_PCTL 16K	10220000	I2S 16K	20078000	DMAC 16K
1011c000	Reserved 8K	20000000	CRU 16K	1021C000	eMMC 16K	20074000	SPI 16K
10118000	VOP 16K			10218000	SDIO 16K	20072000	I2C0 8K
10114000	Reserved 16K			10214000	SDMMC 16K	2006C000	Reserved 24K
10110000	Reserved 16K			1020C000	Reserved 64k	20068000	UART2 16K
1010c000	Hevc 16K			10208000	SFC 16k	20064000	UART1 16K
10108000	Vcodec 16K			10204000	SPDIF 16K	20060000	UART0 16K
10104000	Reserved 16K			10200000	EMAC 16K	2005C000	reserved 16K
10100000	ROM 16K			101C0000	USB HOST 256K	2005a000	I2C2 8K
100a0000	Reserved 384K			10180000	USB OTG 256K	20056000	I2C1 16K
10090000	GPU 64K					20050000	PWM 24K
10082000	Reserved 56K					2004C000	WDT 16K
10080000	IMEM 8K					20048000	Reserved 16K

Fig. 2-1 RK3036 Address Mapping

2.2 System Boot

RK3036 provides system boot from off-chip devices such as SDMMC card, eMMC memory, 8bits asynchronous nand flash, serial nand or nor flash. 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:
 - Asynchronous Nand Flash, 8bits data width
 - Serial Nand Flash/SPI Nand Flash, 1bit data width
 - Serial Nor Flash, 1bit data width
 - eMMC Interface, 8bits data width
 - SDMMC Card, 4bits data width
- Support system code download by USB OTG

Following figure shows RK3036 boot procedure flow.

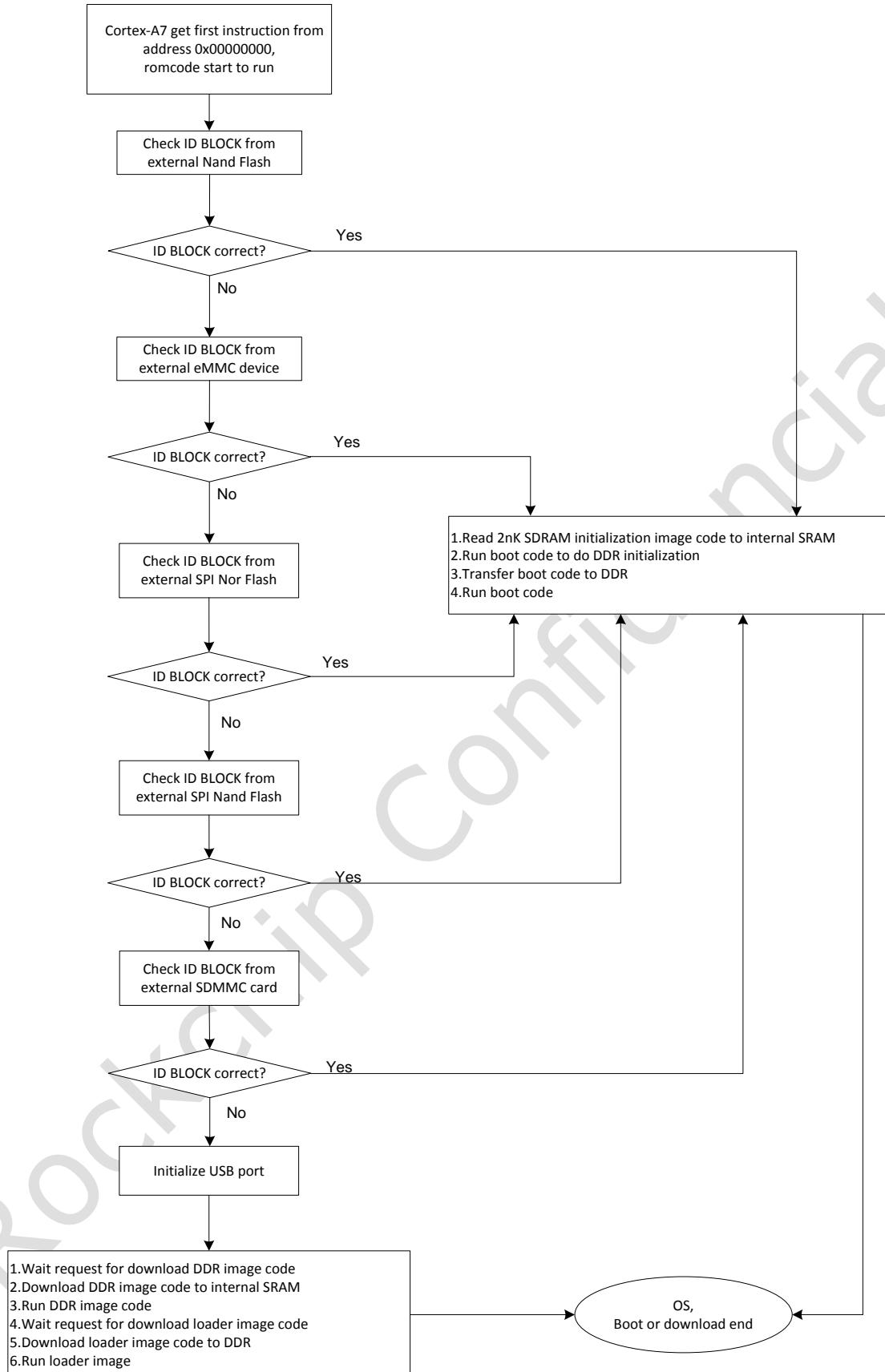


Fig. 2-2 RK3036 boot procedure flow

2.3 System Interrupt connection

RK3036 provides an general interrupt controller(GIC) for CPU, which has 128 SPI (shared peripheral interrupts) interrupt sources and 3 PPI(Private peripheral interrupt) interrupt source and separately generates one nIRQ and one nFIQ to CPU. The triggered type for each

interrupts is high level sensitive, not programmable. The detailed interrupt sources connection is in the following table. For detailed GIC setting, please refer to Chapter 9.

Table 2-1 RK3036 Interrupt connection list

IRQ Type	IRQ ID	Source(spi)	Polarity
PPI	27	Golbal Timer	High level
	29	Private Timer	High level
	30	WDT	High level
SPI	32	DMAC2(0)	High level
	33	DMAC2(1)	High level
	34	DDR_PCTL	High level
	35	gpu_irqgp	High level
	36	gpu_irqmmu	High level
	37	gpu_irqpp	High level
	38	Video encoder	High level
	39	Video decoder	High level
	40	EMAC	High level
	41	SFC	High level
	42	USB OTG	High level
	43	USB Host2.0	High level
	44	Reserved	High level
	45	Reserved	High level
	46	SD/MMC0	High level
	47	SDIO	High level
	48	eMMC	High level
	49	Reserved	High level
	50	NandC	High level
	51	I2S	High level
	52	UART0	High level
	53	UART1	High level
	54	UART2	High level
	55	SPI0	High level
	56	I2C0	High level
	57	I2C1	High level
	58	I2C2	High level
	59	Reserved	High level
	60	Timer0	High level
	61	Timer1	High level
	62	PWM	High level
	63	Reserved	High level
	64	Reserved	High level
	65	Reserved	High level
	66	WDT	High level
	67	otg_bvalid_irq	High level
	68	GPIO0	High level
	69	GPIO1	High level

IRQ Type	IRQ ID	Source(spi)	Polarity
	70	GPIO2	High level
	71	Reserved	High level
	72	Reserved	High level
	73	Reserved	High level
	74	peri_ahb_usb arbiter	High level
	75	VOP	High level
	76	Reserved	High level
	77	hdmi	High level
	78	SD/MMC detect	High level
	79	SDIO detect	High level
	80	Reserved	High level
	81	Reserved	High level
	82	Reserved	High level
	83	otg0_id_irq	High level
	84	otg0_linestate_irq	High level
	85	otg1_linestate_irq	High level
	86	sd_detectn_irq	High level
	87	VPU_mmu_irq	High level
	88	Hevc_mmu_irq	High level
	89	Hevc_dec_irq	High level
	90	Timer2	High level
	91	Timer3	High level
	92	SPDIF	High level
	93	Reserved	High level
	94	Reserved	High level
	95	Reserved	High level
	96	Reserved	High level
	97	Reserved	High level
	98	Reserved	High level
	99	Reserved	High level
	100	Reserved	High level
	101	Reserved	High level
	102	Reserved	High level
	103	Reserved	High level
	104	Reserved	High level
	105	Reserved	High level
	106	Reserved	High level
	107	Reserved	High level
	108	core_npmuirq0	High level
	109	core_npmuirq1	High level
	110	core_npmuirq2	High level
	111	core_npmuirq3	High level
	112	axierrirq	High level
	113~159	Reserved	High level

2.4 System DMA hardware request connection

RK3036 provides one DMA controller: DMAC_PERI inside peripheral system. As for DMAC_PERI, there are 14 hardware request ports. The trigger type for each of them is high level, not programmable. For detailed descriptions of DMAC_PERI, please refer to Chapter 8.

Table 2-2 RK3036 DMAC_PERI Hardware request connection list

Req Number	Source	Polarity
0	I2S tx	High level
1	I2S rx	High level
2	Uart0 tx	High level
3	Uart0 rx	High level
4	Uart1 tx	High level
5	Uart1 rx	High level
6	Uart2 tx	High level
7	Uart2 rx	High level
8	SPI tx	High level
9	SPI rx	High level
10	SDMMC	High level
11	SDIO	High level
12	EMMC	High level
13	SPDIF tx	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 clocks 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 three PLLs
- 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

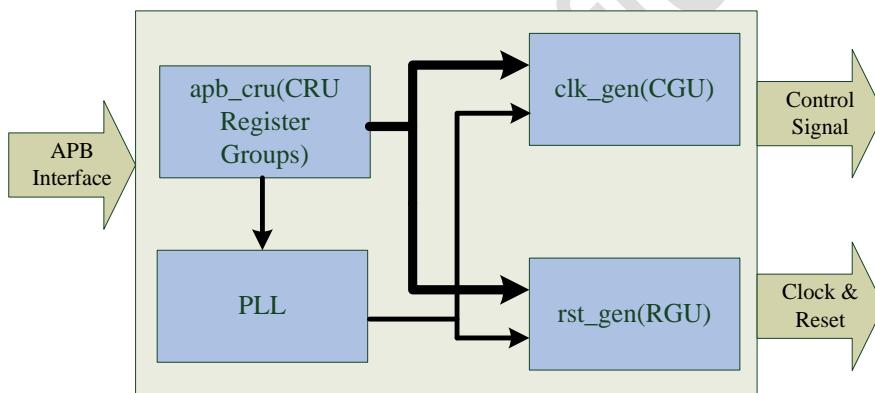


Fig. 3-1 CRU Architecture

3.3 System Reset Solution

The following diagram shows reset architecture.

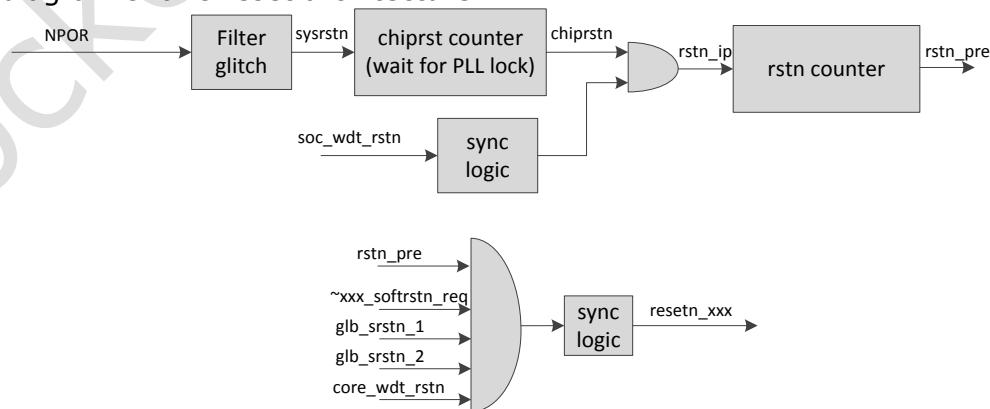


Fig. 3-2 Reset Architecture Diagram

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 A7 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 A7 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 0xfd9, glb_srstn_1 will be asserted, and when writing register CRU_GLB_SRST_SND_VALUE as 0xea8, glb_srstn_2 will be asserted. The two software resets will be self-cleared by hardware. glb_srstn_1 will reset the all logic, and glb_srstn_2 will reset the all logic except GRF and all GPIOs.

3.4 Function Description

There are three PLLs in the chip: ARM PLL, DDR PLL and GENERAL PLL, and it supports only one crystal oscillator: 24MHz. Each PLL can only receive 24MHz oscillator.

Three 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 three PLLs.

To provide some specific frequency, another solution is integrated: fractional divider. In order to guarantee 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 resets can be software generated.

3.5 PLL Introduction

3.5.1 Overview

The chip uses 2.4GHz PLL for all three PLLs. The 2.4GHz 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, we can greatly simplify an 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.

3.5.2 Block diagram

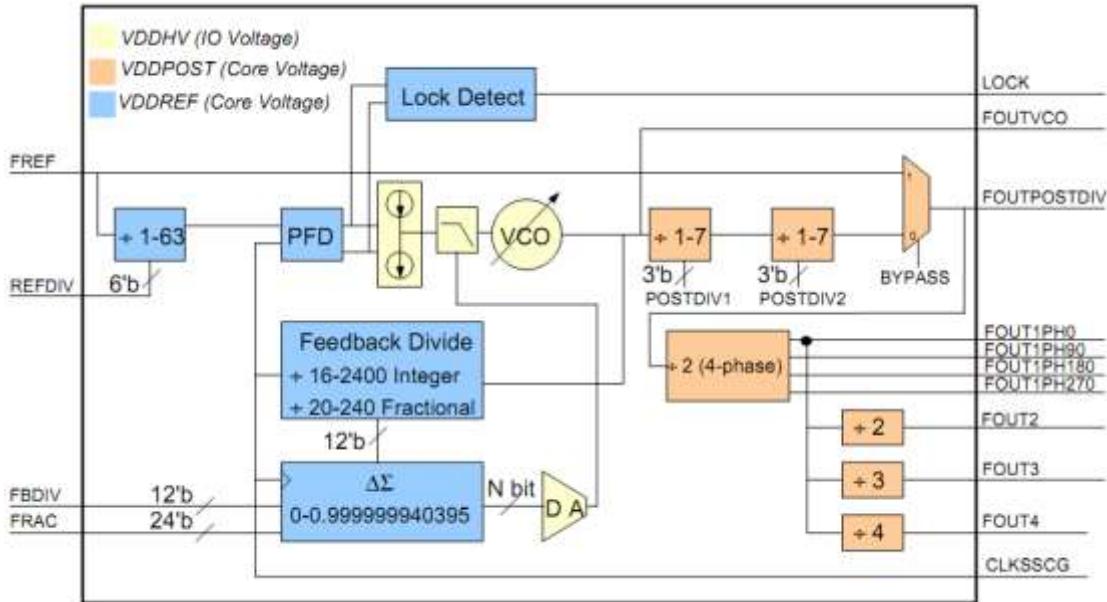


Fig. 3-3 PLL Block Diagram

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")

$$\text{FOUTVCO} = \text{FREF} / \text{REFDIV} * \text{FBDIV}$$

$$\text{FOUTPOSTDIV} = \text{FOUTVCO} / \text{POSTDIV1} / \text{POSTDIV2}$$

If DSMPD = 0 (DSM is enabled, "fractional mode")

$$\text{FOUTVCO} = \text{FREF} / \text{REFDIV} * (\text{FBDIV} + \text{FRAC} / 224)$$

$$\text{FOUTPOSTDIV} = \text{FOUTVCO} / \text{POSTDIV1} / \text{POSTDIV2}$$

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

3.6 Register Description

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

3.6.1 Registers Summary

Name	Offset	Size	Reset Value	Description
CRU_APPL_CON0	0x0000	W	0x000010af	ARM PLL control register 0
CRU_APPL_CON1	0x0004	W	0x00001046	ARM PLL control register 1

Name	Offset	Size	Reset Value	Description
CRU_APLL_CON2	0x0008	W	0x00000001	ARM PLL control register 2
CRU_DPLL_CON0	0x0010	W	0x00001064	DDR PLL control register 0
CRU_DPLL_CON1	0x0014	W	0x00001043	DDR PLL control register 1
CRU_DPLL_CON2	0x0018	W	0x00000001	DDR PLL control register 2
CRU_GPLL_CON0	0x0030	W	0x00004063	General PLL control register 0
CRU_GPLL_CON1	0x0034	W	0x00001042	General PLL control register 1
CRU_GPLL_CON2	0x0038	W	0x00f5c28f	General PLL control register 2
CRU_MODE_CON	0x0040	W	0x00000000	System work mode control register
CRU_CLKSEL0_CON	0x0044	W	0x00000200	Internal clock select and divide register 0
CRU_CLKSEL1_CON	0x0048	W	0x00003113	Internal clock select and divide register 1
CRU_CLKSEL2_CON	0x004c	W	0x00000000	Internal clock select and divide register 2
CRU_CLKSEL3_CON	0x0050	W	0x00008200	Internal clock select and divide register 3
CRU_CLKSEL5_CON	0x0058	W	0x00000800	Internal clock select and divide register 5
CRU_CLKSEL7_CON	0x0060	W	0x0bb8ea60	Internal clock select and divide register 7
CRU_CLKSEL9_CON	0x0068	W	0x0bb8ea60	Internal clock select and divide register 9
CRU_CLKSEL10_CON	0x006c	W	0x0000a100	Internal clock select and divide register 10
CRU_CLKSEL11_CON	0x0070	W	0x00001717	Internal clock select and divide register 11
CRU_CLKSEL12_CON	0x0074	W	0x00002a17	Internal clock select and divide register 12
CRU_CLKSEL13_CON	0x0078	W	0x00000200	Internal clock select and divide register 13
CRU_CLKSEL14_CON	0x007c	W	0x00000200	Internal clock select and divide register 14
CRU_CLKSEL15_CON	0x0080	W	0x00000200	Internal clock select and divide register 15
CRU_CLKSEL16_CON	0x0084	W	0x00000202	Internal clock select and divide register 16
CRU_CLKSEL17_CON	0x0088	W	0x0bb8ea60	Internal clock select and divide register 17
CRU_CLKSEL18_CON	0x008c	W	0x0bb8ea60	Internal clock select and divide register 18
CRU_CLKSEL19_CON	0x0090	W	0x0bb8ea60	Internal clock select and divide register 19
CRU_CLKSEL20_CON	0x0094	W	0x00000002	Internal clock select and divide register 20

Name	Offset	Size	Reset Value	Description
CRU_CLKSEL21_CON	0x0098	W	0x00000012	Internal clock select and divide register 21
CRU_CLKSEL25_CON	0x00a8	W	0x00000207	Internal clock select and divide register 25
CRU_CLKSEL26_CON	0x00ac	W	0x00000000	Internal clock select and divide register 26
CRU_CLKSEL28_CON	0x00b4	W	0x00000702	Internal clock select and divide register 28
CRU_CLKSEL30_CON	0x00bc	W	0x00008300	Internal clock select and divide register 30
CRU_CLKSEL31_CON	0x00c0	W	0x00008000	Internal clock select and divide register 31
CRU_CLKSEL32_CON	0x00c4	W	0x00008000	Internal clock select and divide register 32
CRU_CLKSEL34_CON	0x00cc	W	0x00000200	Internal clock select and divide register 34
CRU_CLKGATE0_CON	0x00d0	W	0x00000000	Internal clock gating control register 0
CRU_CLKGATE1_CON	0x00d4	W	0x00000000	Internal clock gating control register 1
CRU_CLKGATE2_CON	0x00d8	W	0x00000000	Internal clock gating control register 2
CRU_CLKGATE3_CON	0x00dc	W	0x00000000	Internal clock gating control register 3
CRU_CLKGATE4_CON	0x00e0	W	0x00000000	Internal clock gating control register 4
CRU_CLKGATE5_CON	0x00e4	W	0x00000000	Internal clock gating control register 5
CRU_CLKGATE6_CON	0x00e8	W	0x00000000	Internal clock gating control register 6
CRU_CLKGATE7_CON	0x00ec	W	0x00000000	Internal clock gating control register 7
CRU_CLKGATE8_CON	0x00f0	W	0x00000000	Internal clock gating control register 8
CRU_CLKGATE9_CON	0x00f4	W	0x00000000	Internal clock gating control register 9
CRU_CLKGATE10_CON	0x00f8	W	0x00000000	Internal clock gating control register 10
CRU_GLB_SRST_FST_VALUE	0x0100	W	0x00000000	The first global software reset config value
CRU_GLB_SRST SND_VAL	0x0104	W	0x00000000	The second global software reset config value
CRU_SOFRST0_CON	0x0110	W	0x00000000	Internal software reset control register 0

Name	Offset	Size	Reset Value	Description
CRU_SOFRST1_CON	0x0114	W	0x00000000	Internal software reset control register 1
CRU_SOFRST2_CON	0x0118	W	0x00000000	Internal software reset control register 2
CRU_SOFRST3_CON	0x011c	W	0x00000000	Internal software reset control register 3
CRU_SOFRST4_CON	0x0120	W	0x00000000	Internal software reset control register 4
CRU_SOFRST5_CON	0x0124	W	0x00000000	Internal software reset control register 5
CRU_SOFRST6_CON	0x0128	W	0x00000000	Internal software reset control register 6
CRU_SOFRST7_CON	0x012c	W	0x00000000	Internal software reset control register 7
CRU_SOFRST8_CON	0x0130	W	0x00000000	Internal software reset control register 8
CRU_MISC_CON	0x0134	W	0x00000000	SCU control register
CRU_GLB_CNT_TH	0x0140	W	0x3a980064	global reset wait counter threshold
CRU_SDMMC_CON0	0x0144	W	0x00000004	SDMMC clock generation control register 0
CRU_SDMMC_CON1	0x0148	W	0x00000000	SDMMC clock generation control register 1
CRU_SDIO_CON0	0x014c	W	0x00000004	SDIO clock generation control register 0
CRU_SDIO_CON1	0x0150	W	0x00000000	SDIO clock generation control register 1
CRU_EMMC_CON0	0x0154	W	0x00000004	EMMC clock generation control register 0
CRU_EMMC_CON1	0x0158	W	0x00000000	EMMC clock generation control register 1
CRU_RST_ST	0x0160	W	0x00000000	WDT and global reset status
CRU_PLL_MASK_CON	0x01f0	W	0x00005a5a	PLL mask control register

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

3.6.2 Detail Register Description

CRU_APPL_CON0

Address: Operational Base + offset (0x0000)

ARM PLL control register 0

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	bp PLL bypass
14:12	RW	0x1	postdiv1 PLL factor postdiv1
11:0	RW	0x0af	fbdv PLL factor fbdv

CRU_APPL_CON1

Address: Operational Base + offset (0x0004)

ARM PLL control register 1

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_APPL_CON2

Address: Operational Base + offset (0x0008)

ARM PLL control register 2

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27	RW	0x0	fout4phasepd 4 phase clock power down, active high

Bit	Attr	Reset Value	Description
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 register 0

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	0x064	fbdv PLL factor fbdv

CRU_DPLL_CON1

Address: Operational Base + offset (0x0014)

DDR PLL control register 1

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

Bit	Attr	Reset Value	Description
12	RW	0x1	dsmpd when 1, PLL work at integer 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 register 2

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 register 0

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	fbdv PLL factor fbdv

CRU_GPLL_CON1

Address: Operational Base + offset (0x0034)

General PLL control register 1

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 integer 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 register 2

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	0xf5c28f	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 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
13:12	RW	0x0	gpll_work_mode General PLL work mode select 2'b00: Slow mode, clock from external 24MHz OSC (default) 2'b01: Normal mode, clock from PLL output 2'b10: Deep slow mode, clock from external 32.768kHz or PVTM clk
11:5	RO	0x0	reserved
4	RW	0x0	dpll_work_mode DDR PLL work mode select 0: Slow mode, clock from external 24MHz OSC (default) 1: Normal mode, clock from PLL output
3:1	RO	0x0	reserved
0	RW	0x0	apll_work_mode ARM PLL work mode select 0: Slow mode, clock from external 24MHz OSC (default) 1: Normal mode, clock from PLL output

CRU_CLKSEL0_CON

Address: Operational Base + offset (0x0044)

Internal clock select and divide register 0

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	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	cpu_clk_pll_sel pd_cpu aclk_cpu pll source selection 2'b00: select ARM PLL 2'b01: select DDR PLL 2'b10: select GENERAL PLL
13	RO	0x0	reserved
12:8	RW	0x02	aclk_cpu_div_con aclk_cpu clock divider frequency $aclk_cpu = \text{cpu_clk_src} / (\text{aclk_cpu_div_con} + 1)$
7	RW	0x0	core_clk_pll_sel core clock pll source selection 1'b0: select ARM PLL 1'b1: select GENERAL PLL
6:5	RO	0x0	reserved

Bit	Attr	Reset Value	Description
4:0	RW	0x00	a7_core_div_con Control A7 core clock divider frequency $\text{clk_core} = \text{core_clk_src}/(\text{a7_core_div_con}+1)$

CRU_CLKSEL1_CON

Address: Operational Base + offset (0x0048)

Internal clock select and divide register 1

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	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 $\text{pclk_cpu} = \text{cpu_aclk_src}/(\text{cpu_pclk_div_con}+1)$
11:10	RO	0x0	reserved
9:8	RW	0x1	cpu_hclk_div_con Control cpu subsystem AHB clock divider frequency $\text{hclk_cpu} = \text{cpu_aclk_src}/(\text{cpu_hclk_div_con}+1)$
7	RO	0x0	reserved
6:4	RW	0x1	core_aclk_div_con Control A7 core axi clock divider frequency $\text{aclk_core} = \text{core_clk_src}/(\text{core_aclk_div_con}+1)$
3:0	RW	0x3	core_peri_div_con core periph div control $\text{clk_core_peri} = \text{clk_peri_src}/(\text{core_peri_div_con}+1)$

CRU_CLKSEL2_CON

Address: Operational Base + offset (0x004c)

Internal clock select and divide register 2

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask 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_timer3_sel timer3 clock select 1'b0: 24MHz 1'b1: pclk_peri
6	RW	0x0	clk_timer2_sel timer2 clock select 1'b0: 24MHz 1'b1: pclk_peri

Bit	Attr	Reset Value	Description
5	RW	0x0	clk_timer1_sel timer1 clock select 1'b0: 24MHz 1'b1: pclk_peri
4	RW	0x0	clk_timer0_sel timer0 clock select 1'b0: 24MHz 1'b1: pclk_peri
3:0	RO	0x0	reserved

CRU_CLKSEL3_CON

Address: Operational Base + offset (0x0050)

Internal clock select and divide register 3

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	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	i2s_pll_sel 2'b00: select ARM PLL 2'b01: select DDR PLL 2'b10: select GENERAL PLL
13	RO	0x0	reserved
12	RW	0x0	i2s1_out_sel 1'b0: select cru generated clock 1'b1: select io input clock
11:10	RO	0x0	reserved
9:8	RW	0x2	i2s_clk_sel Control I2S clock work frequency selection 2'b00: select divider output from pll divider 2'b01: select divider output from fraction divider 2'b10: select io input clock 2'b11: select 12MHz from osc input
7	RO	0x0	reserved
6:0	RW	0x00	i2s_pll_div_con Control I2S PLL output divider frequency $i2s1_div_clk = i2s1_div_src / (i2s1_pll_div_con + 1)$

CRU_CLKSEL5_CON

Address: Operational Base + offset (0x0058)

Internal clock select and divide register 5

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	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

Bit	Attr	Reset Value	Description
11:10	RW	0x2	spdif_pll_sel 2'b00: select ARM PLL 2'b01: select DDR PLL 2'b10: select GENERAL PLL
9:8	RW	0x0	spdif_clk_sel 2'b00: select divider output from pll divider 2'b01: select divider output from fraction divider 2'b10: select 12MHz from osc input
7	RO	0x0	reserved
6:0	RW	0x00	spdif_pll_div_con clk_spdif_div_out=clk_spdif_src/(spdif_pll_div_con+1)

CRU_CLKSEL7_CON

Address: Operational Base + offset (0x0060)

Internal clock select and divide register 7

Bit	Attr	Reset Value	Description
31:0	RW	0xbb8ea60	i2s_frac_factor Control I2S 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 register 9

Bit	Attr	Reset Value	Description
31:0	RW	0xbb8ea60	spdif_frac_factor Control I2S fraction divider frequency High 16-bit for numerator Low 16-bit for denominator

CRU_CLKSEL10_CON

Address: Operational Base + offset (0x006c)

Internal clock select and divide register 10

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	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 ARM PLL 2'b01: select DDR PLL 2'b10: select GENERAL PLL

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 peripheral 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 register 11

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	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
14:8	RW	0x17	sdio_div_con Control SDIO divider frequency clk_sdio0=general_pll_clk/(mmc0_div_con+1)
7	RO	0x0	reserved
6:0	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 register 12

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	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

Bit	Attr	Reset Value	Description
13:12	RW	0x2	emmc_pll_sel Control emmc clock PLL source selection 2'b00: select ARM PLL 2'b01: select DDR PLL 2'b10: select GENERAL PLL 2'b11: select X24M
11:10	RW	0x2	sdio_pll_sel Control sdio clock PLL source selection 2'b00: select ARM PLL 2'b01: select DDR PLL 2'b10: select GENERAL PLL 2'b11: select X24M
9:8	RW	0x2	mmc0_pll_sel Control sdio clock PLL source selection 2'b00: select ARM PLL 2'b01: select DDR PLL 2'b10: select GENERAL PLL 2'b11: select X24M
7	RO	0x0	reserved
6:0	RW	0x17	emmc_div_con Control EMMC divider frequency $\text{clk_emmc} = \text{general_pll_clk} / (\text{emmc_div_con} + 1)$

CRU_CLKSEL13_CON

Address: Operational Base + offset (0x0078)

Internal clock select and divide register 13

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	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
11:10	RW	0x0	uart_pll_sel Control UART clock PLL source selection 2'b00: select ARM PLL 2'b01: select DDR PLL 2'b10: select GENERAL PLL 2'b11: select usb_480m clock
9:8	RW	0x2	uart0_clk_sel Control UART0 clock work frequency selection 2'b00: select divider output from pll divider 2'b01: select divider output from fraction divider 2'b10: select 24MHz from osc input
7	RO	0x0	reserved

Bit	Attr	Reset Value	Description
6:0	RW	0x00	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 register 14

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	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 output from pll divider 2'b01: select divider output from fraction divider 2'b10: select 24MHz from osc input
7	RO	0x0	reserved
6:0	RW	0x00	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 register 15

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	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 output from pll divider 2'b01: select divider output from fraction divider 2'b10: select 24MHz from osc input
7	RO	0x0	reserved
6:0	RW	0x00	uart2_div_con Control UART2 divider frequency clk_uart2=uart_clk_src/(uart2_div_con+1)

CRU_CLKSEL16_CON

Address: Operational Base + offset (0x0084)

Internal clock select and divide register 16

Bit	Attr	Reset Value	Description
31:15	RO	0x0	reserved

Bit	Attr	Reset Value	Description
14:10	RW	0x00	nandc_div_con clk_nandc_div_out=clk_nandc_src/(nandc_div_con+1)
9:8	RW	0x2	nandc_pll_sel 2'b00: select ARM PLL 2'b01: select DDR PLL 2'b10: select GENERAL PLL
7	RO	0x0	reserved
6:2	RW	0x00	sfc_div_con clk_sfc_div_out=clk_sfc_src/(sfc_div_con+1)
1:0	RW	0x2	sfc_pll_sel 2'b00: select ARM PLL 2'b01: select DDR PLL 2'b10: select GENERAL PLL 2'b11: select X24M

CRU_CLKSEL17_CON

Address: Operational Base + offset (0x0088)

Internal clock select and divide register 17

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 register 18

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 register 19

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 register 20

Bit	Attr	Reset Value	Description
31:7	RO	0x0	reserved
6:2	RW	0x00	hevc_div_con clk_hevc_div_out=clk_hevc_src/(hevc_div_con+1)
1:0	RW	0x2	hevc_pll_sel 2'b00: select ARM PLL 2'b01: select DDR PLL 2'b10: select GENERAL PLL

CRU_CLKSEL21_CON

Address: Operational Base + offset (0x0098)

Internal clock select and divide register 21

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	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
13:9	RW	0x00	mac_div_con clk_mac_div_out=clk_mac_src/(mac_div_con+1)
8:4	RW	0x01	macref_div_con clk_macref=clk_macreref_muxout/(macref_div_con+1)
3	RW	0x0	rmii_extclk_sel 0: clk_macref_div_out 1: rmii_clkin
2	RO	0x0	reserved
1:0	RW	0x2	mac_pll_sel 2'b00: select ARM PLL 2'b01: select DDR PLL 2'b10: select GENERAL PLL

CRU_CLKSEL25_CON

Address: Operational Base + offset (0x00a8)

Internal clock select and divide register 25

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	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	spi_clk_pll_sel SPI clock pll source selection 2'b00: select ARM PLL 2'b01: select DDR PLL 2'b10: select GENERAL PLL
7	RO	0x0	reserved

Bit	Attr	Reset Value	Description
6:0	RW	0x07	spi0_div_con Control SPI0 clock divider frequency $\text{clk_spi0} = \text{general_pll_clk}/(\text{spi0_div_con}+1)$

CRU_CLKSEL26_CON

Address: Operational Base + offset (0x00ac)

Internal clock select and divide register 26

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	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
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_CLKSEL28_CON

Address: Operational Base + offset (0x00b4)

Internal clock select and divide register 28

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	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	0x07	lc当地_dclk_div_con Control LCD display clock divider frequency $\text{dclk_lc当地1} = \text{lc当地_dclk_src}/(\text{lc当地_dclk_div_con}+1)$
7:2	RO	0x0	reserved
1:0	RW	0x2	lc当地_dclk_pll_sel Control LCD display clock PLL source selection 2'b00: select ARM PLL 2'b01: select DDR PLL 2'b10: select GENERAL PLL

CRU_CLKSEL30_CON

Address: Operational Base + offset (0x00bc)

Internal clock select and divide register 30

Bit	Attr	Reset Value	Description
-----	------	-------------	-------------

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	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	vio_hclk_pll_sel Control VIO AHB clock PLL source selection 2'b00: select ARM PLL 2'b01: select DDR PLL 2'b10: select GENERAL PLL
13	RO	0x0	reserved
12:8	RW	0x03	vio_hclk_div_con Control VIO AHB clock divider frequency $hclk_vio = vio_hclk_src / (vio_hclk_div_con + 1)$
7:5	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 register 31

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	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	lcdc_aclk_pll_sel Control LCDC AXI clock PLL source selection 2'b00: select ARM PLL 2'b01: select DDR PLL 2'b10: select GENERAL PLL
13	RO	0x0	reserved
12:8	RW	0x00	lcdc_aclk_div_con Control LCDC AXI clock divider frequency $aclk_{lcdc1} = lcdc1_aclk_src / (lcdc1_aclk_div_con + 1)$
7:1	RO	0x0	reserved
0	RW	0x0	hdmi_clk_sel 0: hdmi vclk_pllref from vop dclk; 1: hdmi vclk_pllref from cru dclk ;

CRU_CLKSEL32_CON

Address: Operational Base + offset (0x00c4)

Internal clock select and divide register 32

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	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	0x2	vdpu_aclk_pll_sel Control VDPU AXI clock PLL source selection 2'b00: select ARM PLL 2'b01: select DDR PLL 2'b10: select GENERAL PLL
13	RO	0x0	reserved
12:8	RW	0x00	vdpu_aclk_div_con Control VDPU AXI clock divider frequency $aclk_{vdpu} = aclk_{src} / (vdpu_aclk_div_con + 1)$
7:0	RO	0x0	reserved

CRU_CLKSEL34_CON

Address: Operational Base + offset (0x00cc)

Internal clock select and divide register 34

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	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	gpu_aclk_pll_sel Control GPU AXI clock PLL source selection 2'b00: select ARM PLL 2'b01: select DDR PLL 2'b10: select GENERAL PLL
7:5	RO	0x0	reserved
4:0	RW	0x00	gpu_aclk_div_con Control GPU AXI clock divider frequency $aclk_{gpu} = aclk_{src} / (gpu_aclk_div_con + 1)$

CRU_CLKGATE0_CON

Address: Operational Base + offset (0x00d0)

Internal clock gating control register 0

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	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_i2s1_en When HIGH, disable clock
13	RW	0x0	clk_i2s1_out_en When HIGH, disable clock

Bit	Attr	Reset Value	Description
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_frac_src_gate_en I2S fraction divider source clock disable. When HIGH, disable clock
9	RW	0x0	clk_i2s_src_gate_en I2S source clock disable. When HIGH, disable clock
8	RW	0x0	clk_ddrphy_src_gpll_gate_en DDR PHY clock(clk_ddrphy) gpll source disable. When HIGH, disable clock
7	RW	0x0	aclk_core_gate_en ARM core axi clock(aclk_core) disable. When HIGH, disable clock
6	RW	0x0	core_gpll_clk_gate_en CORE clock GPLL 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
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_gpll_clk_gate_en CPU clock GPLL path 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 register 1

Bit	Attr	Reset Value	Description
------------	-------------	--------------------	--------------------

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	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	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	RO	0x0	reserved
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	ack_lcdc_src_gate_en ack_lcdc_src clock disable. When HIGH, disable clock
3	RW	0x0	clk_jtag_gate_en JTAG clock disable. When HIGH, disable clock
2	RO	0x0	reserved
1	RW	0x0	clk_timer1_gate_en Timer1 clock(clk_timer1) disable. When HIGH, disable clock
0	RW	0x0	clk_timer0_gate_en Timer0 clock(clk_timer0) disable. When HIGH, disable clock

CRU_CLKGATE2_CON

Address: Operational Base + offset (0x00d8)

Internal clock gating control register 2

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	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_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_en 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_en When HIGH, disable clock
9	RW	0x0	clk_spi0_src_gate_en SPI0 source clock disable. When HIGH, disable clock
8:7	RO	0x0	reserved
6	RW	0x0	clk_mac_lbtest_en When HIGH, disable clock
5	RW	0x0	clk_timer3_en When HIGH, disable clock
4	RW	0x0	clk_timer2_en 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 register 3

Bit	Attr	Reset Value	Description
------------	-------------	--------------------	--------------------

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	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	hclk_sfc_gate_en SFC 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
11	RW	0x0	aclk_vdpu_src_gate_en VDPU AXI source clock disable. When HIGH, disable clock
10:3	RO	0x0	reserved
2	RW	0x0	dclk_lcdc_src_gate_en LCDC DCLK souce clock disable. When HIGH, disable clock
1:0	RO	0x0	reserved

CRU_CLKGATE4_CON

Address: Operational Base + offset (0x00e0)

Internal clock gating control register 4

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	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	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
9:4	RO	0x0	reserved
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

Bit	Attr	Reset Value	Description
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 register 5

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	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
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

Bit	Attr	Reset Value	Description
2	RW	0x0	pclk_efuse_gate_en EFUSE APB clock disable. When HIGH, disable clock
1	RW	0x0	ackl_dmac2_gate_en DMAC2 AXI clock disable. When HIGH, disable clock
0	RO	0x0	reserved

CRU_CLKGATE6_CON

Address: Operational Base + offset (0x00e8)

Internal clock gating control register 6

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	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	ackl_vio_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
11:0	RO	0x0	reserved

CRU_CLKGATE7_CON

Address: Operational Base + offset (0x00ec)

Internal clock gating control register 7

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	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
14:13	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

Bit	Attr	Reset Value	Description
7	RW	0x0	pclk_timer0_gate_en TIMER0 APB clock disable. When HIGH, disable clock
6:4	RO	0x0	reserved
3	RW	0x0	hclk_otg1_gate_en USB OTG PHY1 AHB clock disable. When HIGH, disable clock
2	RW	0x0	hclk_i2s_gate_en I2S AHB clock disable. When HIGH, disable clock
1	RO	0x0	reserved
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 register 8

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	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	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
8:7	RO	0x0	reserved
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
4	RW	0x0	pclk_i2c0_gate_en I2C0 APB clock disable. When HIGH, disable clock
3	RO	0x0	reserved

Bit	Attr	Reset Value	Description
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 register 9

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	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	ackl_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
12:7	RO	0x0	reserved
6	RW	0x0	ackl_lcdc_clock_en LCDC1 AXI clock disable. When HIGH, disable clock
5	RW	0x0	hclk_lcdc_clock_en LCDC1 AHB clock disable. When HIGH, disable clock
4:0	RO	0x0	reserved

CRU_CLKGATE10_CON

Address: Operational Base + offset (0x00f8)

Internal clock gating control register 10

Bit	Attr	Reset Value	Description
31:9	RO	0x0	reserved
8	RW	0x0	cpu_dpll_clk_en To generate ackl_cpu_src, we use 3 clock source: apll, dpll, gpll. This gating is gating dpll source.
7	RO	0x0	reserved

Bit	Attr	Reset Value	Description
6	RW	0x0	clk_hevc_core_en When HIGH, disable clock
5	RW	0x0	clk_sfc_src_en When HIGH, disable clock
4	RW	0x0	clk_nandc_src_en When HIGH, disable clock
3	RO	0x0	reserved
2	RW	0x0	clk_pvtm_video_en When HIGH, disable clock
1	RW	0x0	clk_pvtm_gpu_en When HIGH, disable clock
0	RW	0x0	clk_pvtm_core_en 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 0xecfa8, it will generate second global software reset.

CRU_SOFTRST0_CON

Address: Operational Base + offset (0x0110)

Internal software reset control register 0

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	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	pd_core_niu_arstn_req Core noc aclk software reset request. When HIGH, reset relative logic
14	RW	0x0	strc_sys_asrstn_req Structre system AXI software reset request. When HIGH, reset relative logic

Bit	Attr	Reset Value	Description
13	RW	0x0	topdbg_srstn_req When HIGH, reset relative logic
12	RW	0x0	I2c_srstn_req L2C software reset request. When HIGH, reset relative logic
11:10	RO	0x0	reserved
9	R/W SC	0x0	core1_por_srstn_req core1 por reset software reset request. When HIGH, reset relative logic
8	R/W SC	0x0	core0_por_srstn_req core0 por reset software reset request. When HIGH, reset relative logic
7:6	RO	0x0	reserved
5	RW	0x0	core1_dbg_srstn_req core1 CPU debug software reset request. When HIGH, reset relative logic
4	RW	0x0	core0_dbg_srstn_req core0 CPU debug software reset request. When HIGH, reset relative logic
3:2	RO	0x0	reserved
1	R/W SC	0x0	core1_srstn_req core1 CPU software reset request. When HIGH, reset relative logic
0	R/W SC	0x0	core0_srstn_req core0 CPU software reset request. When HIGH, reset relative logic

CRU_SOFTRST1_CON

Address: Operational Base + offset (0x0114)

Internal software reset control register 1

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	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_dll_srstn_req CORE DLL software reset request. When HIGH, reset relative logic

Bit	Attr	Reset Value	Description
12	RW	0x0	timer1_srstn_req Timer1 software reset request. When HIGH, reset relative logic
11	RW	0x0	timer0_srstn_req Timer0 software reset request. When HIGH, reset relative logic
10	RW	0x0	gpu_dll_srstn_req GPU DLL software reset request. When HIGH, reset relative logic
9	RW	0x0	video_pll_srstn_req Video pll software reset request. When HIGH, reset relative logic
8	RW	0x0	i2s_srstn_req I2S 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
4	RW	0x0	timer3_srstn_req Timer3 clock software reset request. When HIGH, reset relative logic
3	RW	0x0	ahb2apb_hsrstn_req AHB2APB software reset request. When HIGH, reset relative logic
2	RO	0x0	reserved
1	RW	0x0	cpusys_hsrstn_req CPU AHB software reset request. When HIGH, reset relative logic
0	RW	0x0	timer2_srstn_req Timer2 clock software reset request. When HIGH, reset relative logic

CRU_SOFTRST2_CON

Address: Operational Base + offset (0x0118)

Internal software reset control register 2

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	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	RW	0x0	sfc_srstn_req SFC clock software reset request. When HIGH, reset relative logic
14	RO	0x0	reserved
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
6:3	RO	0x0	reserved
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_SOFRST3_CON

Address: Operational Base + offset (0x011c)

Internal software reset control register 3

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	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

Bit	Attr	Reset Value	Description
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	RO	0x0	reserved
11	RW	0x0	periphsys_psrstn_req PERIPH APP 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	RO	0x0	reserved
7	RW	0x0	grf_srstn_req GRF software reset request. When HIGH, reset relative logic
6:5	RO	0x0	reserved
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
2:1	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 register 4

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	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

Bit	Attr	Reset Value	Description
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	usb0tg1_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	usb0tg0_srstn_req USBOTG0 software reset request. When HIGH, reset relative logic OTG Controller hclk domain reset.
4	RW	0x0	nandc_srstn_req NANDC software reset request. When HIGH, reset relative logic
3	RO	0x0	reserved
2	RW	0x0	mac_srstn_req Mac clock software reset request. When HIGH, reset relative logic
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 register 5

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	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	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

Bit	Attr	Reset Value	Description
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	RO	0x0	reserved
6	RW	0x0	wdt_srstn_req WDT software reset request. When HIGH, reset relative logic
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 register 6

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	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	usbpor_srst_req USBPHY POR software reset request. When HIGH, reset relative logic. USB phy analog domain reset, including both OTG and HOST phy .
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.

Bit	Attr	Reset Value	Description
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.
6:4	RO	0x0	reserved
3	RW	0x0	vio_bus_hsrstn_req VIO bus AHB software reset request. When HIGH, reset relative logic
2:1	RO	0x0	reserved
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 register 7

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	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	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	lc当地d_srstn_req LCDC1 DCLK software reset request. When HIGH, reset relative logic
6	RW	0x0	lc当地c_hsrstn_req LCDC1 AHB software reset request. When HIGH, reset relative logic
5	RW	0x0	lc当地c_asrstn_req LCDC1 AXI 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_srstn_req Hevc clock software reset request. When HIGH, reset relative logi

Bit	Attr	Reset Value	Description
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_SOFRST8_CON

Address: Operational Base + offset (0x0130)

Internal software reset control register 8

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:4	RO	0x0	reserved
3	RW	0x0	dbg_psrstn_req DEBUG APB software reset request. When HIGH, reset relative logic
2:0	RO	0x0	reserved

CRU_MISC_CON

Address: Operational Base + offset (0x0134)

SCU control register

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	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	usb480m_24m_sel 0: usbphy480m; 1: xin24m
14:11	RO	0x0	reserved
10:8	RW	0x0	testclk_sel Output clock selection for test 3'b001: clk_i2s 3'b010: clk_core 3'b011: clk_ddrphy 3'b100: sclk_lcdc 3'b101: clk_gpu 3'b110: aclk_peri 3'b111: aclk_cpu
7:1	RO	0x0	reserved

Bit	Attr	Reset Value	Description
0	RW	0x0	core0_porst_wdt_sel Select reset watchdog when A7 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	cru_wdt_con 1'b1: watch dog first soft reset 1'b0: watch dog second soft reset
14:10	RO	0x0	reserved
9:0	RW	0x064	glb_RST_CNT_TH Global soft reset counter threshold

CRU_SDMMC_CON0

Address: Operational Base + offset (0x0144)

SDMMC clock generation control register 0

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	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	RW	0x0	drv_sel drv select
10:3	RW	0x00	drv_delaynum drv delaynum
2:1	RW	0x2	drv_degree drive degree
0	RW	0x0	init_state initial state

CRU_SDMMC_CON1

Address: Operational Base + offset (0x0148)

SDMMC clock generation control register 1

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	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	RW	0x0	sample_sel sample select

Bit	Attr	Reset Value	Description
9:2	RW	0x00	sample_delaynum sample delaynum
1:0	RW	0x0	sample_degree sample degree

CRU_SDIO_CON0

Address: Operational Base + offset (0x014c)

SDIO clock generation control register 0

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	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	RW	0x0	drv_sel drv select
10:3	RW	0x00	drv_delaynum drv delaynum
2:1	RW	0x2	drv_degree drv degree
0	RW	0x0	init_state initial state

CRU_SDIO_CON1

Address: Operational Base + offset (0x0150)

SDIO clock generation control register 1

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	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	RW	0x0	sample_sel sample select
9:2	RW	0x00	sample_delaynum sample delaynum
1:0	RW	0x0	sample_degree sample degree

CRU_EMMC_CON0

Address: Operational Base + offset (0x0154)

EMMC clock generation control register 0

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	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

Bit	Attr	Reset Value	Description
11	RW	0x0	drv_sel drv select
10:3	RW	0x00	drv_delaynum drv delaynum
2:1	RW	0x2	drv_degree drv degree
0	RW	0x0	init_state initial state

CRU_EMMC_CON1

Address: Operational Base + offset (0x0158)

EMMC clock generation control register 1

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	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	RW	0x0	sample_sel sample select
9:2	RW	0x00	sample_delaynum sample delaynum
1:0	RW	0x0	sample_degree sample degree

CRU_RST_ST

Address: Operational Base + offset (0x0160)

WDT and global reset status

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3	RO	0x0	wdt_glb_na_srstn_st second watch dog reset request
2	RO	0x0	wdt_glb_srstn_st watch dog first reset request
1	RO	0x0	glb_na_srstn_st second globle reset request
0	RO	0x0	glb_srstn_st first global reset request

CRU_PLL_MASK_CON

Address: Operational Base + offset (0x01f0)

PLL mask control register

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved

Bit	Attr	Reset Value	Description
15:0	RW	0x5a5a	pll_mask_con When pll_mask_con = 0x5a5a, PLL_CON register would be Read_Write , otherwise it would be Read Only.

3.7 Timing Diagram

Power on reset timing is shown as follow:

Fig. 3-4 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 de-assert, and the PLL max lock time is 1500 PLL REFCLK cycles. And then the system will wait about 138us, and then de-assert 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 de-assert signal rstn_pre, which is used to generate power on reset of all IPs.

3.8 Application Notes

3.8.1 PLL usage

The chip uses 2.4GHz for all three PLLs (ARM PLL, DDR PLL and GENERAL PLL).

A. PLL output frequency configuration

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

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

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

If DSMPD = 1 (DSM is disabled, "integer mode")

$$\text{FOUTVCO} = \text{FREF} / \text{REFDIV} * \text{FBDIV}$$

$$\text{FOUTPOSTDIV} = \text{FOUTVCO} / \text{POSTDIV1} / \text{POSTDIV2}$$

When FREF is 24MHz, and if 700MHz FOUTPOSTDIV is needed. The configuration can be:

$$\begin{aligned} \text{DSMPD} &= 1 \\ \text{REFDIV} &= 6 \\ \text{FBDIV} &= 175 \\ \text{POSTDIV1} &= 1 \\ \text{POSTDIV2} &= 1 \end{aligned}$$

And then

$$\text{FOUTVCO} = \text{FREF} / \text{REFDIV} * \text{FBDIV} = 24/6*175=700$$

$$\text{FOUTPOSTDIV} = \text{FOUTVCO} / \text{POSTDIV1} / \text{POSTDIV2}=700/1/1=700$$

If DSMPD = 0 (DSM is enabled, "fractional mode")

$$\text{FOUTVCO} = \text{FREF} / \text{REFDIV} * (\text{FBDIV} + \text{FRAC} / 224)$$

$$\text{FOUTPOSTDIV} = \text{FOUTVCO} / \text{POSTDIV1} / \text{POSTDIV2}$$

When FREF is 24MHz, and if 491.52MHz FOUTPOSTDIV is needed. The configuration can be:

$$\begin{aligned} \text{DSMPD} &= 0 \\ \text{REFDIV} &= 1 \\ \text{FBDIV} &= 40 \\ \text{FRAC} &= 24'hf5c28f \\ \text{POSTDIV1} &= 2 \\ \text{POSTDIV2} &= 1 \end{aligned}$$

And then

$$\text{FOUTVCO} = \text{FREF} / \text{REFDIV} * (\text{FBDIV} + \text{FRAC} / 224) = 24/1*(40+24'hf5c28f /224)= 983.04$$

$$\text{FOUTPOSTDIV} = \text{FOUTVCO} / \text{POSTDIV1} / \text{POSTDIV2}=983.04/2/1=491.52$$

B. PLL frequency range requirement

All the value range requirements are as follow.

FREF(Input Frequency Range in Integer Mode):	1MHz to 800MHz
FREF(Input Frequency Range in Fractional Mode):	10MHz to 800MHz
FREF/REFDIV(The divided reference frequency):	1 to 50MHz
FOUTVCO:	600MHz 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.

3.8.2 PLL frequency change and lock check

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

PLL lock state can be checked in CRU_APLL_CON1[10], CRU_DPLL_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 2~5s. 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 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.

3.8.3 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.

3.8.4 Global software reset

Two global software resets are designed in the chip, you can program CRU_GLB_SRST_FST_VALUE[15:0] as 0fdb9 to assert the first global software reset glb_srstn_1 and program CRU_GLB_SRST_SND_VALUE[15:0] as 0eca8 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 Embedded Processor (Cortex-A7)

4.1 Overview

The Cortex-A7 MP subsystem of the device is based on the symmetric multiprocessor (SMP) architecture, thus the Dual Cortex-A7 MPU subsystem delivers higher performance and optimal power management, debug and emulation capabilities.

The Cortex-A7 MP subsystem incorporates two Cortex-A7 central processing units (CPUs), level 2(L2) cache shared between the two CPUs, and uses PL310 as L2 cache controller. Each CPU has 32KB of level 1 (L1) instruction cache, 32KB of L1 data cache, separate dedicated power domain, and includes one Neon and Vector Floating Point Unit coprocessors. The Cortex-A7 MP subsystem also includes standard CoreSight components to support SMP debug and emulation, snoop control unit (SCU), interrupt controller (GIC), and clock and reset manager.

The key features of the Cortex-A7 MP subsystem include:

- ARM Coretex-A7 based dual MPU subsystem with SMP architecture
 - Full implements the ARMv7-A architecture profile that includes SIMD and VFP
 - 32KB L1 I-cache and 32KB L1 D-cache per CPU
 - In-order pipeline with direct and indirect branch prediction
 - Harvard Level 1 (L1) memory system with a Memory Management Unit (MMU)
 - SCU ensures memory coherency between the two CPUs
 - Interrupt controller with 128 hardware interrupt inputs
- 128KB L2 cache shared between the two CPUs
 - Fixed line length of 64 bytes
 - Physically indexed and tagged cache
 - 8-way set-associative cache structure
 - Pseudo-random cache replacement policy

4.2 Block Diagram

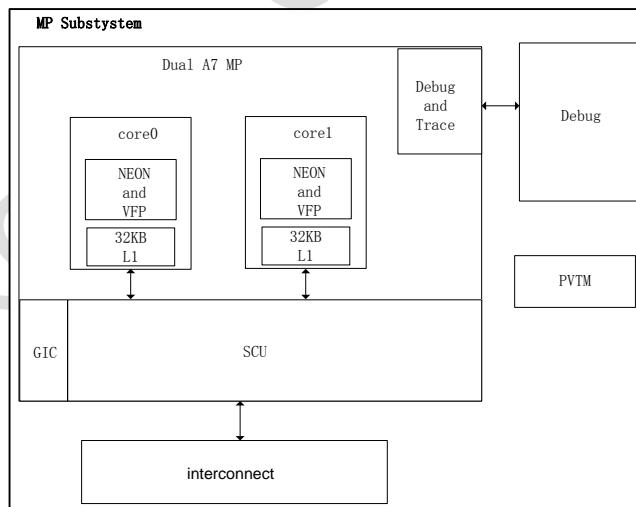


Fig. 4-1 MP Subsystem architecture

4.3 Function Description

Please refer to the document Cortex-A7_MPCore_Technical_Reference_Manual.pdf for the CPU detail description.

4.4 Register Description

Please refer to the document Cortex-A7_MPCore_Technical_Reference_Manual.pdf for the CPU detail description.

Chapter 5 General Register Files (GRF)

5.1 Overview

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

5.2 Function Description

The function of general register file is:

- IOMUX control
- GPIO PAD pulldown and pullup control
- Common system control
- Record the system state

5.3 Register Description

5.3.1 Registers Summary

Name	Offset	Size	Reset Value	Description
GRF_GPIO0A_IOMUX	0x00a8	W	0x00000000	GPIO0A iomux control
GRF_GPIO0B_IOMUX	0x00ac	W	0x00000000	GPIO0B iomux control
GRF_GPIO0C_IOMUX	0x00b0	W	0x00000000	GPIO0C iomux control
GRF_GPIO0D_IOMUX	0x00b4	W	0x00000000	GPIO0D iomux control
GRF_GPIO1A_IOMUX	0x00b8	W	0x00000000	GPIO1A iomux control
GRF_GPIO1B_IOMUX	0x00bc	W	0x00000000	GPIO1B iomux control
GRF_GPIO1C_IOMUX	0x00c0	W	0x00000000	GPIO1C iomux control
GRF_GPIO1D_IOMUX	0x00c4	W	0x00000000	GPIO1D iomux control
GRF_GPIO2A_IOMUX	0x00c8	W	0x00000000	GPIO2A iomux control
GRF_GPIO2B_IOMUX	0x00cc	W	0x00000000	GPIO2B iomux control
GRF_GPIO2C_IOMUX	0x00d0	W	0x00000000	GPIO2C iomux control
GRF_GPIO2D_IOMUX	0x00d4	W	0x00000000	GPIO2D iomux control
GRF_GPIO_DS	0x0100	W	0x00000000	GPIO DS control
GRF_GPIO0L_PULL	0x0118	W	0x00000000	GPIO0A/GPIO0B pull up/down control
GRF_GPIO0H_PULL	0x011c	W	0x00000000	GPIO0C/GPIO0D pull up/down control
GRF_GPIO1L_PULL	0x0120	W	0x00000000	GPIO1A/GPIO1B pull up/down control
GRF_GPIO1H_PULL	0x0124	W	0x00000000	GPIO1C/GPIO1D pull up/down control
GRF_GPIO2L_PULL	0x0128	W	0x00000000	GPIO2A/GPIO2B pull up/down control
GRF_GPIO2H_PULL	0x012c	W	0x00000000	GPIO2C/GPIO2D pull up/down control
GRF_SOC_CON0	0x0140	W	0x00000820	SoC control register 0
GRF_SOC_CON1	0x0144	W	0x00000000	SoC control register 1
GRF_SOC_CON2	0x0148	W	0x00000082	SoC control register 2
GRF_SOC_STATUS0	0x014c	W	0x00000800	SoC status register 0

Name	Offset	Size	Reset Value	Description
GRF_SOC_CON3	0x0154	W	0x00001f86	SoC control register 3
GRF_UOC0_CON5	0x017c	W	0x00000000	OTG control register
GRF_UOC1_CON4	0x0190	W	0x00000001c	USB HOST 2.0 control register
GRF_UOC1_CON5	0x0194	W	0x000000000	USB HOST 2.0 control register
GRF_DDRC_STAT	0x019c	W	0x00180000	DDRC status register
GRF_UOC_CON6	0x01a0	W	0x000042a0	USB control register
GRF_SOC_STATUS1	0x01a4	W	0x000000000	SoC status register 1
GRF_CPU_CON0	0x01a8	W	0x00002002	CPU control register 0
GRF_CPU_CON1	0x01ac	W	0x000000000	CPU control register 1
GRF_CPU_CON2	0x01b0	W	0x00000003f	CPU control register 2
GRF_CPU_STATUS0	0x01c0	W	0x000000006	CPU status register 0
GRF_CPU_STATUS1	0x01c4	W	0x000000000	CPU status register 2
GRF_OS_REG0	0x01c8	W	0x000000000	Software OS register 0
GRF_OS_REG1	0x01cc	W	0x000000000	Software OS register 1
GRF_OS_REG2	0x01d0	W	0x000000000	Software OS register 2
GRF_OS_REG3	0x01d4	W	0x000000000	Software OS register 3
GRF_OS_REG4	0x01d8	W	0x000000000	Software OS register 4
GRF_OS_REG5	0x01dc	W	0x000000000	Software OS register 5
GRF_OS_REG6	0x01e0	W	0x000000000	Software OS register 6
GRF_OS_REG7	0x01e4	W	0x000000000	Software OS register 7
GRF_PVTM_CON0	0x0200	W	0x000000000	PVTM control register 0
GRF_PVTM_CON1	0x0204	W	0x000000000	PVTM control register 1
GRF_PVTM_CON2	0x0208	W	0x000000000	PVTM control register 2
GRF_PVTM_CON3	0x020c	W	0x000000000	PVTM control register 3
GRF_PVTM_STATUS0	0x0210	W	0x000000000	PVTM status register 0
GRF_PVTM_STATUS1	0x0214	W	0x000000000	PVTM status register 1
GRF_PVTM_STATUS2	0x0218	W	0x000000000	PVTM status register 2
GRF_PVTM_STATUS3	0x021c	W	0x000000000	PVTM status register 3
GRF_DFI_WRNUM	0x0220	W	0x000000000	DFI write number register
GRF_DFI_RDNUM	0x0224	W	0x000000000	DFI read number register
GRF_DFI_ACTNUM	0x0228	W	0x000000000	DFI active number register
GRF_DFI_TIMERVAL	0x022c	W	0x000000000	DFI work time
GRF_NIF_FIFO0	0x0230	W	0x000000000	NIF status register 0
GRF_NIF_FIFO1	0x0234	W	0x000000000	NIF status register 1
GRF_NIF_FIFO2	0x0238	W	0x000000000	NIF status register 2
GRF_NIF_FIFO3	0x023c	W	0x000000000	NIF status register 3
GRF_USBPHY0_CON0	0x0280	W	0x00008618	usbphy control register
GRF_USBPHY0_CON1	0x0284	W	0x0000e007	usbphy control register
GRF_USBPHY0_CON2	0x0288	W	0x000082aa	usbphy control register
GRF_USBPHY0_CON3	0x028c	W	0x00000200	usbphy control register
GRF_USBPHY0_CON4	0x0290	W	0x00000002	usbphy control register
GRF_USBPHY0_CON5	0x0294	W	0x000000000	usbphy control register
GRF_USBPHY0_CON6	0x0298	W	0x000000004	usbphy control register
GRF_USBPHY0_CON7	0x029c	W	0x000068c0	usbphy control register

Name	Offset	Size	Reset Value	Description
GRF_USBPHY1_CON0	0x02a0	W	0x00008618	usbphy control register
GRF_USBPHY1_CON1	0x02a4	W	0x0000e007	usbphy control register
GRF_USBPHY1_CON2	0x02a8	W	0x000082aa	usbphy control register
GRF_USBPHY1_CON3	0x02ac	W	0x00000200	usbphy control register
GRF_USBPHY1_CON4	0x02b0	W	0x00000002	usbphy control register
GRF_USBPHY1_CON5	0x02b4	W	0x00000000	usbphy control register
GRF_USBPHY1_CON6	0x02b8	W	0x00000004	usbphy control register
GRF_USBPHY1_CON7	0x02bc	W	0x000068c0	usbphy control register
GRF_CHIP_TAG	0x0300	W	0x00003012	chip tag register
GRF_SDMMC_DET_CNT	0x0304	W	0x00000cff	SDMMC detectn filter count
GRF_EFUSE_PRG	0x037c	W	0x00000000	eFuse program enable register

Notes: **S**ize: **B**- Byte (8 bits) access, **H**W- Half WORD (16 bits) access, **W**-WORD (32 bits) access

5.3.2 Detail Register Description

GRF_GPIO0A_IOMUX

Address: Operational Base + offset (0x00a8)

GPIO0A 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.
15:7	RO	0x0	reserved
6	RW	0x0	gpio0a3_sel GPIO0A[3] iomux select 2'b01: i2c1_sda 2'b00: gpio
5	RO	0x0	reserved
4	RW	0x0	gpio0a2_sel GPIO0A[2] iomux select 2'b01: i2c1_scl 2'b00: gpio
3:2	RW	0x0	gpio0a1_sel GPIO0A[1] iomux select 2'b10: pwm2 2'b01: i2c0_sda 2'b00: gpio

Bit	Attr	Reset Value	Description
1:0	RW	0x0	gpio0a0_sel GPIO0A[0] iomux select 2'b10: pwm1 2'b01: i2c0_scl 2'b00: gpio

GRF_GPIOOB_IOMUX

Address: Operational Base + offset (0x00ac)

GPIO0B 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	RO	0x0	reserved
13:12	RW	0x0	gpio0b6_sel GPIO0B[6] iomux select 2'b10: i2s1_sclk 2'b01: mmc1_d3 2'b00: gpio
11:10	RW	0x0	gpio0b5_sel GPIO0B[5] iomux select 2'b10: i2s1_sdi 2'b01: mmc1_d2 2'b00: gpio
9:8	RW	0x0	gpio0b4_sel GPIO0B[4] iomux select 2'b10: i2s1_lrcktx 2'b01: mmc1_d1 2'b00: gpio
7:6	RW	0x0	gpio0b3_sel GPIO0B[3] iomux select 2'b10: i2s1_lrckrx 2'b01: mmc1_d0 2'b00: gpio
5:4	RO	0x0	reserved

Bit	Attr	Reset Value	Description
3:2	RW	0x0	gpio0b1_sel GPIO0B[1] iomux select 2'b10: i2s1_mclk 2'b01: mmc1_clkout 2'b00: gpio
1:0	RW	0x0	gpio0b0_sel GPIO0B[0] iomux select 2'b10: i2s1_sdo 2'b01: mmc1_cmd 2'b00: gpio

GRF_GPIOOC_IOMUX

Address: Operational Base + offset (0x00b0)

GPIO0C 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.
15:9	RO	0x0	reserved
8	RW	0x0	gpio0c4_sel GPIO0C[4] iomux select 1'b1: drive_vbus 1'b0: gpio
7	RO	0x0	reserved
6	RW	0x0	gpio0c3_sel GPIO0C[3] iomux select 1'b1: uart0_ctsn 1'b0: gpio
5	RO	0x0	reserved
4	RW	0x0	gpio0c2_sel GPIO0C[2] iomux select 1'b1: uart0_rtsn 1'b0: gpio
3	RO	0x0	reserved
2	RW	0x0	gpio0c1_sel GPIO0C[1] iomux select 1'b1: uart0_sin 1'b0: gpio
1	RO	0x0	reserved

Bit	Attr	Reset Value	Description
0	RW	0x0	gpio0c0_sel GPIO0C[0] iomux select 1'b1: uart0_sout 1'b0: gpio

GRF_GPIO0D_IOMUX

Address: Operational Base + offset (0x00b4)

GPIO0D 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.
15:9	RO	0x0	reserved
8	RW	0x0	gpio0d4_sel GPIO0D[4] iomux select 1'b1: spdif 1'b0: gpio
7	RO	0x0	reserved
6	RW	0x0	gpio0d3_sel GPIO0D[3] iomux select 1'b1: pwm3(IR) 1'b0: gpio
5	RO	0x0	reserved
4	RW	0x0	gpio0d2_sel GPIO0D[2] iomux select 1'b1: pwm0 1'b0: gpio
3:0	RO	0x0	reserved

GRF_GPIO1A_IOMUX

Address: Operational Base + offset (0x00b8)

GPIO1A iomux control

Bit	Attr	Reset Value	Description
-----	------	-------------	-------------

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.
15:11	RO	0x0	reserved
10	RW	0x0	gpio1a5_sel GPIO1A[5] iomux select 1'b1: i2s_sdi 1'b0: gpio
9	RO	0x0	reserved
8	RW	0x0	gpio1a4_sel GPIO1A[4] iomux select 1'b1: i2s_sdo 1'b0: gpio
7	RO	0x0	reserved
6	RW	0x0	gpio1a3_sel GPIO1A[3] iomux select 1'b1: i2s_lrcktx 1'b0: gpio
5:4	RW	0x0	gpio1a2_sel GPIO1A[2] iomux select 2'b10: pwm1_0 2'b01: i2s_lrckrx 2'b00: gpio
3	RO	0x0	reserved
2	RW	0x0	gpio1a1_sel GPIO1A[1] iomux select 1'b1: i2s_sclk 1'b0: gpio
1	RO	0x0	reserved
0	RW	0x0	gpio1a0_sel GPIO1A[0] iomux select 1'b1: i2s_mclk 1'b0: gpio

GRF_GPIO1B_IOMUX

Address: Operational Base + offset (0x00bc)

GPIO1B iomux control

Bit	Attr	Reset Value	Description
-----	------	-------------	-------------

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	RO	0x0	reserved
14	RW	0x0	gpio1b7_sel GPIO1B[7] iomux select 1'b1: mmc0_cmd 1'b0: gpio
13:7	RO	0x0	reserved
6	RW	0x0	gpio1b3_sel GPIO1B[3] iomux select 1'b1: hdmi_hpd 1'b0: gpio
5	RO	0x0	reserved
4	RW	0x0	gpio1b2_sel GPIO1B[2] iomux select 1'b1: hdmi_scl 1'b0: gpio
3	RO	0x0	reserved
2	RW	0x0	gpio1b1_sel GPIO1B[1] iomux select 1'b1: hdmi_sda 1'b0: gpio
1	RO	0x0	reserved
0	RW	0x0	gpio1b0_sel GPIO1B[0] iomux select 1'b1: hdmi_cec 1'b0: gpio

GRF_GPIO1C_IOMUX

Address: Operational Base + offset (0x00c0)

GPIO1C iomux control

Bit	Attr	Reset Value	Description
-----	------	-------------	-------------

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.
15:12	RO	0x0	reserved
11:10	RW	0x0	gpio1c5_sel GPIO1C[5] iomux select 2'b10: jtag_tms when sdmmc0_detectn is invalid 2'b01: mmc0_d3 2'b00: gpio
9:8	RW	0x0	gpio1c4_sel GPIO1C[4] iomux select 2'b10: jtag_tck when sdmmc0_detectn is invalid 2'b01: mmc0_d2 2'b00: gpio
7:6	RW	0x0	gpio1c3_sel GPIO1C[3] iomux select 2'b10: uart2_sout 2'b01: mmc0_d1 2'b00: gpio
5:4	RW	0x0	gpio1c2_sel GPIO1C[2] iomux select 2'b10:uart2_sin 2'b01:mmc0_d0 2'b00:gpio
3	RO	0x0	reserved
2	RW	0x0	gpio1c1_sel GPIO1C[1] iomux select 1'b1: mmc0_detn 1'b0: gpio
1	RO	0x0	reserved
0	RW	0x0	gpio1c0_sel GPIO1C[0] iomux select 1'b1: mmc0_clkout 1'b0: gpio

GRF_GPIO1D_IOMUX

Address: Operational Base + offset (0x00c4)

GPIO1D iomux control

Bit	Attr	Reset Value	Description
------------	-------------	--------------------	--------------------

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	<p>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.</p>
15:14	RW	0x0	<p>gpio1d7_sel GPIO1D[7] iomux select 2'b11: spi_csn1 2'b10: emmc_d7 2'b01: nand_d7 2'b00: gpio</p>
13:12	RW	0x0	<p>gpio1d6_sel GPIO1D[6] iomux select 2'b11: spi_csn0 2'b10: emmc_d6 2'b01: nand_d6 2'b00: gpio</p>
11:10	RW	0x0	<p>gpio1d5_sel GPIO1D[5] iomux select 2'b11: spi_txd 2'b10: emmc_d5 2'b01: nand_d5 2'b00: gpio</p>
9:8	RW	0x0	<p>gpio1d4_sel GPIO1D[4] iomux select 2'b11: spi_rxd 2'b10: emmc_d4 2'b01: nand_d4 2'b00: gpio</p>
7:6	RW	0x0	<p>gpio1d3_sel GPIO1D[3] iomux select 2'b11: sfc_sio3 2'b10: emmc_d3 2'b01: nand_d3 2'b00: gpio</p>
5:4	RW	0x0	<p>gpio1d2_sel GPIO1D[2] iomux select 2'b11: sfc_sio2 2'b10: emmc_d2 2'b01: nand_d2 2'b00: gpio</p>

Bit	Attr	Reset Value	Description
3:2	RW	0x0	gpio1d1_sel GPIO1D[1] iomux select 2'b11: sfc_sio1 2'b10: emmc_d1 2'b01: nand_d1 2'b00: gpio
1:0	RW	0x0	gpio1d0_sel GPIO1D[0] iomux select 2'b11: sfc_sio0 2'b10: emmc_d0 2'b01: nand_d0 2'b00: gpio

GRF_GPIO2A_IOMUX

Address: Operational Base + offset (0x00c8)

GPIO2A 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	RO	0x0	reserved
14	RW	0x0	gpio2a7_sel GPIO2A[7] iomux select 1'b1: testclk_out 1'b0: gpio
13	RO	0x0	reserved
12	RW	0x0	gpio2a6_sel GPIO2A[6] iomux select 1'b1: nand_cs0 1'b0: gpio
11:10	RO	0x0	reserved
9:8	RW	0x0	gpio2a4_sel GPIO2A[4] iomux select 2'b11: sfc_clk 2'b10: emmc_cmd 2'b01: nand_rdy 2'b00: gpio

Bit	Attr	Reset Value	Description
7:6	RW	0x0	gpio2a3_sel GPIO2A[3] iomux select 2'b10: sfc_csn1 2'b01: nand_rdn 2'b00: gpio
5:4	RW	0x0	gpio2a2_sel GPIO2A[2] iomux select 2'b10: sfc_csn0 2'b01: nand_wrn 2'b00: gpio
3:2	RW	0x0	gpio2a1_sel GPIO2A[1] iomux select 2'b10: emmc_clkout 2'b01: nand_cle 2'b00: gpio
1:0	RW	0x0	gpio2a0_sel GPIO2A[0] iomux select 2'b10: spi_clk 2'b01: nand_ale 2'b00: gpio

GRF_GPIO2B_IOMUX

Address: Operational Base + offset (0x00cc)

GPIO2B 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	RO	0x0	reserved
14	RW	0x0	gpio2b7_sel GPIO2B[7] iomux select 1'b1: mac_rxer 1'b0: gpio
13:12	RW	0x0	gpio2b6_sel GPIO2B[6] iomux select 2'b10: mac_clkin 2'b01: mac_clkout 2'b00: gpio
11	RO	0x0	reserved

Bit	Attr	Reset Value	Description
10	RW	0x0	gpio2b5_sel GPIO2B[5] iomux select 1'b1: mac_txen 1'b0: gpio
9	RO	0x0	reserved
8	RW	0x0	gpio2b4_sel GPIO2B[4] iomux select 1'b1: mac_mdio 1'b0: gpio
7:5	RO	0x0	reserved
4	RW	0x0	gpio2b2_sel GPIO2B[2] iomux select 1'b1: mac_crs 1'b0: gpio
3:0	RO	0x0	reserved

GRF_GPIO2C_IOMUX

Address: Operational Base + offset (0x00d0)

GPIO2C 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	gpio2c7_sel GPIO2C[7] iomux select 2'b10: testclk_out1 2'b01: uart1_sout 2'b00: gpio
13	RO	0x0	reserved
12	RW	0x0	gpio2c6_sel GPIO2C[6] iomux select 1'b1: uart1_sin 1'b0: gpio
11	RO	0x0	reserved
10	RW	0x0	gpio2c5_sel GPIO2C[5] iomux select 1'b1: i2c2_scl 1'b0: gpio
9	RO	0x0	reserved

Bit	Attr	Reset Value	Description
8	RW	0x0	gpio2c4_sel GPIO2C[4] iomux select 1'b1: i2c2_sda 1'b0: gpio
7	RO	0x0	reserved
6	RW	0x0	gpio2c3_sel GPIO2C[3] iomux select 1'b1: mac_txd0 1'b0: gpio
5	RO	0x0	reserved
4	RW	0x0	gpio2c2_sel GPIO2C[2] iomux select 1'b1: mac_txd1 1'b0: gpio
3	RO	0x0	reserved
2	RW	0x0	gpio2c1_sel GPIO2C[1] iomux select 1'b1: mac_rxd0 1'b0: gpio
1	RO	0x0	reserved
0	RW	0x0	gpio2c0_sel GPIO2C[0] iomux select 1'b1: mac_rxd1 1'b0: gpio

GRF_GPIO2D_IOMUX

Address: Operational Base + offset (0x00d4)

GPIO2D 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.
15:13	RO	0x0	reserved
12	RW	0x0	gpio2d6_sel GPIO2D[6] iomux select 1'b1: i2s_sdo1 1'b0: gpio
11	RO	0x0	reserved

Bit	Attr	Reset Value	Description
10	RW	0x0	gpio2d5_sel GPIO2D[5] iomux select 1'b1: i2s_sdo2 1'b0: gpio
9	RO	0x0	reserved
8	RW	0x0	gpio2d4_sel GPIO2D[4] iomux select 1'b1: i2s_sdo3 1'b0: gpio
7:3	RO	0x0	reserved
2	RW	0x0	gpio2d1_sel GPIO2D[1] iomux select 1'b1: mac_mdc 1'b0: gpio
1:0	RO	0x0	reserved

GRF_GPIO_DS

Address: Operational Base + offset (0x0100)

GPIO DS control

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~bit15 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	gpio_1b3_ds driver strength setting 2'b00: 2.6mA ~ 6.8mA 2'b01: 5.1mA ~ 14mA 2'b10: 7.7mA ~ 20mA 2'b11: 10mA ~ 27mA
9:8	RW	0x0	gpio_1b2_ds driver strength setting 2'b00: 2.6mA ~ 6.8mA 2'b01: 5.1mA ~ 14mA 2'b10: 7.7mA ~ 20mA 2'b11: 10mA ~ 27mA

Bit	Attr	Reset Value	Description
7:6	RW	0x0	gpio_1b1_ds driver strength setting 2'b00: 2.6mA ~ 6.8mA 2'b01: 5.1mA ~ 14mA 2'b10: 7.7mA ~ 20mA 2'b11: 10mA ~ 27mA
5:4	RW	0x0	gpio_1b0_ds driver strength setting 2'b00: 2.6mA ~ 6.8mA 2'b01: 5.1mA ~ 14mA 2'b10: 7.7mA ~ 20mA 2'b11: 10mA ~ 27mA
3	RW	0x0	gpio_1b3_sl Control the slew rata of gpio_1b3 1'b0: slow 1'b1: fast
2	RW	0x0	gpio_1b2_sl Control the slew rata of gpio_1b2 1'b0: slow 1'b1: fast
1	RW	0x0	gpio_1b1_sl Control the slew rata of gpio_1b1 1'b0: slow 1'b1: fast
0	RW	0x0	gpio_1b0_sl Control the slew rata of gpio_1b0 1'b0: slow 1'b1: fast

GRF_GPIO0L_PULL

Address: Operational Base + offset (0x0118)

GPIO0A/GPIO0B pull up/down control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~bit15 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	<p>gpio0b_pull GPIO0B pull up/down enable Values written to this register independently control Pullup/Pulldown or not for the corresponding data bit in GPIO. 1'b0: pull up/down enable, PAD type will decide to be up or down, not related with this value 1'b1: pull up/down disable bit8 - GPIO0B[0] pull up/down control bit9 - GPIO0B[1] pull up/down control bit10 - GPIO0B[2] pull up/down control ... bit15 - GPIO0B[7] pull up/down control</p>
7:0	RW	0x00	<p>gpio0a_pull GPIO0A pull up/down enable Values written to this register independently control Pullup/Pulldown or not for the corresponding data bit in GPIO. 1'b0: pull up/down enable, PAD type will decide to be up or down, not related with this value 1'b1: pull up/down disable bit0 - GPIO0A[0] pull up/down control bit1 - GPIO0A[1] pull up/down control bit2 - GPIO0A[2] pull up/down control ... bit7 - GPIO0A[7] pull up/down control</p>

GRF_GPIOH_PULL

Address: Operational Base + offset (0x011c)

GPIO0C/GPIO0D pull up/down control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	<p>write_enable bit0~bit15 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.</p>

Bit	Attr	Reset Value	Description
15:8	RW	0x00	<p>gpio0d_pull GPIO0D pull up/down enable Values written to this register independently control Pullup/Pulldown or not for the corresponding data bit in GPIO. 1'b0: pull up/down enable, PAD type will decide to be up or down, not related with this value 1'b1: pull up/down disable bit8 - GPIO0D[0] pull up/down control bit9 - GPIO0D[1] pull up/down control bit10 - GPIO0D[2] pull up/down control ... bit15 - GPIO0D[7] pull up/down control</p>
7:0	RW	0x00	<p>gpio0c_pull GPIO0C pull up/down enable Values written to this register independently control Pullup/Pulldown or not for the corresponding data bit in GPIO. 1'b0: pull up/down enable, PAD type will decide to be up or down, not related with this value 1'b1: pull up/down disable bit0 - GPIO0C[0] pull up/down control bit1 - GPIO0C[1] pull up/down control bit2 - GPIO0C[2] pull up/down control ... bit7 - GPIO0C[7] pull up/down control</p>

GRF_GPIO1L_PULL

Address: Operational Base + offset (0x0120)

GPIO1A/GPIO1B pull up/down control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	<p>write_enable bit0~bit15 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.</p>

Bit	Attr	Reset Value	Description
15:8	RW	0x00	<p>gpio1b_pull GPIO1B pull up/down enable Values written to this register independently control Pullup/Pulldown or not for the corresponding data bit in GPIO. 1'b0: pull up/down enable, PAD type will decide to be up or down, not related with this value 1'b1: pull up/down disable bit8 - GPIO1B[0] pull up/down control bit9 - GPIO1B[1] pull up/down control bit10 - GPIO1B[2] pull up/down control ... bit15 - GPIO1B[7] pull up/down control</p>
7:0	RW	0x00	<p>gpio1a_pull GPIO1A pull up/down enable Values written to this register independently control Pullup/Pulldown or not for the corresponding data bit in GPIO. 1'b0: pull up/down enable, PAD type will decide to be up or down, not related with this value 1'b1: pull up/down disable bit0 - GPIO1A[0] pull up/down control bit1 - GPIO1A[1] pull up/down control bit2 - GPIO1A[2] pull up/down control ... bit7 - GPIO1A[7] pull up/down control</p>

GRF_GPIO1H_PULL

Address: Operational Base + offset (0x0124)

GPIO1C(GPIO1D) pull up/down control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	<p>write_enable bit0~bit15 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.</p>

Bit	Attr	Reset Value	Description
15:8	RW	0x00	<p>gpio1d_pull GPIO1d pull up/down enable Values written to this register independently control Pullup/Pulldown or not for the corresponding data bit in GPIO. 1'b0: pull up/down enable, PAD type will decide to be up or down, not related with this value 1'b1: pull up/down disable bit8 - GPIO1D[0] pull up/down control bit9 - GPIO1D[1] pull up/down control bit10 - GPIO1D[2] pull up/down control ... bit15 - GPIO1D[7] pull up/down control</p>
7:0	RW	0x00	<p>gpio1c_pull GPIO1C pull up/down enable Values written to this register independently control Pullup/Pulldown or not for the corresponding data bit in GPIO. 1'b0: pull up/down enable, PAD type will decide to be up or down, not related with this value 1'b1: pull up/down disable bit0 - GPIO1C[0] pull up/down control bit1 - GPIO1C[1] pull up/down control bit2 - GPIO1C[2] pull up/down control ... bit7 - GPIO1C[7] pull up/down control</p>

GRF_GPIO2L_PULL

Address: Operational Base + offset (0x0128)

GPIO2A/GPIO2B pull up/down control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	<p>write_enable bit0~bit15 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.</p>

Bit	Attr	Reset Value	Description
15:8	RW	0x00	<p>gpio2b_pull GPIO2B pull up/down enable Values written to this register independently control Pullup/Pulldown or not for the corresponding data bit in GPIO. 1'b0: pull up/down enable, PAD type will decide to be up or down, not related with this value 1'b1: pull up/down disable bit8 - GPIO2B[0] pull up/down control bit9 - GPIO2B[1] pull up/down control bit10 - GPIO2B[2] pull up/down control ... bit15 - GPIO2B[7] pull up/down control</p>
7:0	RW	0x00	<p>gpio2a_pull GPIO2A pull up/down enable Values written to this register independently control Pullup/Pulldown or not for the corresponding data bit in GPIO. 1'b0: pull up/down enable, PAD type will decide to be up or down, not related with this value 1'b1: pull up/down disable bit0 - GPIO2A[0] pull up/down control bit1 - GPIO2A[1] pull up/down control bit2 - GPIO2A[2] pull up/down control ... bit7 - GPIO2A[7] pull up/down control</p>

GRF_GPIO2H_PULL

Address: Operational Base + offset (0x012c)

GPIO2C/GPIO2D pull up/down control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	<p>write_enable bit0~bit15 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.</p>

Bit	Attr	Reset Value	Description
15:8	RW	0x00	<p>gpio2d_pull GPIO2d pull up/down enable Values written to this register independently control Pullup/Pulldown or not for the corresponding data bit in GPIO. 1'b0: pull up/down enable, PAD type will decide to be up or down, not related with this value 1'b1: pull up/down disable bit8 - GPIO2D[0] pull up/down control bit9 - GPIO2D[1] pull up/down control bit10 - GPIO2D[2] pull up/down control ... bit15 - GPIO2D[7] pull up/down control</p>
7:0	RW	0x00	<p>gpio2c_pull GPIO2C pull up/down enable Values written to this register independently control Pullup/Pulldown or not for the corresponding data bit in GPIO. 1'b0: pull up/down enable, PAD type will decide to be up or down, not related with this value 1'b1: pull up/down disable bit0 - GPIO2C[0] pull up/down control bit1 - GPIO2C[1] pull up/down control bit2 - GPIO2C[2] pull up/down control ... bit7 - GPIO2C[7] pull up/down control</p>

GRF_SOC_CON0

Address: Operational Base + offset (0x0140)

SoC control register 0

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	<p>write_enable bit 0~bit 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.</p>
15	RW	0x0	<p>emac_newrcv_en the selection of RMII receive selection 1'b0: don't support the data package without header 1'b1: support the data package without header</p>
14	RW	0x0	<p>ddr_8bit_en When 8bit ddr is used, this bit should be 1.</p>

Bit	Attr	Reset Value	Description
13	RW	0x0	msch4_mainpartialpop 1'b0: 16bit ddr 1'b1: 8bit ddr
12	RW	0x0	soc_remap remap bit control When soc_remap = 1, the bootrom is mapped to address 0x10100000 and internal memory is mapped to address 0x0.
11	RW	0x1	grf_force_jtag Force select jtag function from sdmmc0 IO, High valid.
10	RW	0x0	acodec_sel 1'b0: i2s_sdi from gpio is selected 1'b1: i2s_sdi from acodec is selected
9	RW	0x0	emac_speed 1'b0: 10MHz 1'b1: 100MHz
8	RW	0x0	emac_mode 1'b0: rmii 1'b1: mii
7	RW	0x0	mobile_ddr_sel This bit is used to tell ddr monitor the type of ddr used. 1'b0: DDR3/DDR3L 1'b1: not used
6	RW	0x0	dfi_eff_stat_en dfi monitor start to work. 1'b1: dfi monitor works. 1'b0: dfi monitor stops.
5:4	RW	0x2	sd_detectn_debounce_sel sd_detectn debounce time select 2'b00: 5ms 2'b01: 15ms 2'b10: 35ms 2'b11: 50ms
3	RW	0x0	sd_detectn_fall_int_en SD detectn negedge interrupt enable 1'b0: interrupt disable 1'b1: interrupt enable
2	RW	0x0	sd_detectn_rise_int_en SD detectn posedge interrupt enable 1'b0: interrupt disable 1'b1: interrupt enable
1	RW	0x0	sd_detectn_fall_int_pd SD detectn negedge interrupt pending bit wirte 1 to it, it will be cleared.

Bit	Attr	Reset Value	Description
0	RW	0x0	sd_detectn_rise_int_pd SD detectn posedge interrupt pending bit wirte 1 to it, it will be cleared.

GRF_SOC_CON1

Address: Operational Base + offset (0x0144)

SoC control register 1

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit 0~bit 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	peri_usb_pause peri usb ahb bus arbiter pause control USB AHB bus arbiter pause control 1'b1: pause 1'b0: work
7	RO	0x0	reserved
6	RW	0x0	i2s_lrck_sel 1'b0: seperate tx_lrck/rx_lrck signal; 1'b1: mux tx_lrck/rx_lrck signal
5	RW	0x0	pwm_sel Select pwm_0 iomux 1'b0: pwm_0(gpio0d2) 1'b1: pwm1_0(gpio1a2)
4	RW	0x0	i2s_sel 1'b0: i2s0 1'b1: i2s1
3	RW	0x0	hevclite_vcodec_sel 1'b0: vpu 1'b1: hevc
2:0	RO	0x0	reserved

GRF_SOC_CON2

Address: Operational Base + offset (0x0148)

SoC control register 2

Bit	Attr	Reset Value	Description
-----	------	-------------	-------------

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	<p>write_enable bit0~bit15 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.</p>
15	RO	0x0	reserved
14	RW	0x0	<p>msch_pwr_idlereq NOC memory scheduler idle request, high valid.</p>
13	RW	0x0	<p>core_pwr_idlereq NOC pd_core idle request, high valid.</p>
12	RW	0x0	<p>peri_pwr_idlereq NOC pd_peri idle request, high valid.</p>
11	RW	0x0	<p>vio_pwr_idlereq NOC pd_vio idle request, high valid.</p>
10	RW	0x0	<p>vpu_pwr_idlereq NOC pd_video idle request, high valid.</p>
9	RW	0x0	<p>gpu_pwr_idlereq NOC pd_gpu idle request, high valid.</p>
8	RW	0x0	<p>sys_pwr_idlereq NOC pd_cpu domain idle request, high valid.</p>
7	RW	0x1	<p>msch4_mainaddr3 When DDR3 is used, software should configure this bit to 1.</p>
6	RW	0x0	<p>vop_hdmi_den_pol HDMI den polarity 1'b1: low active 1'b0: high active</p>
5	RW	0x0	<p>vop_hdmi_vsync_pol HDMI vsync polarity 1'b1: low active 1'b0: high active</p>
4	RW	0x0	<p>vop_hdmi_hsync_pol HDMI hsync polarity 1'b1: low active 1'b0: high active</p>
3:2	RO	0x0	reserved
1	RW	0x1	<p>upctl_c_sysreq software configure enter DDR self-refresh by lowpower interface System low-power request. Low active.</p>

Bit	Attr	Reset Value	Description
0	RW	0x0	<p>upctl_c_active_in DDR clock active in. External signal from system that flags if a hardware low power request can be accepted or should always be denied.</p> <p>1'b0: may be accepted 1'b1: will be denied</p>

GRF_SOC_STATUS0

Address: Operational Base + offset (0x014c)

SoC status register 0

Bit	Attr	Reset Value	Description
31	RO	0x0	reserved
30	RW	0x0	<p>msch_pwr_idle NOC memory scheduler idle state. "1" indicates idle.</p>
29	RO	0x0	<p>sys_pwr_idle NOC pd_cpu idle state. "1" indicates idle.</p>
28	RO	0x0	<p>gpu_pwr_idle NOC pd_gpu idle state. "1" indicates idle.</p>
27	RO	0x0	<p>vpu_pwr_idle NOC pd_vpu idle state. "1" indicates idle.</p>
26	RO	0x0	<p>vio_pwr_idle NOC pd_vio idle state. "1" indicates idle.</p>
25	RO	0x0	<p>peri_pwr_idle NOC pd_peri idle state. "1" indicates idle.</p>
24	RO	0x0	<p>core_pwr_idle NOC pd_core idle state. "1" indicates idle.</p>
23	RW	0x0	<p>msch_pwr_idleack NOC memory scheduler idle acknowledge. high valid.</p>
22	RO	0x0	<p>sys_pwr_idleack NOC pd_cpu idle acknowledge. high valid.</p>
21	RO	0x0	<p>gpu_pwr_idleack NOC pd_gpu idle acknowledge. high valid.</p>
20	RO	0x0	<p>vpu_pwr_idleack NOC pd_vpu idle acknowledge. high valid.</p>
19	RO	0x0	<p>vio_pwr_idleack NOC pd_vio idle acknowledge. high valid.</p>
18	RO	0x0	<p>peri_pwr_idleack NOC pd_peri idle acknowledge. high valid.</p>
17	RO	0x0	<p>core_pwr_idleack NOC pd_core idle acknowledge. high valid.</p>
16	RO	0x0	<p>host20_iddig Host 2.0 iddig status Host 2.0 iddig state. It will always be "0".</p>

Bit	Attr	Reset Value	Description
15:14	RO	0x0	<p>host20_linestate Host 2.0 linestate status This bus reflects the state of the single-ended receivers. In Suspend or Sleep mode, this bus is a combinatorial output (directly reflecting the current state of D- and D+, respectively).</p> <p>2'b11: SE1 (D+ high, D- high) 2'b10: K state for high-speed and full-speed USB traffic; J state for low-speed USB traffic (D+ low, D- high) 2'b01: J state for high-speed and full-speed USB traffic; K state for low-speed USB traffic (D+ high, D- low) 2'b00: SE0 (D+ low, D- low)</p> <p>During normal high-speed packet transfers, the line indicates a high-speed J state.</p>
13	RO	0x0	<p>host20_bvalid Host 2.0 bvalid status B-Device Session Valid Indicator Function: This controller signal is output from the USB 2.0 Session Valid comparator and indicates whether the session for a B-device is valid.</p> <p>1'b1: The session for the B-device is valid. 1'b0: The session for the B-device is not valid.</p>
12	RO	0x0	<p>host20_vbusvalid Host 2.0 vbus valid status VBUS Valid Indicator Function: This controller signal is output from the USB 2.0 VBUS Valid comparator and indicates whether the VBUS output is at a valid level.</p> <p>1'b1: The VBUS output is valid. 1'b0: The VBUS output is not valid.</p>
11	RO	0x1	<p>otg0_iddig otg0 iddig status 1'b0: indicate otg work as host 1'b1: indicate otg work as device</p>
10:9	RO	0x0	<p>otg0_linestate otg0 linestate status This bus reflects the state of the single-ended receivers. In Suspend or Sleep mode, this bus is a combinatorial output (directly reflecting the current state of D- and D+, respectively).</p> <p>2'b11: SE1 (D+ high, D- high) 2'b10: K state for high-speed and full-speed USB traffic; J state for low-speed USB traffic (D+ low, D- high) 2'b01: J state for high-speed and full-speed USB traffic; K state for low-speed USB traffic (D+ high, D- low) 2'b00: SE0 (D+ low, D- low)</p> <p>During normal high-speed packet transfers, the line indicates a high-speed J state.</p>

Bit	Attr	Reset Value	Description
8	RO	0x0	otg0_bvalid otg0 bvalid status B-Device Session Valid Indicator Function: This controller signal is output from the USB 2.0 Session Valid comparator and indicates whether the session for a B-device is valid. 1'b1: The session for the B-device is valid. 1'b0: The session for the B-device is not valid.
7	RO	0x0	otg0_vbusvalid otg0 vbus valid status VBUS Valid Indicator Function: This controller signal is output from the USB 2.0 VBUS Valid comparator and indicates whether the VBUS output is at a valid level. 1'b1: The VBUS output is valid. 1'b0: The VBUS output is not valid.
6:4	RO	0x0	pll_lock Pll lock status: bit2: generalpll_lock bit1: armpll_lock bit0: ddrpll_lock 1'b1: pll is lock 1'b0: pll is unlock
3:0	RO	0x0	timer_en_status bit0: timer 0 enable status bit1: timer 1 enable status bit2: timer 2 enable status bit3: timer 3 enable status High active

GRF_SOC_CON3

Address: Operational Base + offset (0x0154)

SoC control register 3

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~bit15 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
12:7	RW	0x3f	cvbsdac_gain_grf Gain setting digital input word for channel 0
6:4	RW	0x0	cvbsdac_enctr_grf Enable control pins for analog biasing Test. In normal mode set enctr[2..0]=000
3	RW	0x0	cvbsdac_ensc0_grf Sense comparator enable for cable connection detection of DAC0
2	RW	0x1	cvbsdac_endac0_grf Enable control pin to power up theDAC0
1	RW	0x1	cvbsdac_envbg_grf Enable control pin to power up internal bandgap. 1'b1: Internal bandgap is ON; 1'b0: Internal bandgap is OFF.
0	RW	0x0	cvbsdac_enextref_grf Enable control pin to allow internally generated bandgap to be probed in the analog pin 'vbg'. Under normal operation and envbg=1: 1'b1: Internal bandgap is passed to vbg pin; 1'b0: vbg pins is at high impedance.

GRF_UOC0_CON5

Address: Operational Base + offset (0x017c)

OTG control register

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit 0 ~ bit 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	otg0_bvalid_irq_pd otg0 bvalid interrupt pending bit write 1 to this bit, it will be cleared.
14	RW	0x0	otg0_bvalid_irq_en otg0 bvalid interrupt enable 1'b1: interrupt enable 1'b0: interrupt disable
13	RW	0x0	otg0_linestate_irq_pd otg0 linestate interrupt pending write 1 to this bit, it will be cleared.

Bit	Attr	Reset Value	Description
12	RW	0x0	otg0_linestate_irq_en otg0 linestate change interrupt enable 1'b1: interrupt enable 1'b0: interrupt disable
11	RW	0x0	otg0_usbphy_commonon OTG0 Common Block Power-Down Control This signal controls the power-down signals in the XO, Bias, and PLL blocks when the USB 2.0 PHY is in Suspend or Sleep mode 1'b1: In Suspend mode, the XO, Bias, and PLL blocks are powered down. In Sleep mode, the Bias and PLL blocks are powered down 1'b0: In Suspend mode, the XO, Bias, and PLL blocks remain powered in Suspend mode. In Sleep mode, if the reference clock is a crystal, the XO block remains powered
10	RW	0x0	otg0_iddig_status OTG0 control software iddig value 1'b0: host 1'b1: device
9	RW	0x0	otg0_iddig_sft_sel 1'b0: iddig to otg controller select usbphy output 1'b1: iddig to otg controller select grf_uoc0_con5[10]
8	RW	0x0	otg0_utmi_dmpulldown 1'b0: DM 15 KOhm pull down disabled 1'b1: DM 15 KOhm pull down enable
7	RW	0x0	otg0_utmi_dppulldown 1'b0: DP 15 KOhm pull down disabled 1'b1: DP 15 KOhm pull down enable
6	RW	0x0	otg0_utmi_termselect OTG0 USB Termination Select 1'b1: Full-speed terminations are enabled. 1'b0: High-speed terminations are enabled.
5:4	RW	0x0	otg0_utmi_xcvrselect OTG0 Transceiver Select 2'b11: Sends an LS packet on an FS bus or receives an LS packet. 2'b10: LS Transceiver 2'b01: FS Transceiver 2'b00: HS Transceiver
3:2	RW	0x0	otg0_utmi_opmode OTG0 UTMI+ Operational Mode Function: This controller bus selects the UTMI+ operational mode. 2'b11: Normal operation without SYNC or EOP generation. If the XCVRSEL bus is not set to 00 while OPMODE[1:0] is set to 11, USB PHY behavior is undefined. 2'b10: Disable bit stuffing and NRZI encoding 2'b01: Non-Driving 2'b00: Normal

Bit	Attr	Reset Value	Description
1	RW	0x0	otg0_utmi_suspend_n OTG0 Suspend Assertion 1'b1: Normal operating mode 1'b0: Suspend mode
0	RW	0x0	otg0_phy_soft_con_sel 1'b0: software control usb phy disable 1'b1: software control usb phy enable

GRF_UOC1_CON4

Address: Operational Base + offset (0x0190)

USB HOST 2.0 control register

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~bit15 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	bypasssel0 Transmitter Digital Bypass mode Enable. When 1, otg used as a UART2 port.
12	RW	0x0	bypassdmen0 DM0 Transmitter Digital Bypass Enable. high valid.
11	RW	0x0	host20_phydisable when 1, host 2.0 phy disable
10	RW	0x0	otg0_phydisable when 1, otgphy is disabled.
9:8	RW	0x0	otg0_scaledown Scale-Down Mode 2'b00: Disables all scale-downs. Actual timing values are used. Required for synthesis. 2'b01: Enables scale-down of all timing values except Device mode suspend and resume. These include: - Speed enumeration. - HNP/SRP. - Host mode suspend and resume. 2'b10: Enables scale-down of Device mode suspend and resume timing values only. 2'b11: Enables bit 0 and bit 1 scale-down timing values.

Bit	Attr	Reset Value	Description
7:6	RW	0x0	<p>host20_scaledown Host20 Scale-Down Mode 2'b00: Disables all scale-downs. Actual timing values are used. Required for synthesis.</p> <p>2'b01: Enables scale-down of all timing values except Device mode suspend and resume. These include:</p> <ul style="list-style-type: none"> - Speed enumeration. - HNP/SRP. - Host mode suspend and resume. <p>2'b10: Enables scale-down of Device mode suspend and resume timing values only.</p> <p>2'b11: Enables bit 0 and bit 1 scale-down timing values.</p>
5	RW	0x0	<p>host20_utmiotg_idpullup Host20 Analog ID Input Sample Enable Function: This controller signal controls ID line sampling.</p> <p>1'b1: ID pin sampling is enabled, and the IDDIG output is valid.</p> <p>1'b0: ID pin sampling is disabled, and the IDDIG output is not valid.</p>
4	RW	0x1	<p>host20_utmiotg_dppulldown Host20 D+ Pull-Down Resistor Enable</p>
3	RW	0x1	<p>host20_utmiotg_dmpulldown Host20 D- Pull-Down Resistor Enable</p>
2	RW	0x1	<p>host20_utmiotg_drvvbus Host20 Drive VBUS</p> <p>1'b1: The VBUS Valid comparator is enabled.</p> <p>1'b0: The VBUS Valid comparator is disabled.</p>
1	RW	0x0	<p>host20_utmisrp_chrgvbus Host20 VBUS Input Charge Enable</p>
0	RW	0x0	<p>host20_utmisrp_dischrgvbus Host20 VBUS Input Discharge Enable</p>

GRF_UOC1_CON5

Address: Operational Base + offset (0x0194)

USB HOST 2.0 control register

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	<p>write_enable bit 0 ~ bit 15 write enable</p> <p>When bit 16=1, bit 0 can be written by software;</p> <p>When bit 16=0, bit 0 cannot be written by software;</p> <p>When bit 17=1, bit 1 can be written by software;</p> <p>When bit 17=0, bit 1 cannot be written by software;</p> <p>.....</p> <p>When bit 31=1, bit 15 can be written by software;</p> <p>When bit 31=0, bit 15 cannot be written by software.</p>

Bit	Attr	Reset Value	Description
15	RW	0x0	host20_linestate_irq_pd Host2.0 linestate interrupt pending write 1 to this bit, it will be cleared.
14	RW	0x0	host20_linestate_irq_en Host2.0 linestate interrupt enable
13:9	RO	0x0	reserved
8	RW	0x0	host20_utmi_dmpulldown 1'b0: DM 15 KOhm pull down disabled 1'b1: DM 15 KOhm pull down enable
7	RW	0x0	host20_utmi_dppulldown 1'b0: DP 15 KOhm pull down disabled 1'b1: DP 15 KOhm pull down enable
6	RW	0x0	host20_utmi_termselect Host2.0 USB Termination Select 1'b1: Full-speed terminations are enabled 1'b0: High-speed terminations are enabled
5:4	RW	0x0	host20_utmi_xcvrselect Host2.0 Transceiver Select 2'b11: Sends an LS packet on an FS bus or receives an LS packet. 2'b10: LS Transceiver 2'b01: FS Transceiver 2'b00: HS Transceiver
3:2	RW	0x0	host20_utmi_opmode Host2.0 UTMI+ Operational Mode Function: This controller bus selects the UTMI+ operational mode. 2'b11: Normal operation without SYNC or EOP generation. If the XCVRSEL bus is not set to 00 while OPMODE[1:0] is set to 11, USB PHY behavior is undefined. 2'b10: Disable bit stuffing and NRZI encoding 2'b01: Non-Driving 2'b00: Normal
1	RW	0x0	host20_utmi_suspend_n Host2.0 Suspend Assertion 1'b1: Normal operating mode 1'b0: Suspend mode
0	RW	0x0	host20_phy_soft_con_sel 1'b0: software control usb phy disable 1'b1: software control usb phy enable

GRF_DDRC_STAT

Address: Operational Base + offset (0x019c)

DDRC status register

Bit	Attr	Reset Value	Description
31:21	RW	0x000	gpu_idle gpu idle staus

Bit	Attr	Reset Value	Description
20	RW	0x1	ddrupctl_c_active confirm that system external to PCTL can accept a Low-power request. low valid.
19	RW	0x1	ddrupctl_c_sysack PCTL low-power request status response. low valid.
18:16	RO	0x0	ddrupctl_stat Current state of the protocol controller 3'b000: Init_mem 3'b001: Config 3'b010: Config_req 3'b011: Access 3'b100: Access_req 3'b101: Low_power 3'b110: Low_power_entry_req 3'b111: Low_power_exit_req
15:0	RO	0x0000	ddrupctl_bbflags Bank busy indication NIF output vector which provides combined information about the status of each memory bank. The de-assertion is based on when precharge, activates, reads/writes. Bit0 indication Bank0 busy, bit1 indication Bank1 busy, and so on.

GRF_UOC_CON6

Address: Operational Base + offset (0x01a0)

USB control register

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit 0 ~ bit 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	otg0_bvalid_f otg0 vbus bvalid status after filter B-Device Session Valid Indicator Function: This controller signal is output from the USB 2.0 Session Valid comparator and indicates whether the session for a B-device is valid. 1'b1: The session for the B-device is valid. 1'b0: The session for the B-device is not valid.

Bit	Attr	Reset Value	Description
14	RO	0x1	otg0_iddig_f otg0 iddig status after filter 1'b0: otg works as a host 1'b1: otg works as a device.
13:10	RO	0x0	reserved
9:8	RW	0x2	host20_linestate_filter_sel Host2.0 linestate filter time select 2'b00: 100us 2'b01: 500us 2'b10: 1ms 2'b11: 10ms
7:6	RW	0x2	otg0_linestate_filter_sel otg0 linestate filter time select 2'b00: 100us 2'b01: 500us 2'b10: 1ms 2'b11: 10ms
5:4	RW	0x2	otg0_id_filter_sel otg0 iddig filter time select 2'b00: 5ms 2'b01: 15ms 2'b10: 35ms 2'b11: 50ms
3	RW	0x0	otg0_id_fall_irq_pd otg0 id negedge interrupt pending write 1 to this bit, it will be cleared.
2	RO	0x0	otg0_id_fall_irq_en otg0 id negedge interrupt enable 1'b1 : interrupt enable 1'b0 : interrupt disable
1	RW	0x0	otg0_id_rise_irq_pd otg0 id posedge interrupt pending write 1 to this bit, it will be cleared.
0	RW	0x0	otg0_id_rise_irq_en otg0 id posedge interrupt enable 1'b1: interrupt enable 1'b0: interrupt disable

GRF_SOC_STATUS1

Address: Operational Base + offset (0x01a4)

SoC status register 1

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved

Bit	Attr	Reset Value	Description
1	RW	0x0	vdac_enctr3_grf Enable control pins for analog biasing Test. In normal mode set enctr[2..0]=000
0	RW	0x0	vdac_cableout0_grf CVBS DAC cable out flag to indicate that the cable of the DAC is unplugged. cableoutx=1 indicates that the DAC is not connected to the outside.

GRF_CPU_CON0

Address: Operational Base + offset (0x01a8)

CPU control register 0

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0 ~ bit15 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	deviceen_dap Enabling access to the connected debug device or memory system 1'b0: disable 1'b1: enable
12	RW	0x0	I2rstdisable A7 I2rstdisable bit control
11:10	RO	0x0	reserved
9:8	RW	0x0	I1rstdisable A7 I1rstdisable bit control Every bit for one core, bit1 is for core1, bit0 is for core0.
7:3	RO	0x0	reserved
2:0	RW	0x2	ema_l2_data A7 L2 memory EMA control

GRF_CPU_CON1

Address: Operational Base + offset (0x01ac)

CPU control register 1

Bit	Attr	Reset Value	Description
------------	-------------	--------------------	--------------------

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	<p>write_enable</p> <p>bit0 ~ bit15 write enable</p> <p>When bit 16=1, bit 0 can be written by software;</p> <p>When bit 16=0, bit 0 cannot be written by software;</p> <p>When bit 17=1, bit 1 can be written by software;</p> <p>When bit 17=0, bit 1 cannot be written by software;</p> <p>.....</p> <p>When bit 31=1, bit 15 can be written by software;</p> <p>When bit 31=0, bit 15 cannot be written by software.</p>
15:10	RO	0x0	reserved
9:8	RW	0x0	<p>cfgte</p> <p>Controls processor state for exception handling (TE bit) at reset</p>
7:6	RO	0x0	reserved
5:4	RW	0x0	<p>vinithi</p> <p>Cortex-A7 vinithi bit control.</p> <p>location of the exception vectors at reset. Sampled during reset.</p> <p>1'b0: 0x0000_0000</p> <p>1'b1: 0xffff_0000</p>
3:2	RO	0x0	reserved
1:0	RW	0x0	<p>cfgend</p> <p>One bit for each processor.</p> <p>1'b0: Little-endian</p> <p>1'b1: Big-endian</p>

GRF_CPU_CON2

Address: Operational Base + offset (0x01b0)

CPU control register 2

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	<p>wirte_enable</p> <p>bit0 ~ bit15 write enable</p> <p>When bit 16=1, bit 0 can be written by software;</p> <p>When bit 16=0, bit 0 cannot be written by software;</p> <p>When bit 17=1, bit 1 can be written by software;</p> <p>When bit 17=0, bit 1 cannot be written by software;</p> <p>.....</p> <p>When bit 31=1, bit 15 can be written by software;</p> <p>When bit 31=0, bit 15 cannot be written by software.</p>
15:9	RO	0x0	reserved
8	RW	0x0	<p>cfgsdisable</p> <p>Disables write access to some secure GIC registers</p> <p>When CFGSDISABLE is asserted, the GIC prevents writes to any register locations that control the operating state of an LSPI</p> <p>1'b0: enable</p> <p>1'b1: disable</p>

Bit	Attr	Reset Value	Description
7	RW	0x0	evento_clear Event output Evento is active when one SEV instruction is executed. This bit used to clear evento signal. 1'b0: un-clear 1'b1: clear
6	RW	0x0	eventi Event input for processor wake-up from WFE state This pin must be asserted for at least one CLKIN clock cycle. When this signal is asserted, it acts as a WFE wake-up event to all the processors in the multiprocessor device
5	RW	0x1	dbgselfaddrv Debug self-address offset valid 1'b0: unvalid 1'b1: valid
4	RW	0x1	dbgromaddrv Debug ROM physical address valid 1'b0: unvalid 1'b1: valid
3	RW	0x1	spniden Secure privileged non-invasive debug enable
2	RW	0x1	niden Non-invasive debug enable
1	RW	0x1	spiden Secure privileged invasive debug enable
0	RW	0x1	dbgen Debug enable

GRF_CPU_STATUS0

Address: Operational Base + offset (0x01c0)

CPU status register 0

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4:3	RO	0x0	smpnamp Signals AMP or SMP mode for each Cortex-A7 processor. 1'b0: Asymmetric. 1'b1: Symmetric.
2	RO	0x1	jtagnw_dap coresight jtagnw signal status 1'b1: JTAG is selected. 1'b0: SWD is selected.
1	RO	0x1	jtagtop_dap coresight jtagtop signal status "1" means jtag state machine is in one of the top four modes: test-logic-reset, run-test/idle, select-DR-scan, select-IR-scan.

Bit	Attr	Reset Value	Description
0	RO	0x0	evento_rising_edge evento signal rising edge status

GRF_CPU_STATUS1

Address: Operational Base + offset (0x01c4)

CPU status register 2

Bit	Attr	Reset Value	Description
31:7	RO	0x0	reserved
6:5	RW	0x0	standbywfi core WFI status, 2bit corresponding 2 cores
4:3	RO	0x0	reserved
2:1	RW	0x0	standbywfe core WFE status, 2bit corresponding 2 cores
0	RO	0x0	standbywfil2 L2 WFI status

GRF_OS_REG0

Address: Operational Base + offset (0x01c8)

Software OS register 0

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	os_reg software OS register

GRF_OS_REG1

Address: Operational Base + offset (0x01cc)

Software OS register 1

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	os_reg software OS register

GRF_OS_REG2

Address: Operational Base + offset (0x01d0)

Software OS register 2

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	os_reg software OS register

GRF_OS_REG3

Address: Operational Base + offset (0x01d4)

Software OS register 3

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	os_reg software OS register

GRF_OS_REG4

Address: Operational Base + offset (0x01d8)

Software OS register 4

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	os_reg software OS register

GRF_OS_REG5

Address: Operational Base + offset (0x01dc)

Software OS register 5

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	os_reg software OS register

GRF_OS_REG6

Address: Operational Base + offset (0x01e0)

Software OS register 6

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	os_reg software OS register

GRF_OS_REG7

Address: Operational Base + offset (0x01e4)

Software OS register 7

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	os_reg software OS register

GRF_PVTM_CON0

Address: Operational Base + offset (0x0200)

PVTM control register 0

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~bit15 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	RW	0x0	pvtm_video_osc_en pd_video PVT monitor oscillator enable 1'b1: enable 1'b0: disable

Bit	Attr	Reset Value	Description
4	RW	0x0	pvtm_video_start pd_video PVT monitor start control
3	RW	0x0	pvtm_gpu_osc_en pd_gpu PVT monitor oscillator enable 1'b1: enable 1'b0: disable
2	RW	0x0	pvtm_gpu_start pd_gpu PVT monitor start control
1	RW	0x0	pvtm_core_osc_en pd_core PVT monitor oscillator enable 1'b1: enable 1'b0: disable
0	RW	0x0	pvtm_core_start pd_core PVT monitor start control

GRF_PVTM_CON1

Address: Operational Base + offset (0x0204)

PVTM control register 1

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	pvtm_core_cal_cnt pd_core pvtm calculator counter

GRF_PVTM_CON2

Address: Operational Base + offset (0x0208)

PVTM control register 2

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	pvtm_gpu_cal_cnt pd_gpu pvtm calculator counter

GRF_PVTM_CON3

Address: Operational Base + offset (0x020c)

PVTM control register 3

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	pvtm_video_cal_cnt pd_video pvtm calculator counter

GRF_PVTM_STATUS0

Address: Operational Base + offset (0x0210)

PVTM status register 0

Bit	Attr	Reset Value	Description
31:3	RO	0x0	reserved
2	RO	0x0	pvtm_video_freq_done pd_video pvtm frequency calculate done status
1	RO	0x0	pvtm_core_freq_done pd_core pvtm frequency calculate done status

Bit	Attr	Reset Value	Description
0	RO	0x0	pvtm_gpu_freq_done pd_gpu pvtm frequency calculate done status

GRF_PVTM_STATUS1

Address: Operational Base + offset (0x0214)

PVTM status register 1

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	pvtm_core_freq_cnt pd_core pvtm frequency count

GRF_PVTM_STATUS2

Address: Operational Base + offset (0x0218)

PVTM status register 2

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	pvtm_gpu_freq_cnt pd_gpu pvtm frequency count

GRF_PVTM_STATUS3

Address: Operational Base + offset (0x021c)

PVTM status register 3

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	pvtm_video_freq_cnt pd_video pvtm frequency count

GRF_DFI_WRNUM

Address: Operational Base + offset (0x0220)

DFI write number register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	dfi_eff_wr_num the total number of write operation on DFI interface.

GRF_DFI_RDNUM

Address: Operational Base + offset (0x0224)

DFI read number register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	dfi_eff_rd_num the total number of read operation on DFI interface.

GRF_DFI_ACTNUM

Address: Operational Base + offset (0x0228)

DFI active number register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	dfi_eff_act_num the total number of active operation on DFI interface.

GRF_DFI_TIMERVAL

Address: Operational Base + offset (0x022c)

DFI work time

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	dfi_timer_val the total time for DFI monitor works.

GRF_NIF_FIFO0

Address: Operational Base + offset (0x0230)

NIF status register 0

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	nif0_fifo0 status for msch4 signals. It will not be cleared by system reset. bit 10 ~ bit 0 : msch4_n_acol[10:0] bit 11 ~ bit 13 : msch4_n_abank[2:0] bit 14 ~ bit 29 : msch4_n_arow[15:0] bit 31 ~ bit 30 : msch4_n_arank_sel[1:0]

GRF_NIF_FIFO1

Address: Operational Base + offset (0x0234)

NIF status register 1

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	nif0_fifo1 status for msch4 signals. It will not be cleared by system reset. bit 10 ~ bit 0 : msch4_n_acol[10:0] bit 11 ~ bit 13 : msch4_n_abank[2:0] bit 14 ~ bit 29 : msch4_n_arow[15:0] bit 31 ~ bit 30 : msch4_n_arank_sel[1:0]

GRF_NIF_FIFO2

Address: Operational Base + offset (0x0238)

NIF status register 2

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	nif0_fifo2 status for msch4 signals. It will not be cleared by system reset. bit 10 ~ bit 0 : msch4_n_acol[10:0] bit 11 ~ bit 13 : msch4_n_abank[2:0] bit 14 ~ bit 29 : msch4_n_arow[15:0] bit 31 ~ bit 30 : msch4_n_arank_sel[1:0]

GRF_NIF_FIFO3

Address: Operational Base + offset (0x023c)

NIF status register 3

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	nif0_fifo3 status for msch4 signals. It will not be cleared by system reset. bit 10 ~ bit 0 : msch4_n_acol[10:0] bit 11 ~ bit 13 : msch4_n_abank[2:0] bit 14 ~ bit 29 : msch4_n_arow[15:0] bit 31 ~ bit 30 : msch4_n_arank_sel[1:0]

GRF_USBPHY0_CON0

Address: Operational Base + offset (0x0280)

usbphy control register

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~bit15 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	RW	0x4	squel_trigger_con bit2 ~ bit0 of squel_trigger_con. 4'b0000: 112.5mV 4'b1001: 162.5mV 4'b1011: 175mV 4'b1100: 150mV(default) 4'b1110: 125mV
12:11	RW	0x0	non_driving Registers for non-driving state control. non-driving state is controlled by op-mode by default, when bit[11] is configured with "1", user can control non-driving state through bit[12].
10:8	RW	0x6	tx_clk_phase_con USB Tx Clock phase configure, 3'b000 represent the earliest phase, and 3'b111 the latest, single step delay is 256ps
7:5	RW	0x0	rx_clk_phase_con USB Rx Clock phase configure, 3'b000 represent the earliest phase, and 3'b111 the latest, single step delay is 256ps
4:3	RW	0x3	fls_eye_height FS/LS eye height configure, 2'b00 represent the largest slew rate, 2'b11 represent the smallest slew rate

Bit	Attr	Reset Value	Description
2:0	RW	0x0	<p>hs_eye_diag_adjust HS eye diagram adjust, open HS pre-emphasize function to increase HS slew rate, only used when large cap loading is attached.</p> <p>3'b001: open pre-emphasize in sof or eop state 3'b010: open pre-emphasize in chirp state 3'b100: open pre-emphasize in non-chirp state 3'b111: always open pre-emphasize other combinations: reserved</p>

GRF_USBPHY0_CON1

Address: Operational Base + offset (0x0284)

usbphy control register

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	<p>write_enable bit0~bit15 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.</p>
15:13	RW	0x7	<p>hs_eye_height bit2 ~ bit 0 of hs_eye_height. HS eye height tuning, more zeros represent bigger eye, more ones represent smaller eye.</p>
12:3	RO	0x0	reserved
2	RW	0x1	<p>current_comp_en Enable current compensation, active high.</p>
1	RW	0x1	<p>res_comp_en Enable resistance compensation, active high.</p>
0	RW	0x1	<p>squel_trigger_con bit3 of squel_trigger_con. 4'b0000: 112.5mV 4'b1001: 162.5mV 4'b1011: 175mV 4'b1100: 150mV(default) 4'b1110: 125mV</p>

GRF_USBPHY0_CON2

Address: Operational Base + offset (0x0288)

usbphy control register

Bit	Attr	Reset Value	Description
-----	------	-------------	-------------

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	<p>write_enable bit0~bit15 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.</p>
15	RW	0x1	<p>odt_compensation bit 0 of odt_compensation. ODT Compensation voltage reference 3'b000: 200mV 3'b001: 187.5mV(default) 3'b010: 225mV 3'b110: 175mV 3'b111: 162.5mV</p>
14:13	RW	0x0	<p>voltage_tolerance_adjust 5V tolerance detection reference adjust, 11 represent the highest trigger point, keeping the default value is greatly appreciated</p>
12	RO	0x0	reserved
11:10	RW	0x0	<p>auto_compensation_bypass auto compensation bypass, 2'b11 will bypass current and ODT compensation, customers can set the driver strength and current manually. For larger HS eye height, customer can give more "0" for hs_eye_height; For larger HS/FS/LS slew rate, give more "1" for hfs_driver_strength.</p>
9:5	RW	0x15	<p>hfs_driver_strength HS/FS driver strength tuning, 5'b11111 represent the largest slew rate and 5'b00000 represents the smallest slew rate</p>
4:0	RW	0x0a	<p>hs_eye_height bit7 ~ bit 3 of hs_eye_height. HS eye height tuning, more zeros represent bigger eye, more ones represent smaller eye</p>

GRF_USBPHY0_CON3

Address: Operational Base + offset (0x028c)
usbphy control register

Bit	Attr	Reset Value	Description
------------	-------------	--------------------	--------------------

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	<p>write_enable bit0~bit15 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.</p>
15	RO	0x0	reserved
14	RW	0x0	<p>vol_toleran_det_con 5V tolerance detection function controlling bit through registers, only active when GRF_USBPHY0_CON4[1] is set "1".</p>
13:10	RO	0x0	reserved
9	RW	0x1	<p>odt_auto_refresh A port ODT auto refresh bypass, active low, this register should only be used when auto_compensation_bypass were set to "11". In bypass mode, customer can configure driver strength through hfs_driver_strength.</p>
8	RW	0x0	<p>bg_out_voltage_adjust BG output voltage reference adjust, keeping the default value is greatly appreciated.</p>
7:5	RW	0x0	<p>compen_current_ref compensation current tuning reference 3'b000: 200mV(default) 3'b001: 187.5mV 3'b010: 225mV 3'b110: 175mV 3'b111: 162.5mV</p>
4:2	RW	0x0	<p>bias_current_ref bias current tuning reference 3'b000: 400mV(default) 3'b001: 362.5mV 3'b010: 350mV 3'b101: 425mV 3'b111: 450mV</p>
1:0	RW	0x0	<p>odt_compensation bit 2 ~ bit 1 of odt_compensation. ODT Compensation voltage reference 3'b000: 200mV 3'b001: 187.5mV(default) 3'b010: 225mV 3'b110: 175mV 3'b111: 162.5mV</p>

GRF_USBPHY0_CON4

Address: Operational Base + offset (0x0290)

usbphy control register

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~bit15 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	RW	0x1	bypass_5v_tolerance_det Bypass 5V tolerance detection function, active high
0	RO	0x0	reserved

GRF_USBPHY0_CON5

Address: Operational Base + offset (0x0294)

usbphy control register

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~bit15 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	RO	0x0	reserved

GRF_USBPHY0_CON6

Address: Operational Base + offset (0x0298)

usbphy control register

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~bit15 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
15:13	RW	0x0	session_end_con session end reference tuning
12:10	RW	0x0	b_session_con B_session valid reference tuning
9:7	RW	0x0	a_session_con A_session valid reference tuning
6	RW	0x0	force_vbus_valid force output vbus_valid asserted, active high
5	RW	0x0	force_session_end_val force output session end asserted, active high
4	RW	0x0	force_b_session_val force output B_session valid asserted, active high
3	RW	0x0	force_a_session_val force output A_session valid asserted, active high
2	RW	0x1	turn_off_diff_receiver Turn off differential receiver in suspend mode to save power, active low.
1:0	RO	0x0	reserved

GRF_USBPHY0_CON7

Address: Operational Base + offset (0x029c)

usbphy control register

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~bit15 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
14:11	RW	0xd	host_discon_con HOST disconnect detection trigger point configure, only used in HOST mode 4'b0000: 575mV 4'b0001: 600mV 4'b1001: 625mV 4'b1101: 650mV(default)
10:8	RO	0x0	reserved
7	RW	0x1	bypass_squelch_trigger bypass squelch trigger point auto configure in chirp modes, active high

Bit	Attr	Reset Value	Description
6	RW	0x1	half_bit_pre_empha_en half bit pre-emphasize enable, active high. "1" represent half bit pre-emphasis, "0" for full bit
5:3	RO	0x0	reserved
2:0	RW	0x0	vbus_valid_con vbus_valid reference tuning

GRF_USBPHY1_CON0

Address: Operational Base + offset (0x02a0)

usbphy control register

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~bit15 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	RW	0x4	squel_trigger_con bit 2 ~ bit 0 of squel_trigger_con. 4'b0000: 112.5mV 4'b1001: 162.5mV 4'b1011: 175mV 4'b1100: 150mV(default) 4'b1110: 125mV
12:11	RW	0x0	non_driving Registers for non-driving state control. non-driving state is controlled by op-mode by default, when bit[11] is configured with "1", user can control non-driving state through bit[12].
10:8	RW	0x6	tx_clk_phase_con USB Tx Clock phase configure, 3'b000 represent the earliest phase, and 3'b111 the latest, single step delay is 256ps
7:5	RW	0x0	rx_clk_phase_con USB Rx Clock phase configure, 3'b000 represent the earliest phase, and 3'b111 the latest, single step delay is 256ps
4:3	RW	0x3	fsls_eye_height FS/LS eye height configure, 2'b00 represent the largest slew rate, 2'b11 represent the smallest slew rate

Bit	Attr	Reset Value	Description
2:0	RW	0x0	<p>hs_eye_diag_adjust HS eye diagram adjust, open HS pre-emphasize function to increase HS slew rate, only used when large cap loading is attached.</p> <p>3'b001: open pre-emphasize in sof or eop state 3'b010: open pre-emphasize in chirp state 3'b100: open pre-emphasize in non-chirp state 3'b111: always open pre-emphasize other combinations : reserved</p>

GRF_USBPHY1_CON1

Address: Operational Base + offset (0x02a4)

usbphy control register

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	<p>write_enable bit0~bit15 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.</p>
15:13	RW	0x7	<p>hs_eye_height bit2 ~ bit 0 of hs_eye_height. HS eye height tuning, more zeros represent bigger eye, more ones represent smaller eye</p>
12:3	RO	0x0	reserved
2	RW	0x1	<p>current_comp_en Enable current compensation, active high.</p>
1	RW	0x1	<p>res_comp_en Enable resistance compensation, active high.</p>
0	RW	0x1	<p>squel_trigger_con bit 3 of squel_trigger_con. 4'b0000: 112.5mV 4'b1001: 162.5mV 4'b1011: 175mV 4'b1100: 150mV(default) 4'b1110: 125mV</p>

GRF_USBPHY1_CON2

Address: Operational Base + offset (0x02a8)

usbphy control register

Bit	Attr	Reset Value	Description
-----	------	-------------	-------------

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	<p>write_enable bit0~bit15 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.</p>
15	RW	0x1	<p>odt_compensation bit 0 of odt_compensation. ODT Compensation voltage reference 3'b000: 200mV 3'b001: 187.5mV(default) 3'b010: 225mV 3'b110: 175mV 3'b111: 162.5mV</p>
14:13	RW	0x0	<p>voltage_tolerance_adjust 5V tolerance detection reference adjust, 2'b11 represent the highest trigger point, keeping the default value is greatly appreciated</p>
12	RO	0x0	reserved
11:10	RW	0x0	<p>auto_compensation_bypass auto compensation bypass, 2'b11 will bypass current and ODT compensation, customers can set the driver strength and current manually. For larger HS eye height, customer can give more "0" for hs_eye_height; For larger HS/FS/LS slew rate, give more "1" for hfs_driver_strength.</p>
9:5	RW	0x15	<p>hfs_driver_strength HS/FS driver strength tuning , 5'b11111 represent the largest slew rate and 5'b10000 represents the smallest slew rate</p>
4:0	RW	0x0a	<p>hs_eye_height bit7 ~ bit 3 of hs_eye_height. HS eye height tuning, more zeros represent bigger eye, more ones represent smaller eye</p>

GRF_USBPHY1_CON3

Address: Operational Base + offset (0x02ac)
usbphy control register

Bit	Attr	Reset Value	Description
-----	------	-------------	-------------

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	<p>write_enable bit0~bit15 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.</p>
15	RO	0x0	reserved
14	RW	0x0	<p>vol_toleran_det_con 5V tolerance detection function controlling bit through registers, only active when GRF_USBPHY0_CON4[1] is set 1'b1.</p>
13:10	RO	0x0	reserved
9	RW	0x1	<p>odt_auto_refresh A port ODT auto refresh bypass, active low, this register should only be used when auto_compensation_bypass were set to 2'b11. In bypass mode, customer can configure driver strength through hfs_driver_strength.</p>
8	RW	0x0	<p>bg_out_voltage_adjust BG output voltage reference adjust, keeping the default value is greatly appreciated.</p>
7:5	RW	0x0	<p>compen_current_ref compensation current tuning reference 3'b000: 200mV(default) 3'b001: 187.5mV 3'b010: 225mV 3'b110: 175mV 3'b111: 162.5mV</p>
4:2	RW	0x0	<p>bias_current_ref bias current tuning reference 3'b000: 400mV(default) 3'b001: 362.5mV 3'b010: 350mV 3'b101: 425mV 3'b111: 450mV</p>
1:0	RW	0x0	<p>odt_compensation bit 2 ~ bit 1 of odt_compensation. ODT Compensation voltage reference 3'b000: 200mV 3'b001: 187.5mV(default) 3'b010: 225mV 3'b110: 175mV 3'b111: 162.5mV</p>

GRF_USBPHY1_CON4

Address: Operational Base + offset (0x02b0)

usbphy control register

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~bit15 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	RW	0x1	bypass_5v_tolerance_det Bypass 5V tolerance detection function, active high
0	RO	0x0	reserved

GRF_USBPHY1_CON5

Address: Operational Base + offset (0x02b4)

usbphy control register

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~bit15 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	RO	0x0	reserved

GRF_USBPHY1_CON6

Address: Operational Base + offset (0x02b8)

usbphy control register

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~bit15 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
15:13	RW	0x0	session_end_con session end reference tuning
12:10	RW	0x0	b_session_con B_session valid reference tuning
9:7	RW	0x0	a_session_con A_session valid reference tuning
6	RW	0x0	force_vbus_valid force output vbus_valid asserted, active high
5	RW	0x0	force_session_end_val force output session end asserted, active high
4	RW	0x0	force_b_session_val force output B_session valid asserted, active high
3	RW	0x0	force_a_session_val force output A_session valid asserted, active high
2	RW	0x1	turn_off_diff_receiver Turn off differential receiver in suspend mode to save power, active low.
1:0	RO	0x0	reserved

GRF_USBPHY1_CON7

Address: Operational Base + offset (0x02bc)

usbphy control register

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~bit15 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
14:11	RW	0xd	host_discon_con HOST disconnect detection trigger point configure, only used in HOST mode 4'b0000: 575mV 4'b0001: 600mV 4'b1001: 625mV 4'b1101: 650mV(default)
10:8	RO	0x0	reserved
7	RW	0x1	bypass_squelch_trigger bypass squelch trigger point auto configure in chirp modes, active high

Bit	Attr	Reset Value	Description
6	RW	0x1	half_bit_pre_empha_en half bit pre-emphasize enable, active high. "1" represent half bit pre-emphasis, "0" for full bit
5:3	RO	0x0	reserved
2:0	RW	0x0	vbus_valid_con vbus_valid reference tuning

GRF_CHIP_TAG

Address: Operational Base + offset (0x0300)

chip tag register

Bit	Attr	Reset Value	Description
31:0	RO	0x00003012	chip_tag Chip tag

GRF_SDMMC_DET_CNT

Address: Operational Base + offset (0x0304)

SDMMC detectn filter count

Bit	Attr	Reset Value	Description
31:20	RO	0x0	reserved
19:0	RW	0x00cff	sdmmc_detn_cnt SDMMC detectn filter count. Number of XIN24M clock, used for the detect of sdmmc card when grf_force_jtag=1

GRF_EFUSE_PRG

Address: Operational Base + offset (0x037c)

eFuse program enable register

Bit	Attr	Reset Value	Description
31:14	RO	0x0	reserved
13	WO	0x0	eFuse program enable 1'b1: enable program 1'b0: disable program
12:0	RO	0x0	reserved

Chapter 6 DMC (Dynamic Memory Interface)

6.1 Overview

The DMC includes two section: dynamic ram protocol controller (PCTL) and PHY controller. The PCTL SoC application bus interface supports a lowest-latency native application interface (NIF). To maximize data transfer efficiency, NIF commands transfer data without flow control. To simplify command processing, the NIF accepts addresses in rank, bank, row, column format.

The DDR PHY provides control features to ease the customer implementation of digitally controlled features of the PHY such as initialization, DQS gate training, and programmable configuration controls. The DDR PHY has built-in self-test features to provide support for production testing of the compatible PHY. It also provides a DFI 2.1 interface to the PHY.

The DMC supports the following features:

- Complete, integrated DDR3 solution
- DFI 2.1 interface compatibility
- Up to 800 Mbps in 1:1 frequency ratio, using a 400MHz controller clock and 400MHz memory clock.
- Support for x8, x16 memories, for a total memory data path width of 16 bits
- Up to 2 memory ranks; devices within a rank tie to a common chip select
- Up to 8 open memory banks, maximum of eight per rank
- Per-NIF transaction controllable bank management policies: open-page, close-page
- Low area, low power architecture with minimal buffering on the data, avoiding duplication of storage resources within the system
- PCTL NIF slave interface facilitates easy integration with an external scheduler or standard on-chip buses
- Efficient DDR protocol implementation with in-order column (Read and Write) commands and out-of-order Activate and Precharge commands
- Three clock cycles best case command latency (best case is when a command is to an open page and the shift array in the PCTL is empty)
- 1T or 2T memory command timing
- Automatic power-down and self-refresh entry and exit
- Software and hardware driven self-refresh entry and exit
- Programmable memory initialization
- Partial population of memories, where not all DDR byte lanes are populated with memory chips
- Programmable per rank memory ODT (On-Die Termination) support for reads and writes
- APB interface for controller software-accessible registers
- Programmable data training interface:
 - Assists in training of the data eye of the memory channel
 - Provides a method for testing large sections of memory
 - Automatic DQS gate training
- At-speed built-in-self-test (BIST) loopback testing on both the address and data channels for DDR PHYs
- PHY control and configuration registers
- Optional, additional JTAG interface to configure registers

6.2 Block Diagram

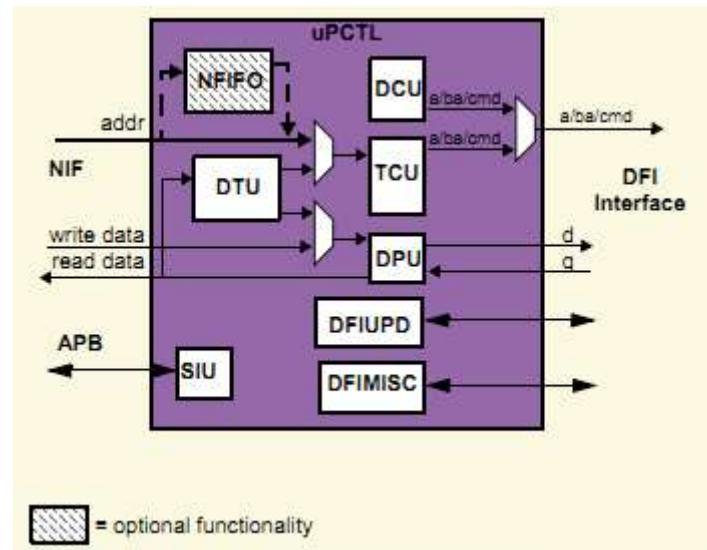


Fig. 6-1 Protocol controller architecture

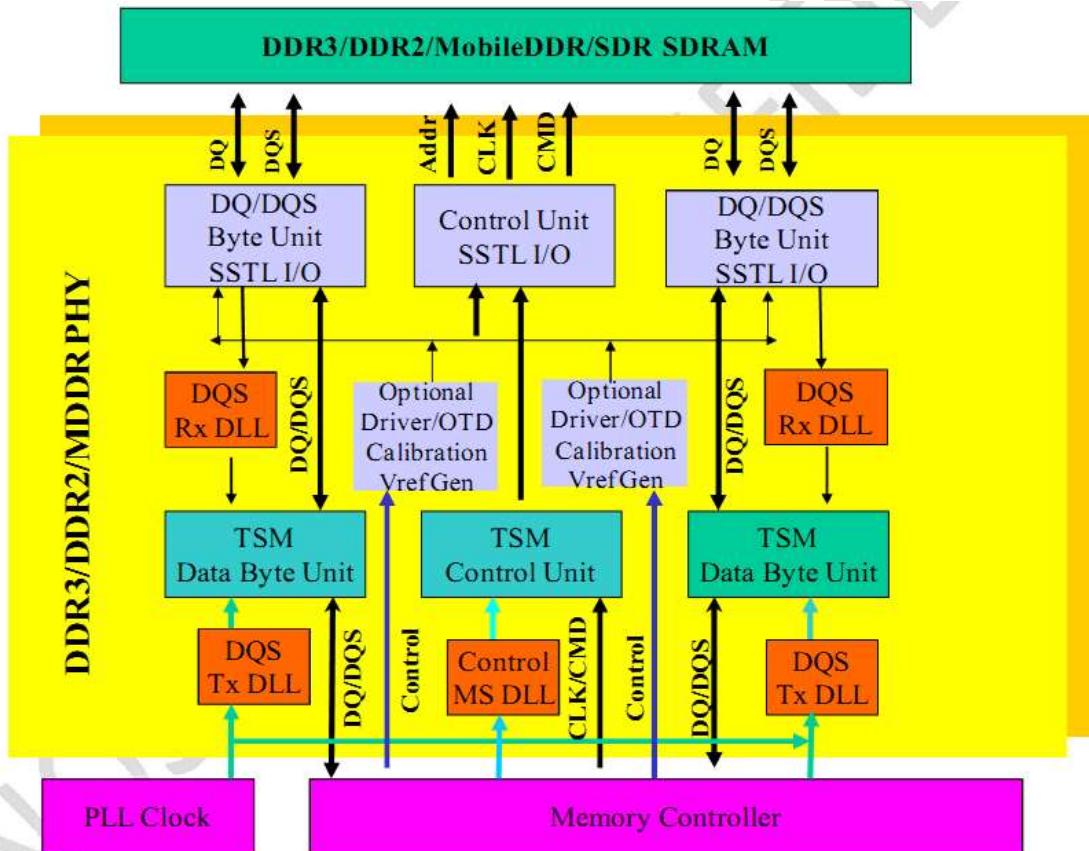


Fig. 6-2 PHY controller architecture

6.3 Function Description

6.3.1 Protocol controller (PCTL)

PCTL operations are defined in terms of the current state of the Operational State Machine. Software can move PCTL in any of the operational states by issuing commands via the SCTL register. Transitions from one operational state to the other occur pass through a “transitional” state. Transitional states are exited automatically by the PCTL after all the necessary actions required to change operational state have been completed. The current operational state of PCTL is reported by the STAT register and is also available from the p_ctl_stat output.

PCTL supports the following operational states:

- Init_mem - This state is the default state entered after reset. All writable registers can be programmed. While in this state software can program PCTL and initialize the PHY and the memories. The memories are not refreshed and data that has previously been written to the memories may be lost as a result. The Init_mem state is also used when it is desirable to stop any automatic PCTL function that directly affects the memories, like Power Down and Refresh, or when a software reset of the memory subsystem has to be executed.
- ConFig - This state is used to suspend temporarily the normal NIF traffic and allow software to reprogram PCTL and memories if necessary, while still keeping active the periodic generation of Refresh cycles to the memories. Power Down entry and exit sequences are possible while in ConFig state.
- Access - This is the operational state where NIF transactions are accepted by the PCTL and converted into memory read and writes. None of the registers can be programmed except SCFG, SCTL, ECCCLR and DTU* registers.
- Low_power - Memories are in self-refresh mode. The PCTL does not generate refresh cycles while in this state.

Access and Low_power states can also be entered and exited by the hardware low power signals (c_*)¹. In case of conflicting software and hardware low-power commands, the resulting operational state taken by the controller can be either one of the two conflicting requests.

Figure 1-3 illustrates the operational and transitional states.

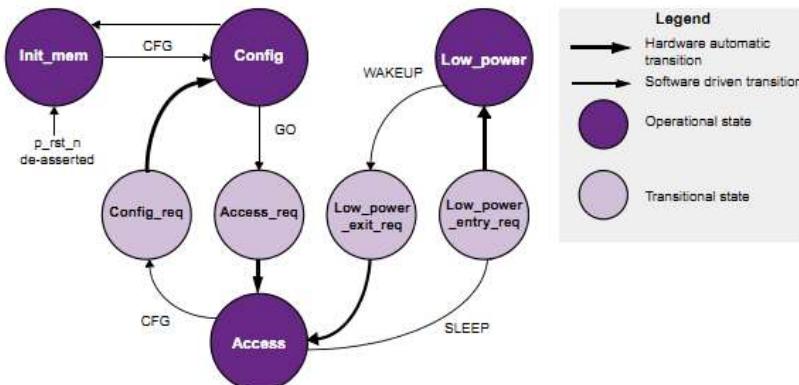


Fig. 6-3 Protocol controller architecture

The controller clock is the same clock driving the memory controller and will be the same frequency as the SDRAM clock (ck). The configuration clock can run at a frequency equal to or less than the controller clock. The configuration clock drives all non-DDR timing logic, such as configuration registers, PHY initialization, output impedance, and so on.

6.3.2 DDR PHY

DDR PHY provides turn key physical interface solutions for chip requiring access to DDR3 SDRAM device. It is optimized for low power and high speed (up to 800Mbps for DDR3) applications with robust timing and small silicon area. It supports DDR3 SDRAM components in the market. The PHY components contain DDR specialized functional and utility SSTL I/Os up to 800MHz, critical timing synchronization module (TSM) and a low power/jitter DLLs with programmable fine-grain control for any SDRAM interface.

DDR PHY uses a DFI digital interface to connect the memory controller. All interface timing is in 1X SDR clock domain. The controller to PHY interface is running at single data rate (SDR) therefore read/write bus is double width. DDR muxing is done in the PHY block together with all related per-byte lane timing adjustment. The interface is fairly generic and support high performance input and output data flow gearing toward 100Mbps to 800Mbps DDR3 SDRAM speed in wide range.

With configurable timing and driving strength and ODT parameters to interface to the wide variety of SDRAMs, the PHY is very flexible with advanced command capability to increase SDRAM operation efficiency.

6.4 Register Description

Slave address can be divided into different length for different usage, which is shown as follows.

6.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
DDR_PCTL_SCFG	0x0000	W	0x00000300	State Configuration Register
DDR_PCTL_SCTL	0x0004	W	0x00000000	Operational State Control Register
DDR_PCTL_STAT	0x0008	W	0x00000000	Operational State Status Register
DDR_PCTL_INTRSTAT	0x000c	W	0x00000000	Interrupt Status Register
DDR_PCTL_MCMD	0x0040	W	0x00100000	Memory Command Register
DDR_PCTL_POWCTL	0x0044	W	0x00000000	Power Up Control Register
DDR_PCTL_POWSTAT	0x0048	W	0x00000000	Power Up Status Register
DDR_PCTL_CMDTSTAT	0x004c	W	0x00000000	Command Timers Status Register
DDR_PCTL_CMDTSTATEN	0x0050	W	0x00000000	Command Timers Status Enable Register
DDR_PCTL_MRRCFG0	0x0060	W	0x00000000	Mode Register Read Configuration 0
DDR_PCTL_MRRSTAT0	0x0064	W	0x00000000	Mode Register Read Status 0 Register
DDR_PCTL_MRRSTAT1	0x0068	W	0x00000000	Mode Register Read Status 0 Register
DDR_PCTL_MCFG	0x0080	W	0x00040020	Memory Configuration Register
DDR_PCTL_PPCFG	0x0084	W	0x00000000	Partially Populated Memories Configuration Register
DDR_PCTL_MSTAT	0x0088	W	0x00000000	Memory Status Register
DDR_PCTL_MCFG1	0x0090	W	0x00000000	Memory Configuration 1 Register
DDR_PCTL_DTUPDES	0x0094	W	0x00000000	DTU Status Register
DDR_PCTL_DTUNA	0x0098	W	0x00000000	DTU Number of Addresses Created Register
DDR_PCTL_DTUNE	0x009c	W	0x00000000	DTU Number of Errors Register
DDR_PCTL_DTUPRD0	0x00a0	W	0x00000000	DTU Parallel Read 0 Register
DDR_PCTL_DTUPRD1	0x00a4	W	0x00000000	DTU Parallel Read 1 Register
DDR_PCTL_DTUPRD2	0x00a8	W	0x00000000	DTU Parallel Read 2 Register
DDR_PCTL_DTUPRD3	0x00ac	W	0x00000000	DTU Parallel Read 3 Register
DDR_PCTL_DTUAWDT	0x00b0	W	0x00000290	DTU Address Width Register
DDR_PCTL_TOGCNT1U	0x00c0	W	0x00000064	Toggle Counter 1us Register
DDR_PCTL_TINIT	0x00c4	W	0x000000c8	t_init Timing Register
DDR_PCTL_TRSTH	0x00c8	W	0x00000000	t_rsth Timing Register
DDR_PCTL_TOGCNT100N	0x00cc	W	0x00000001	Toggle Counter 100ns
DDR_PCTL_TREFI	0x00d0	W	0x00000001	t_refi Timing Register
DDR_PCTL_TMRD	0x00d4	W	0x00000001	t_mrd Timing Register
DDR_PCTL_TRFC	0x00d8	W	0x00000001	
DDR_PCTL_TRP	0x00dc	W	0x00010006	t_trp Timing Register
DDR_PCTL_TRTW	0x00e0	W	0x00000002	t_rtw Timing Register
DDR_PCTL_TAL	0x00e4	W	0x00000000	AL Register

Name	Offset	Size	Reset Value	Description
DDR_PCTL_TCL	0x00e8	W	0x00000004	CL Timing Register
DDR_PCTL_TCWL	0x00ec	W	0x00000003	CWL Timing Register
DDR_PCTL_TRAS	0x00f0	W	0x00000010	t_ras Timing Register
DDR_PCTL_TRC	0x00f4	W	0x00000016	t_rc Timing Register
DDR_PCTL_TRCD	0x00f8	W	0x00000006	t_rcd Timing Register
DDR_PCTL_TRRD	0x00fc	W	0x00000004	t_rrd Timing Register
DDR_PCTL_TRTP	0x0100	W	0x00000003	t_rtp Timing Register
DDR_PCTL_TWR	0x0104	W	0x00000006	t_wr Register
DDR_PCTL_TWTR	0x0108	W	0x00000004	t_wtr Timing Register
DDR_PCTL_TEXSR	0x010c	W	0x00000001	t_exsr Timing Register
DDR_PCTL_TXP	0x0110	W	0x00000001	t_xp Timing Register
DDR_PCTL_TXPDLL	0x0114	W	0x00000000	t_xpdll Timing Register
DDR_PCTL_TZQCS	0x0118	W	0x00000000	t_zqcs Timing Register
DDR_PCTL_TZQCSI	0x011c	W	0x00000000	t_zqcsi Timing Register
DDR_PCTL_TDQS	0x0120	W	0x00000001	t_dqs Timing Register
DDR_PCTL_TCKSRE	0x0124	W	0x00000000	t_cksre Timing Register
DDR_PCTL_TCKSRX	0x0128	W	0x00000000	t_cksrx Timing Register
DDR_PCTL_TCKE	0x012c	W	0x00000003	t_cke Timing Register
DDR_PCTL_TMOD	0x0130	W	0x00000000	t_mod Timing Register
DDR_PCTL_TRSTL	0x0134	W	0x00000000	Reset Low Timing Register
DDR_PCTL_TZQCL	0x0138	W	0x00000000	t_zqcl Timing Register
DDR_PCTL_TMRR	0x013c	W	0x00000002	t_mrr Timing Register
DDR_PCTL_TCKESR	0x0140	W	0x00000004	t_ckesr Timing Register
DDR_PCTL_TDPD	0x0144	W	0x00000000	t_dpd Timing Register
DDR_PCTL_DTUWACTL	0x0200	W	0x00000000	DTU Write Address Control
DDR_PCTL_DTURACTL	0x0204	W	0x00000000	DTU Read Address Control Register
DDR_PCTL_DTUCFG	0x0208	W	0x00000000	DTU Configuration Control Register
DDR_PCTL_DTUECTL	0x020c	W	0x00000000	DTU Execute Control Register
DDR_PCTL_DTUWD0	0x0210	W	0x00000000	DTU Write Data #0 Register
DDR_PCTL_DTUWD1	0x0214	W	0x00000000	DTU Write Data #1 Register
DDR_PCTL_DTUWD2	0x0218	W	0x00000000	DTU Write Data #2 Register
DDR_PCTL_DTUWD3	0x021c	W	0x00000000	DTU Write Data #3 Register
DDR_PCTL_DTUWDM	0x0220	W	0x00000000	DTU Write Data Mask Register
DDR_PCTL_DTURD0	0x0224	W	0x00000000	DTU Read Data #0 Register
DDR_PCTL_DTURD1	0x0228	W	0x00000000	DTU Read Data #1 Register
DDR_PCTL_DTURD2	0x022c	W	0x00000000	DTU Read Data #2 Register
DDR_PCTL_DTURD3	0x0230	W	0x00000000	DTU Read Data #3 Register
DDR_PCTL_DTULFSRW	0x0234	W	0x00000000	DTU LFSR Seed for Write Data Generation Register
DDR_PCTL_DTULFSRR	0x0238	W	0x00000000	DTU LFSR Seed for Read Data Generation Register
DDR_PCTL_DTUEAF	0x023c	W	0x00000000	DTU Error Address FIFO Register
DDR_PCTL_DFITCTRLDELAY	0x0240	W	0x00000002	DFI tctrl_delay Register
DDR_PCTL_DFIODTCFG	0x0244	W	0x00000000	DFI ODT Configuration

Name	Offset	Size	Reset Value	Description
DDR_PCTL_DFIODTCFG1	0x0248	W	0x06060000	DFI ODT Timing Configuration 1 (for Latency and Length)
DDR_PCTL_DFIODTRANKMAP	0x024c	W	0x00008421	DFI ODT Rank Mapping
DDR_PCTL_DFITPHYWRDATA	0x0250	W	0x00000001	DFI tphy_wrdata Register
DDR_PCTL_DFITPHYWRLAT	0x0254	W	0x00000001	DFI tphy_wrlat Register
DDR_PCTL_DFITRDDATAEN	0x0260	W	0x00000001	DFI trddata_en Register
DDR_PCTL_DFITPHYRDLAT	0x0264	W	0x0000000f	DFI tphy_rdlat Register
DDR_PCTL_DFITPHYUPDTYPE0	0x0270	W	0x00000010	DFI tphyupd_type0 Register
DDR_PCTL_DFITPHYUPDTYPE1	0x0274	W	0x00000010	DFI tphyupd_type1 Register
DDR_PCTL_DFITPHYUPDTYPE2	0x0278	W	0x00000010	DFI tphyupd_type2 Register
DDR_PCTL_DFITPHYUPDTYPE3	0x027c	W	0x00000010	DFI tphyupd_type3 Register
DDR_PCTL_DFITCTRLUPDMIN	0x0280	W	0x00000010	DFI tctrlupd_min Register
DDR_PCTL_DFITCTRLUPDMAX	0x0284	W	0x00000040	DFI tctrlupd_max Register
DDR_PCTL_DFITCTRLUPDDLY	0x0288	W	0x00000008	DFI tctrlupddly Register
DDR_PCTL_DFIUPDCFG	0x0290	W	0x00000003	DFI Update Configuration Register
DDR_PCTL_DFITREFMSKI	0x0294	W	0x00000000	DFI Masked Refresh Interval
DDR_PCTL_DFITCTRLUPDI	0x0298	W	0x00000000	DFI tctrlupd_interval Register
DDR_PCTL_DFITRCFG0	0x02ac	W	0x00000000	DFI Training Configuration 0 Register
DDR_PCTL_DFITRSTAT0	0x02b0	W	0x00000000	DFI Training Status 0 Register
DDR_PCTL_DFITRWRLVLEN	0x02b4	W	0x00000000	DFI Training dfi_wrlvl_en Register
DDR_PCTL_DFITRRDLVLEN	0x02b8	W	0x00000000	DFI Training dfi_rdlvl_en Register
DDR_PCTL_DFITRRDLVLGATEEN	0x02bc	W	0x00000000	DFI Training dfi_rdlvl_gate_en Register
DDR_PCTL_DFISTSTAT0	0x02c0	W	0x00000000	DFI Status Status 0 Register
DDR_PCTL_DFISTCFG0	0x02c4	W	0x00000000	DFI Status Configuration 0 Register
DDR_PCTL_DFISTCFG1	0x02c8	W	0x00000000	DFI Status Configuration 1 Register

Name	Offset	Size	Reset Value	Description
DDR_PCTL_DFIDRAMCLKEN	0x02d0	W	0x00000002	DFI tdram_clk_enable Register
DDR_PCTL_DFIDRAMCLKDIS	0x02d4	W	0x00000002	DFI tdram_clk_disable Register
DDR_PCTL_DFISTCFG2	0x02d8	W	0x00000000	DFI Status Configuration 2 Register
DDR_PCTL_DFISTPARCLR	0x02dc	W	0x00000000	DFI Status Parity Clear Register
DDR_PCTL_DFISTPARLOG	0x02e0	W	0x00000000	DFI Status Parity Log Register
DDR_PCTL_DFLPCFG0	0x02f0	W	0x00070000	DFI Low Power Configuration 0 Register
DDR_PCTL_DFITRWRLVLERSP0	0x0300	W	0x00000000	DFI Training dfi_wrlvl_resp Status 0 Register
DDR_PCTL_DFITRWRLVLERSP1	0x0304	W	0x00000000	DFI Training dfi_wrlvl_resp Status 1 Register
DDR_PCTL_DFITRWRLVLERSP2	0x0308	W	0x00000000	DFI Training dfi_wrlvl_resp Status 2 Register
DDR_PCTL_DFITRRDLVLERSP0	0x030c	W	0x00000000	DFI Training dfi_rdlvl_resp Status 0 Register
DDR_PCTL_DFITRRDLVLERSP1	0x0310	W	0x00000000	DFI Training dfi_rdlvl_resp Status 1 Register
DDR_PCTL_DFITRRDLVLERSP2	0x0314	W	0x00000000	DFI Training dfi_rdlvl_resp Status 2 Register
DDR_PCTL_DFITRWRLVLEDLAY0	0x0318	W	0x00000000	DFI Training dfi_wrlvl_delay Configuration 0 Register
DDR_PCTL_DFITRWRLVLEDLAY1	0x031c	W	0x00000000	DFI Training dfi_wrlvl_delay Configuration 1 Register
DDR_PCTL_DFITRWRLVLEDLAY2	0x0320	W	0x00000000	DFI Training dfi_wrlvl_delay Configuration 2 Register
DDR_PCTL_DFITRRDLVLEDLAY0	0x0324	W	0x00000000	DFI Training dfi_rdlvl_delay Configuration 0 Register
DDR_PCTL_DFITRRDLVLEDLAY1	0x0328	W	0x00000000	DFI Training dfi_rdlvl_delay Configuration 1 Register
DDR_PCTL_DFITRRDLVLEDLAY2	0x032c	W	0x00000000	DFI Training dfi_rdlvl_delay Configuration 2 Register
DDR_PCTL_DFITRRDLVLGATEDLAY0	0x0330	W	0x00000000	DFI Training dfi_rdlvl_gate_delay Configuration 0
DDR_PCTL_DFITRRDLVLGATEDLAY1	0x0334	W	0x00000000	DFI Training dfi_rdlvl_gate_delay Configuration 1
DDR_PCTL_DFITRRDLVLGATEDLAY2	0x0338	W	0x00000000	DFI Training dfi_rdlvl_gate_delay Configuration 2
DDR_PCTL_DFITRCMD	0x033c	W	0x00000000	DFI Training Command Register
DDR_PCTL_IPVR	0x03f8	W	0x00000000	IP Version Register
DDR_PCTL_IPTR	0x03fc	W	0x44574300	IP Type Register

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

6.4.2 Detail Register Description

DDR_PCTL_SCFG

Address: Operational Base + offset (0x0000)

State Configuration Register

Bit	Attr	Reset Value	Description
31:12	RO	0x0	reserved
11:8	RW	0x3	<p>bbflags_timing The n_bbflags is a NIF output vector which provides combined information about the status of each memory bank. The de-assertion is based on when precharge, activates, reads/writes are scheduled by the TCU block. It may be possible to de-assert n_bbflags earlier than calculated by the TCU block. Programming bbflags_timing is used to achieve this. The maximum recommended value is: TRP.t_rp. The programmed value is the maximum number of "early" cycles that n_bbflags maybe de-asserted. The actual achieved de-assertion depends on the traffic profile.</p>
7	RO	0x0	reserved
6	RW	0x0	<p>nfifo_nif1_dis For internal use only for NFIFO testing. 1'b0 = Only supported setting. 1'b1 = For internal use only.</p>
5:1	RO	0x0	reserved
0	RW	0x0	<p>hw_low_power_en Enables the hardware low-power interface. Allows the system to request via hardware (c_sysreq input) to enter the memories into Self-Refresh. The handshaking between the request and acknowledge hardware low power signals (c_sysreq and c_sysack, respectively) is always performed, but the ddr controller response depends on the value set on this register field and by the value driven on the c_active_in input pin. 1'b0 = Disabled. Requests are always denied and ddr controller is unaffected by c_sysreq 1'b1 = Enabled. Requests are accepted or denied, depending on the current operational state of ddr controller and on the value of c_active_in.</p>

DDR_PCTL_SCTL

Address: Operational Base + offset (0x0004)

Operational State Control Register

Bit	Attr	Reset Value	Description
31:3	RO	0x0	reserved

Bit	Attr	Reset Value	Description
2:0	RW	0x0	<p>state_cmd Issues an operational state transition request to the controller. 3'b000 = INIT (move to Init_mem from Config) 3'b001 = CFG (move to ConFig from Init_mem or Access) 3'b010 = GO (move to Access from Config) 3'b011 = SLEEP (move to Low_power from Access) 3'b100 = WAKEUP (move to Access from Low_power) Others = Reserved</p>

DDR_PCTL_STAT

Address: Operational Base + offset (0x0008)

Operational State Status Register

Bit	Attr	Reset Value	Description
31:7	RO	0x0	reserved
6:4	RO	0x0	<p>lp_trig Reports the status of what triggered an entry to Low_power state. Is only set if in Low_power state. The individual bits report the following: - lp_trig[2]: Software driven due to SCTL.state_cmd==SLEEP. - lp_trig[1]: Hardware driven due to Hardware Low Power Interface. - lp_trig[0]: Hardware driven due to Auto Self-Refresh (MCFG1.sr_idle>0). Note, if more than one trigger happens at the exact same time, more than one bit of lp_trig may be asserted high.</p>
3	RO	0x0	reserved
2:0	RO	0x0	<p>ctl_stat Returns the current operational state of the controller. 3'b000 = Init_mem 3'b001 = ConFig 3'b010 = Config_req 3'b011 = Access 3'b100 = Access_req 3'b101 = Low_power 3'b110 = Low_power_entry_req 3'b111 = Low_power_exit_req Others = Reserved</p>

DDR_PCTL_INTRSTAT

Address: Operational Base + offset (0x000c)

Interrupt Status Register

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved

Bit	Attr	Reset Value	Description
1	RO	0x0	parity_intr Indicates that a DFI parity error has been detected 1'b0 = No error 1'b1 = Parity error
0	RO	0x0	ecc_intr Indicates that an ECC error has been detected 1'b0 = No error 1'b1 = Parity error

DDR_PCTL_MCMD

Address: Operational Base + offset (0x0040)

Memory Command Register

Bit	Attr	Reset Value	Description
31	R/W SC	0x0	start_cmd Start command. When this bit is set to 1, the command operation defined in the cmd_opcode field is started. This bit is automatically cleared by the controller after the command is finished. The application can poll this bit to determine when controller is ready to accept another command. This bit cannot be cleared to 1'b0 by software.
30:28	RO	0x0	reserved
27:24	RW	0x0	cmd_add_del Set the additional delay associated with each command to 2^n internal timers clock cycles, where n is the bit field value. If n=0, the delay is 0. Max value is n=10.
23:20	RW	0x1	rank_sel Rank select for the command to be executed. 4'b0001 = Rank 0 4'b0010 = Rank 1 Others are reserved. Multiple 1'b1s in rank_sel mean multiple ranks are selected, which is useful broadcasting commands in parallel to multiple ranks during initialization and configuration of the memories. If MCMD.cmd_opcode=RSTL, all ranks should be selected as it cannot be performed to individual ranks
19:17	RW	0x0	bank_addr Mode Register address driven on the memory bank address bits, BA1, BA0, during a Mode Register Set operation, defined by cmd_opcode=MRS. For other values of cmd_opcode, this field is ignored. 3'b000 = MR0 3'b001 = MR1 3'b010 = MR2 3'b011 = MR3 Others = Reserved

Bit	Attr	Reset Value	Description
16:4	RW	0x0000	cmd_addr Mode Register value driven on the memory address bits, A12 to A0, during a Mode Register Set operation defined by cmd_opcode = MRS. For other values of cmd_opcode this field is ignored. Refer to the memory specification for the correct settings of the various bits of this field during a MRS operation.
3:0	RW	0x0	cmd_opcode Command to be issued to the memory. 4'b000 = Deselect. This is only used for timing purposes, no actual direct Deselect command is passed to the memories. 4'b0001 = Precharge All (PREA) 4'b0010 = Refresh (REF) 4'b0011 = Mode Register Set (MRS), MRS otherwise 4'b0100 = ZQ Calibration Short (ZQCS, only applies to DDR3) 4'b0101 = ZQ Calibration Long (ZQCL, only applies to DDR3) 4'b0110 = Software Driven Reset (RSTL, only applies to DDR3) 4'b0111 = Reserved 4'b1000 - Mode Register Read (MRR), is SRR in mDRR and is MPR in DDR3 Others - Reserved

DDR_PCTL_POWCTL

Address: Operational Base + offset (0x0044)

Power Up Control Register

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	R/W SC	0x0	power_up_start Start the memory power up sequence. When this bit is set to 1'b1, controller starts the CKE and RESET# power up sequence to the memories. This bit is automatically cleared by controller after the sequence is completed. This bit cannot be cleared to 1'b0 by software.

DDR_PCTL_POWSTAT

Address: Operational Base + offset (0x0048)

Power Up Status Register

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RO	0x0	power_up_done Returns the status of the memory power-up sequence. 1'b0 = Power-up sequence has not been performed. 1'b1 = Power-up sequence has been performed.

DDR_PCTL_CMDTSTAT

Address: Operational Base + offset (0x004c)

Command Timers Status Register

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RO	0x0	cmd_tstat Returns the status of the timers for memory commands. This ANDs all the command timers together. 1'b0 = One or more command timers has not expired. 1'b1 = All command timers have expired.

DDR_PCTL_CMDTSTATEN

Address: Operational Base + offset (0x0050)

Command Timers Status Enable Register

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	cmd_tstat_en Enables the generation of the status of the timers for memory commands. Is enabled before CMDTSTAT register is read. 1'b0 - Disabled 1'b1 - Enabled

DDR_PCTL_MRRCFG0

Address: Operational Base + offset (0x0060)

Mode Register Read Configuration 0

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3:0	RW	0x0	mrr_byte_sel Selects which byte's data to store when performing an MRR command via MCMD. LegalValues: 0 .. 8

DDR_PCTL_MRRSTAT0

Address: Operational Base + offset (0x0064)

Mode Register Read Status 0 Register

Bit	Attr	Reset Value	Description
31:24	RO	0x00	mrrstat_beat3 MRR/MPR read data beat 3
23:16	RO	0x00	mrrstat_beat2 MRR/MPR read data beat 2
15:8	RO	0x00	mrrstat_beat1 MRR/MPR read data beat 1
7:0	RO	0x00	mrrstat_beat0 MRR/MPR read data beat 0

DDR_PCTL_MRRSTAT1

Address: Operational Base + offset (0x0068)

Mode Register Read Status 0 Register

Bit	Attr	Reset Value	Description
31:24	RO	0x00	mrrstat_beat7 MRR/MPR read data beat 7
23:16	RO	0x00	mrrstat_beat6 MRR/MPR read data beat 6
15:8	RO	0x00	mrrstat_beat5 MRR/MPR read data beat 5
7:0	RO	0x00	mrrstat_beat4 MRR/MPR read data beat 4

DDR_PCTL_MCFG1

Address: Operational Base + offset (0x007c)

Memory Configuration 1 Register

Bit	Attr	Reset Value	Description
31	RW	0x0	hw_exit_idle_en When this bit is programmed to 1'b1 the c_active_in pin can be used to exit from the automatic clock stop, power down or self-refresh modes.
30:24	RO	0x0	reserved
23:16	RW	0x00	hw_idle Hardware idle period. The c_active output is driven high if the NIF is idle in Access state for hw_idle * 32 * n_clk cycles. The hardware idle function is disabled when hw_idle=0.
15:8	RO	0x0	reserved
7:0	RW	0x00	sr_idle Self-Refresh idle period. Memories are placed into Self-Refresh mode if the NIF is idle in Access state for sr_idle * 32 * n_clk cycles. The automatic self-refresh function is disabled when sr_idle=0.

DDR_PCTL_MCFG

Address: Operational Base + offset (0x0080)

Memory Configuration Register

Bit	Attr	Reset Value	Description
31:20	RW	0x0	reserved
19:18	RW	0x1	tfaw_cfg Field0000 Abstract Sets tFAW to be 4, 5 or 6 times tRRD. 2'b00 = set tFAW=4*tRRD 2'b01 = set tFAW=5*tRRD 2'b10 = set tFAW=6*tRRD
17	RW	0x0	pd_exit_mode Selects the mode for Power Down Exit. For DDR3, the power down exit mode setting in controller must be consistent with the value programmed into the power down exit mode bit of MR0. 1'b0 = slow exit 1'b1 = fast exit

Bit	Attr	Reset Value	Description
16	RW	0x0	<p>pd_type Sets the Power down type. 1'b0 = Precharge Power Down 1'b1 = Active Power Down</p>
15:8	RW	0x00	<p>pd_idle Power-down idle period in n_clk cycles. Memories are placed into power-down mode if the NIF is idle for pd_idle n_clk cycles. The automatic power down function is disabled when pd_idle=0.</p>
7:5	RO	0x0	reserved
4	RW	0x0	<p>stagger_cs For multi-rank commands from the DCU, stagger the assertion of CS_N to odd and even ranks by one n_clk cycle. This is useful when using RDIMMs, when multi-rank commands may be interpreted as writes to control words in the register chip. 1'b0 = Do not stagger CS_N 1'b1 = Stagger CS_N</p>
3	RW	0x0	<p>two_t_en Enables 2T timing for memory commands. 1'b0= Disabled 1'b1 = Enabled</p>
2:1	RO	0x0	reserved
0	RW	0x0	<p>mem_bl DDR Burst Length. The BL setting in DDR3 must be consistent with the value programmed into the BL field of MR0. 1'b0 = BL4, Burst length of 4 1'b1 = BL8, Burst length of 8 (MR0.BL=2'b00 for DDR3)</p>

DDR_PCTL_PPCFG

Address: Operational Base + offset (0x0084)

Partially Populated Memories Configuration Register

Bit	Attr	Reset Value	Description
31:9	RO	0x0	reserved

Bit	Attr	Reset Value	Description
8:1	RW	0x00	<p>rpmem_dis Reduced Population Disable bits. Setting these bits disables the corresponding NIF/DDR data lanes from writing or reading data. Lane 0 is always present, hence only 8 bits are required for the remaining lanes including the ECC lane.</p> <p>In 1:2 mode bit 0 of rpmem_dis covers n_wdata/n_rdata/m_ctl_d/m_phy_q[63:32], bit 1 [95:64] etc.</p> <p>In 1:1 mode bit 0 of rpmem_dis covers n_wdata/n_rdata/m_ctl_d/m_phy_q[31:16], bit 2 [47:32] etc.</p> <p>There are no restrictions on which byte lanes can be disabled, other than byte lane 0 is required. Gaps between enabled byte lanes are allowed. For each bit:</p> <p>1'b0 = lane exists 1'b1 = lane is disabled</p>
0	RW	0x0	<p>ppmem_en Partially Population Enable bit. Setting this bit enables the partial population of external memories where the entire application bus is routed to a reduced size memory system. The lower half of the SDRAM data bus, bit 0 up to bit 15, is the active portion when Partially Populated memories are enabled.</p> <p>1'b0 = Disabled 1'b1 = Enabled</p>

DDR_PCTL_MSTAT

Address: Operational Base + offset (0x0088)

Memory Status Register

Bit	Attr	Reset Value	Description
31:3	RO	0x0	reserved
2	RO	0x0	<p>self_refresh Indicates if controller, through auto self-refresh, has placed the memories in Self-Refresh.</p> <p>1'b0 = Memory is not in Self-Refresh 1'b1 = Memory is in Self-Refresh</p>
1	RO	0x0	<p>clock_stop Indicates if controller has placed the memories in Clock Stop.</p> <p>1'b0 = Memory is not in Clock Stop 1'b1 = Memory is in Clock Stop</p>
0	RO	0x0	<p>power_down Indicates if controller has placed the memories in Power Down.</p> <p>1'b0 = Memory is not in Power Down 1'b1 = Memory is in Power-Down</p>

DDR_PCTL_DTUPDES

Address: Operational Base + offset (0x0094)

DTU Status Register

Bit	Attr	Reset Value	Description
31:14	RO	0x0	reserved
13	RO	0x0	dtu_rd_missing Indicates if one or more read beats of data did not return from memory.
12:9	RO	0x0	dtu_eaffl Indicates the number of entries in the FIFO that is holding the log of error addresses for data comparison
8	RO	0x0	dtu_random_error Indicates that the random data generated had some failures when written and read to the memories
7	RO	0x0	dtu_err_b7 Detected at least 1 bit error for bit 7 in the programmable data buffers
6	RO	0x0	dtu_err_b6 Detected at least 1 bit error for bit 6 in the programmable data buffers
5	RO	0x0	dtu_err_b5 Detected at least 1 bit error for bit 5 in the programmable data buffers
4	RO	0x0	dtu_err_b4 Detected at least 1 bit error for bit 4 in the programmable data buffers
3	RO	0x0	dtu_err_b3 Detected at least 1 bit error for bit 3 in the programmable data buffers
2	RO	0x0	dtu_err_b2 Detected at least 1 bit error for bit 2 in the programmable data buffers
1	RO	0x0	dtu_err_b1 Detected at least 1 bit error for bit 1 in the programmable data buffers
0	RO	0x0	dtu_err_b0 Detected at least 1 bit error for bit 0 in the programmable data buffers

DDR_PCTL_DTUNA

Address: Operational Base + offset (0x0098)

DTU Number of Addresses Created Register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	dtu_num_address Indicates the number of addresses that were created on the NIF interface during random data generation.

DDR_PCTL_DTUNE

Address: Operational Base + offset (0x009c)

DTU Number of Errors Register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	dtu_num_errors Indicates the number of errors that were detected on the readback of the NIF data during random data generation.

DDR_PCTL_DTUPRD0

Address: Operational Base + offset (0x00a0)

DTU Parallel Read 0 Register

Bit	Attr	Reset Value	Description
31:16	RO	0x0000	dtu_allbits_1 Allows all the bit ones from each of the 16 received read bytes to be read in parallel. Used as part of read data eye training where a transition is required to be monitored to train the eye.
15:0	RO	0x0000	dtu_allbits_0 Allows all the bit zeros from each of the 16 received read bytes to be read in parallel. Used as part of read data eye training where a transition is required to be monitored to train the eye.

DDR_PCTL_DTUPRD1

Address: Operational Base + offset (0x00a4)

DTU Parallel Read 1 Register

Bit	Attr	Reset Value	Description
31:16	RO	0x0000	dtu_allbits_3 Allows all the bit threes from each of the 16 received read bytes to be read in parallel. Used as part of read data eye training where a transition is required to be monitored to train the eye.
15:0	RO	0x0000	dtu_allbits_2 Allows all the bit twos from each of the 16 received read bytes to be read in parallel. Used as part of read data eye training where a transition is required to be monitored to train the eye.

DDR_PCTL_DTUPRD2

Address: Operational Base + offset (0x00a8)

DTU Parallel Read 2 Register

Bit	Attr	Reset Value	Description
31:16	RO	0x0000	dtu_allbits_5 Allows all the bit fives from each of the 16 received read bytes to be read in parallel. Used as part of read data eye training where a transition is required to be monitored to train the eye.
15:0	RO	0x0000	dtu_allbits_4 Allows all the bit fours from each of the 16 received read bytes to be read in parallel. Used as part of read data eye training where a transition is required to be monitored to train the eye.

DDR_PCTL_DTUPRD3

Address: Operational Base + offset (0x00ac)

DTU Parallel Read 3 Register

Bit	Attr	Reset Value	Description
31:16	RO	0x0000	dtu_allbits_7 Allows all the bit sevens from each of the 16 received read bytes to be read in parallel. Used as part of read data eye training where a transition is required to be monitored to train the eye.
15:0	RO	0x0000	dtu_allbits_6 Allows all the bit sixes from each of the 16 received read bytes to be read in parallel. Used as part of read data eye training where a transition is required to be monitored to train the eye.

DDR_PCTL_DTUAWDT

Address: Operational Base + offset (0x00b0)

DTU Address Width Register

Bit	Attr	Reset Value	Description
31:11	RO	0x0	reserved
10:9	RW	0x1	number_ranks Number of supported memory ranks. 2'b00 = 1 rank 2'b01 = 2 ranks
8	RO	0x0	reserved
7:6	RW	0x2	row_addr_width Width of the memory row address bits. 2'b00 = 13 bits wide 2'b01 = 14 bits wide 2'b10 = 15 bits wide 2'b11 = 16 bits wide
5	RO	0x0	reserved
4:3	RW	0x2	bank_addr_width Field0000 Abstract Width of the memory bank address bits. 2'b00 = 2 bits wide (4 banks) 2'b01 = 3 bits wide (8 banks) Others = Reserved
2	RO	0x0	reserved
1:0	RW	0x0	column_addr_width Field0000 Abstract Width of the memory column address bits. 2'b00 = 7 bits wide 2'b01 = 8 bits wide 2'b10 = 9 bits wide 2'b11 = 10 bits wide

DDR_PCTL_TOGCNT1U

Address: Operational Base + offset (0x00c0)

Toggle Counter 1us Register

Bit	Attr	Reset Value	Description
31:10	RO	0x0	reserved
9:0	RW	0x064	toggle_counter_1u The number of internal timers clock cycles

DDR_PCTL_TINIT

Address: Operational Base + offset (0x00c4)

t_init Timing Register

Bit	Attr	Reset Value	Description
31:9	RO	0x0	reserved
8:0	RW	0x0c8	t_init Defines the time period (in us) to hold dfi_cke and dfi_reset_n stable during the memory power up sequence. The value programmed must correspond to at least 200us. The actual time period defined is TINIT * TOGCNT1U * internal timers clock .period

DDR_PCTL_TRSTH

Address: Operational Base + offset (0x00c8)

t_rsth Timing Register

Bit	Attr	Reset Value	Description
31:10	RO	0x0	reserved
9:0	RW	0x000	t_rsth Defines the time period (in us) to hold the dfi_reset_n signal high after it is de-asserted during the DDR3 Power Up/Reset sequence. The value programmed for DDR3 must correspond to minimum 500us of delay. The actual time period defined is TRSTH * TOGCNT1U * internal timers clock period.

DDR_PCTL_TOGCNT100N

Address: Operational Base + offset (0x00cc)

Toggle Counter 100ns

Bit	Attr	Reset Value	Description
31:7	RO	0x0	reserved
6:0	RW	0x01	toggle_counter_100n The number of internal timers clock cycles.

DDR_PCTL_TREFI

Address: Operational Base + offset (0x00d0)

t_refi Timing Register

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RW	0x01	t_refi Defines the time period (in 100ns units) of the Refresh interval. The actual time period defined is TREFI * TOGCNT100N * internal timers clock period.

DDR_PCTL_TMRD

Address: Operational Base + offset (0x00d4)

t_mrd Timing Register

Bit	Attr	Reset Value	Description
31:3	RO	0x0	reserved
2:0	RW	0x1	t_mrd Mode Register Set command cycle time in memory clock cycles. DDR3: Time from MRS to MRS command. DDR3 Legal Values: 2..4

DDR_PCTL_TRFC

Address: Operational Base + offset (0x00d8)

Bit	Attr	Reset Value	Description
31:9	RO	0x0	reserved
8:0	RW	0x001	t_rfc Refresh to Active/Refresh command time in memory clock cycles. DDR3 Legal Values: 36.. 374

DDR_PCTL_TRP

Address: Operational Base + offset (0x00dc)

t_trp Timing Register

Bit	Attr	Reset Value	Description
31:18	RO	0x0	reserved
17:16	RW	0x1	prea_extra Additional cycles required for a Precharge All (PREA) command - in addition to t_rp. In terms of memory clock cycles DDR3 Value: 0
15:4	RO	0x0	reserved
3:0	RW	0x6	t_rp Precharge period in memory clock cycles. DDR3 Legal Values: 5..14

DDR_PCTL_TRTW

Address: Operational Base + offset (0x00e0)

t_rtw Timing Register

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3:0	RW	0x2	t_rtw Read to Write turnaround time in memory clock cycles. DDR3 Legal Values: 2..10

DDR_PCTL_TAL

Address: Operational Base + offset (0x00e4)

AL Register

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3:0	RW	0x0	t_al Additive Latency in memory clock cycles. For DDR3 this must be 0, CL-1, CL-2 depending weather the AL value in MR1 is 0, 1, or 2 respectively. CL is the CAS latency programmed into MR0. DDR3 Legal Values: 0, CL-1, CL-2 (depending on AL=0,1,2 in MR1)

DDR_PCTL_TCL

Address: Operational Base + offset (0x00e8)

CL Timing Register

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3:0	RW	0x4	t_cl CAS Latency in memory clock cycles. The controller setting must match the value programmed into the CL field of MR0. DDR3 Legal Value: CL

DDR_PCTL_TCWL

Address: Operational Base + offset (0x00ec)

CWL Timing Register

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3:0	RW	0x3	t_cwl CAS Write Latency in memory clock cycles. For DDR3, the setting must match the value programmed in the memory CWL field of MR2. DDR3 Legal Value: CWL

DDR_PCTL_TRAS

Address: Operational Base + offset (0x00f0)

t_ras Timing Register

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5:0	RW	0x10	t_ras Activate to Precharge command time in memory clock cycles. DDR3 Legal Values: 15..38

DDR_PCTL_TRC

Address: Operational Base + offset (0x00f4)

t_rc Timing Register

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved

Bit	Attr	Reset Value	Description
5:0	RW	0x16	t_rc Row Cycle time in memory clock cycles. Specifies the minimum Activate to Activate distance for accesses to same bank. DDR3 Legal Values: 20..52

DDR_PCTL_TRCD

Address: Operational Base + offset (0x00f8)

t_rcd Timing Register

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3:0	RW	0x6	t_rcd Row to Column delay in memory clock cycles. Specifies the minimum Activate to Column distance. DDR3 Legal Values: 5..14

DDR_PCTL_TRRD

Address: Operational Base + offset (0x00fc)

t_rrd Timing Register

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3:0	RW	0x4	t_rrd Row-to-Row delay in memory clock cycles. Specifies the minimum Activate-to-Activate distance for consecutive accesses to different banks in the same rank. DDR3 Legal Values: 4..8

DDR_PCTL_TRTP

Address: Operational Base + offset (0x0100)

t_rtp Timing Register

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3:0	RW	0x3	t_rtp Read to Precharge time in memory clock cycles. Specifies the minimum distance Read to Precharge for consecutive accesses to same bank. DDR3 Legal Values: 3..8

DDR_PCTL_TWR

Address: Operational Base + offset (0x0104)

t_wr Register

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved

Bit	Attr	Reset Value	Description
4:0	RW	0x06	t_wr Write recovery time in memory clock cycles. When using close page the controller setting must be consistent with the WR field setting of MRO. DDR3 Legal Values: 6..16

DDR_PCTL_TWTR

Address: Operational Base + offset (0x0108)

t_wtr Timing Register

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3:0	RW	0x4	t_wtr Write to Read turnaround time, in memory clock cycles. DDR3 Legal Values: 3..8

DDR_PCTL_TEXSR

Address: Operational Base + offset (0x010c)

t_exsr Timing Register

Bit	Attr	Reset Value	Description
31:10	RO	0x0	reserved
9:0	RW	0x001	t_exsr Exit Self-Refresh to first valid command delay, in memory clock cycles. For DDR3, this should be programmed to match tXSDLL (SRE to a command requiring DLL locked) as defined by the memory device specification. DDR3 Typical Value: 512

DDR_PCTL_TXP

Address: Operational Base + offset (0x0110)

t_xp Timing Register

Bit	Attr	Reset Value	Description
31:3	RO	0x0	reserved
2:0	RW	0x1	t_xp Exit Power Down to first valid command delay when DLL is on (fast exit), measured in memory clock cycles. Legal Values: 1..7

DDR_PCTL_TXPDLL

Address: Operational Base + offset (0x0114)

t_xpdll Timing Register

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved

Bit	Attr	Reset Value	Description
5:0	RW	0x00	t_xpdll Exit Power Down to first valid command delay when DLL is off (slow exit), measured in memory clock cycles. DDR3 Legal Values: 3..63

DDR_PCTL_TZQCS

Address: Operational Base + offset (0x0118)

t_zqcs Timing Register

Bit	Attr	Reset Value	Description
31:7	RO	0x0	reserved
6:0	RW	0x00	t_zqcs SDRAM ZQ Calibration Short period, in memory clock cycles. Should be programmed to match the tZQCS timing value as defined in the memory specification. DDR3 Typical Value: 64

DDR_PCTL_TZQCSI

Address: Operational Base + offset (0x011c)

t_zqcsi Timing Register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	t_zqcsi SDRAM ZQCS interval, measured in Refresh interval units. The total time period defined is TZQCSI * TREFI * TOGCNT100N * internal timers clock period. Programming a value of 0 in t_zqcsi disables the auto-ZQCS functionality in controller. DDR3 Legal Values: 0..4294967295

DDR_PCTL_TDQS

Address: Operational Base + offset (0x0120)

t_dqs Timing Register

Bit	Attr	Reset Value	Description
31:3	RO	0x0	reserved
2:0	RW	0x1	t_dqs Additional data turnaround time in memory clock cycles for accesses to different ranks. Used to increase the distance between column commands to different ranks, allowing more tolerance as the driver source changes on the bidirectional DQS and/or DQ signals. DDR3 Legal Values: 1..7

DDR_PCTL_TCKSRE

Address: Operational Base + offset (0x0124)

t_cksre Timing Register

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved

Bit	Attr	Reset Value	Description
4:0	RW	0x00	t_cksrc In DDR3, this is the time after Self-Refresh Entry that CKE is held high before going low. In memory clock cycles. Specifies the clock disable delay after SRE. This should be programmed to match the greatest value between 10ns and 5 memory clock periods. DDR3 Legal Values: 5..15

DDR_PCTL_TCKSRX

Address: Operational Base + offset (0x0128)

t_cksrc Timing Register

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4:0	RW	0x00	t_cksrc In DDR3, this is the time (before Self-Refresh Exit) that CKE is maintained high before issuing SRX. In memory clock cycles. Specifies the clock stable time before SRX. This should be programmed to match the greatest value between 10ns and 5 memory clock periods. DDR3 Legal Values: 5..15

DDR_PCTL_TCKE

Address: Operational Base + offset (0x012c)

t_cke Timing Register

Bit	Attr	Reset Value	Description
31:3	RO	0x0	reserved
2:0	RW	0x3	t_cke CKE minimum pulse width in memory clock cycles. DDR3 Legal Values: 3..6

DDR_PCTL_TMOD

Address: Operational Base + offset (0x0130)

t_mod Timing Register

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4:0	RW	0x00	t_mod In DDR3 mode, this is the time from MRS to any valid non-MRS command (except DESELECT or NOP) in memory clock cycles. DDR3 Legal Values: 0..31

DDR_PCTL_TRSTL

Address: Operational Base + offset (0x0134)

Reset Low Timing Register

Bit	Attr	Reset Value	Description
31:7	RO	0x0	reserved

Bit	Attr	Reset Value	Description
6:0	RW	0x00	t_rstl Memory Reset Low time, in memory clock cycles. Defines the time period to hold dfi_reset_n signal low during a software driven DDR3 Reset Operation. The value programmed must correspond to at least 100ns of delay. DDR3 Legal Values: 1..127

DDR_PCTL_TZQCL

Address: Operational Base + offset (0x0138)

t_zqcl Timing Register

Bit	Attr	Reset Value	Description
31:10	RO	0x0	reserved
9:0	RW	0x000	t_zqcl SDRAM ZQ Calibration Long period in memory clock cycles. If DDR3, should be programmed to match the memory tZQinit timing value for the first ZQCL command during memory initialization; should be programmed to match tZQoper timing value after reset and initialization. DDR3 Legal Values: 0..1023

DDR_PCTL_TMRR

Address: Operational Base + offset (0x013c)

t_mrr Timing Register

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RW	0x02	t_mrr Time for a Mode Register Read (MRR command from MCMD).

DDR_PCTL_TCKESR

Address: Operational Base + offset (0x0140)

t_ckesr Timing Register

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3:0	RW	0x4	t_ckesr Minimum CKE low width for Self-Refresh entry to exit timing in memory clock cycles. Recommended settings: For DDR3 : t_ckesr = t_cke + 1 DDR3 Legal Values: 4..7

DDR_PCTL_TDPPD

Address: Operational Base + offset (0x0144)

t_dpd Timing Register

Bit	Attr	Reset Value	Description
31:10	RO	0x0	reserved

Bit	Attr	Reset Value	Description
9:0	RW	0x000	t_dpd Minimum Deep Power Down time. Is in terms of us. When a MCMD.DPDE command occurs, TDPD time is waited before MCMD.start_cmd can be cleared. MCMD_cmd_add_del (if any) does not start until TDPD has completed. This ensures TDPD requirement for the memory is not violated. The actual time period defined is TDPD* TOGCNT1U * internal timers clock period. DDR3 Legal Values: 0

DDR_PCTL_DTUWACTL

Address: Operational Base + offset (0x0200)

DTU Write Address Control

Bit	Attr	Reset Value	Description
31:30	RW	0x0	dtu_wr_rank Write rank to where data is to be targeted
29	RO	0x0	reserved
28:13	RW	0x0000	dtu_wr_row Write row to where data is to be targeted
12:10	RW	0x0	dtu_wr_bank Write bank to where data is to be targeted
9:0	RW	0x000	dtu_wr_col FWrite column to where data is to be targeted

DDR_PCTL_DTURACTL

Address: Operational Base + offset (0x0204)

DTU Read Address Control Register

Bit	Attr	Reset Value	Description
31:30	RW	0x0	dtu_rd_rank Read rank from where data comes
29	RO	0x0	reserved
28:13	RW	0x0000	dtu_rd_row Read row from where data comes
12:10	RW	0x0	dtu_rd_bank Read bank from where data comes
9:0	RW	0x000	dtu_rd_col Read column from where data comes

DDR_PCTL_DTUCFG

Address: Operational Base + offset (0x0208)

DTU Configuration Control Register

Bit	Attr	Reset Value	Description
31:23	RO	0x0	reserved
22:16	RW	0x00	dtu_row_increments Number of times to increment the row address when generating random data, up to a maximum of 127 times.

Bit	Attr	Reset Value	Description
15	RW	0x0	dtu_wr_multi_rd When set puts the DTU into write once multiple reads mode.
14	RW	0x0	dtu_data_mask_en Controls whether random generated data masks are transmitted. Unless enabled all data bytes are written to memory and expected to be read from memory.
13:10	RW	0x0	dtu_target_lane Selects one of the byte lanes for data comparison into the programmable read data buffer.
9	RW	0x0	dtu_generate_random Generate transfers using random data, otherwise generate transfers from the programmable write data buffers.
8	RW	0x0	dtu_incr_banks When the column address rolls over increment the bank address until we reach and conclude bank 7.
7	RW	0x0	dtu_incr_cols Increment the column address until we saturate. Return to zero if DTUCFG.dtu_incr_banks is set to 1 and we are not at bank 7.
6:1	RW	0x00	dtu_nalen Length of the NIF transfer sequence that is passed through the controller for each created address.
0	RW	0x0	dtu_enable When set, allows the DTU module to take ownership of the NIF interface: 1: DTU enabled 0: DTU disabled

DDR_PCTL_DTUectl

Address: Operational Base + offset (0x020c)

DTU Execute Control Register

Bit	Attr	Reset Value	Description
31:3	RO	0x0	reserved
2	R/W SC	0x0	wr_multi_rd_RST When set, resets the DTU in write once multiple reads mode, to allow a new write to be performed. This bit automatically clears.
1	R/W SC	0x0	run_error_reports When set, initiates the calculation of the error status bits. This bit automatically clears when the re-calculation is done. This is only used in debug mode to verify the comparison logic.
0	R/W SC	0x0	run_dtu When set, initiates the running of the DTU read and write transfer. This bit automatically clears when the transfers are completed

DDR_PCTL_DTUwdo

Address: Operational Base + offset (0x0210)

DTU Write Data #0 Register

Bit	Attr	Reset Value	Description
31:24	RW	0x00	dtu_wr_byte3 Write data byte
23:16	RW	0x00	dtu_wr_byte2 Write data byte
15:8	RW	0x00	dtu_wr_byte1 Write data byte
7:0	RW	0x00	dtu_wr_byte0 Write data byte

DDR_PCTL_DTUWD1

Address: Operational Base + offset (0x0214)

DTU Write Data #1 Register

Bit	Attr	Reset Value	Description
31:24	RW	0x00	dtu_wr_byte7 Write data byte
23:16	RW	0x00	dtu_wr_byte6 Write data byte
15:8	RW	0x00	dtu_wr_byte5 Write data byte
7:0	RW	0x00	dtu_wr_byte4 Write data byte

DDR_PCTL_DTUWD2

Address: Operational Base + offset (0x0218)

DTU Write Data #2 Register

Bit	Attr	Reset Value	Description
31:24	RW	0x00	dtu_wr_byte11 Write data byte
23:16	RW	0x00	dtu_wr_byte10 Write data byte
15:8	RW	0x00	dtu_wr_byte9 Write data byte
7:0	RW	0x00	dtu_wr_byte8 Write data byte

DDR_PCTL_DTUWD3

Address: Operational Base + offset (0x021c)

DTU Write Data #3 Register

Bit	Attr	Reset Value	Description
31:24	RW	0x00	dtu_wr_byte15 Write data byte
23:16	RW	0x00	dtu_wr_byte14 Write data byte

Bit	Attr	Reset Value	Description
15:8	RW	0x00	dtu_wr_byte13 Write data byte
7:0	RW	0x00	dtu_wr_byte12 Write data byte

DDR_PCTL_DTUWDM

Address: Operational Base + offset (0x0220)

DTU Write Data Mask Register

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	dm_wr_byte0 Write data mask bit, one bit for each byte. Each bit should be 0 for a byte lane that contains valid write data.

DDR_PCTL_DTURDO

Address: Operational Base + offset (0x0224)

DTU Read Data #0 Register

Bit	Attr	Reset Value	Description
31:24	RO	0x00	dtu_rd_byte3 Read byte
23:16	RO	0x00	dtu_rd_byte2 Read byte
15:8	RO	0x00	dtu_rd_byte1 Read byte
7:0	RO	0x00	dtu_rd_byte0 Read byte

DDR_PCTL_DTURD1

Address: Operational Base + offset (0x0228)

DTU Read Data #1 Register

Bit	Attr	Reset Value	Description
31:24	RO	0x00	dtu_rd_byte7 Read byte
23:16	RO	0x00	dtu_rd_byte6 Read byte
15:8	RO	0x00	dtu_rd_byte5 Read byte
7:0	RO	0x00	dtu_rd_byte4 Read byte

DDR_PCTL_DTURD2

Address: Operational Base + offset (0x022c)

DTU Read Data #2 Register

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:24	RO	0x00	dtu_rd_byte11 Read byte
23:16	RO	0x00	dtu_rd_byte10 Read byte
15:8	RO	0x00	dtu_rd_byte9 Read byte
7:0	RO	0x00	dtu_rd_byte8 Read byte

DDR_PCTL_DTURD3

Address: Operational Base + offset (0x0230)

DTU Read Data #3 Register

Bit	Attr	Reset Value	Description
31:24	RO	0x00	dtu_rd_byte15 Read byte
23:16	RO	0x00	dtu_rd_byte14 Read byte
15:8	RO	0x00	dtu_rd_byte13 Read byte
7:0	RO	0x00	dtu_rd_byte12 Read byte

DDR_PCTL_DTULFSRWD

Address: Operational Base + offset (0x0234)

DTU LFSR Seed for Write Data Generation Register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	dtu_lfsr_wseed This is the initial seed for the random write data generation LFSR (linear feedback shift register), shared with the write mask generation.

DDR_PCTL_DTULFSRRD

Address: Operational Base + offset (0x0238)

DTU LFSR Seed for Read Data Generation Register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	dtu_lfsr_rseed This is the initial seed for the random read data generation LFSR (linear feedback shift register), this is shared with the read mask generation. The read data mask is reconstructed the same as the write data mask was created, allowing the "on the fly comparison" ignore bytes which were not written.

DDR_PCTL_DTUEAF

Address: Operational Base + offset (0x023c)

DTU Error Address FIFO Register

Bit	Attr	Reset Value	Description
31:30	RO	0x0	ea_rank Indicates the rank that the error occurred in during random data generation. There could be a number of entries in this FIFO. If FIFO is empty one reads zeroes.
29	RO	0x0	reserved
28:13	RO	0x0000	ea_row Indicates the row that the error occurred in during random data generation. There could be a number of entries in this FIFO. If FIFO is empty one reads zeroes.
12:10	RO	0x0	ea_bank Indicates the bank that the error occurred in during random data generation. There could be a number of entries in this FIFO. If FIFO is empty one reads zeroes
9:0	RO	0x000	ea_column Indicates the column address that the error occurred in during random data generation. There could be a number of entries in this FIFO. If FIFO is empty one reads zeroes.

DDR_PCTL_DFITCTRLDELAY

Address: Operational Base + offset (0x0240)

DFI tctrl_delay Register

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3:0	RW	0x2	tctrl_delay Specifies the number of DFI clock cycles after an assertion or deassertion of the DFI control signals that the control signals at the PHY-DRAM interface reflect the assertion or de-assertion. If the DFI clock and the memory clock are not phase-aligned, this timing parameter should be rounded up to the next integer value.

DDR_PCTL_DFIODTCFG

Address: Operational Base + offset (0x0244)

DFI ODT Configuration

Bit	Attr	Reset Value	Description
31:13	RO	0x0	reserved
12	RW	0x0	rank1_odt_default Default ODT value of rank 1 when there is no read/write activity
11	RW	0x0	rank1_odt_write_sel Enable/disable ODT for rank 1 when a write access is occurring on this rank
10	RW	0x0	rank1_odt_write_nse Enable/disable ODT for rank 1 when a write access is occurring on a different rank

Bit	Attr	Reset Value	Description
9	RW	0x0	rank1_odt_read_sel Enable/disable ODT for rank 1 when a read access is occurring on this rank
8	RW	0x0	rank1_odt_read_nsel Enable/disable ODT for rank 1 when a read access is occurring on a different rank
7:5	RO	0x0	reserved
4	RW	0x0	rank0_odt_default Default ODT value of rank 0 when there is no read/write activity
3	RW	0x0	rank0_odt_write_sel Enable/disable ODT for rank 0 when a write access is occurring on this rank
2	RW	0x0	rank0_odt_write_nsel Enable/disable ODT for rank 0 when a write access is occurring on a different rank
1	RW	0x0	rank0_odt_read_sel Enable/disable ODT for rank 0 when a read access is occurring on this rank
0	RW	0x0	rank0_odt_read_nsel Enable/disable ODT for rank 0 when a read access is occurring on a different rank

DDR_PCTL_DFIODTCFG1

Address: Operational Base + offset (0x0248)

DFI ODT Timing Configuration 1 (for Latency and Length)

Bit	Attr	Reset Value	Description
31:27	RO	0x0	reserved
26:24	RW	0x6	odt_len_bl8_r ODT length for BL8 read transfers Length of dfi_odt signal for BL8 reads. This is in terms of SDR cycles. For BL4 reads, the length of dfi_odt is always 2 cycles shorter than the value in this register field.
23:19	RO	0x0	reserved
18:16	RW	0x6	odt_len_bl8_w ODT length for BL8 write transfers Length of dfi_odt signal for BL8 writes. This is in terms of SDR cycles. For BL4 writes, the length of dfi_odt is always 2 cycles shorter than the value in this register field.
15:13	RO	0x0	reserved
12:8	RW	0x00	odt_lat_r Field0000 Abstract ODT latency for reads Latency after a read command that dfi_odt is set. This is in terms of SDR cycles.
7:5	RO	0x0	reserved

Bit	Attr	Reset Value	Description
4:0	RW	0x00	odt_lat_w ODT latency for writes Latency after a write command that dfi_odt is set. This is in terms of SDR cycles

DDR_PCTL_DFIODTRANKMAP

Address: Operational Base + offset (0x024c)

DFI ODT Rank Mapping

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5:4	RW	0x2	odt_rank_map1 Rank mapping for dfi_odt[1] Determines which rank access(es) will cause dfi_odt[1] to be asserted Bit 5= 1: dfi_odt[1] will be asserted to terminate rank 1 Bit 4= 1: dfi_odt[1] will be asserted to terminate rank 0
3:2	RO	0x4	reserved
1:0	RW	0x1	odt_rank_map0 Rank mapping for dfi_odt[0] Determines which rank access(es) will cause dfi_odt[0] to be asserted Bit 1= 1: dfi_odt[0] will be asserted to terminate rank 1 Bit 0= 1: dfi_odt[0] will be asserted to terminate rank 0

DDR_PCTL_DFITPHYWRDATA

Address: Operational Base + offset (0x0250)

DFI tphy_wrdata Register

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5:0	RW	0x01	tphy_wrdata Specifies the number of DFI clock cycles between when the dfi_wrdata_en signal is asserted to when the associated write data is driven on the dfi_wrdata signal. This has no impact on performance, only adjusts the relative time between enable and data transfer.

DDR_PCTL_DFITPHYWRLAT

Address: Operational Base + offset (0x0254)

DFI tphy_wrlat Register

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5:0	RW	0x01	tphy_wrlat Specifies the number of DFI clock cycles between when a write command is sent on the DFI control interface and when the dfi_wrdata_en signal is asserted.

DDR_PCTL_DFITRDDATAEN

Address: Operational Base + offset (0x0260)

DFI trddata_en Register

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5:0	RW	0x01	trddata_en Specifies the number of DFI clock cycles from the assertion of a read command on the DFI to the assertion of the dfi_rddata_en signal.

DDR_PCTL_DFITPHYRDLAT

Address: Operational Base + offset (0x0264)

DFI tphy_rdlat Register

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5:0	RW	0x0f	tphy_rdlat Specifies the maximum number of DFI clock cycles allowed from the assertion of the dfi_rddata_en signal to the assertion of the dfi_rddata_valid signal.

DDR_PCTL_DFITPHYUPDTYPE0

Address: Operational Base + offset (0x0270)

DFI tphyupd_type0 Register

Bit	Attr	Reset Value	Description
31:12	RO	0x0	reserved
11:0	RW	0x010	tphyupd_type0 Specifies the maximum number of DFI clock cycles that the dfi_phyupd_req signal may remain asserted after the assertion of the dfi_phyupd_ack signal for dfi_phyupd_type = 0x0. The dfi_phyupd_req signal may de-assert at any cycle after the assertion of the dfi_phyupd_ack signal.

DDR_PCTL_DFITPHYUPDTYPE1

Address: Operational Base + offset (0x0274)

DFI tphyupd_type1 Register

Bit	Attr	Reset Value	Description
31:12	RO	0x0	reserved
11:0	RW	0x010	tphyupd_type1 Specifies the maximum number of DFI clock cycles that the dfi_phyupd_req signal may remain asserted after the assertion of the dfi_phyupd_ack signal for dfi_phyupd_type = 0x1. The dfi_phyupd_req signal may de-assert at any cycle after the assertion of the dfi_phyupd_ack signal.

DDR_PCTL_DFITPHYUPDTYPE2

Address: Operational Base + offset (0x0278)

DFI tphyupd_type2 Register

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:12	RO	0x0	reserved
11:0	RW	0x010	tphyupd_type2 Specifies the maximum number of DFI clock cycles that the dfi_phyupd_req signal may remain asserted after the assertion of the dfi_phyupd_ack signal for dfi_phyupd_type = 0x2. The dfi_phyupd_req signal may de-assert at any cycle after the assertion of the dfi_phyupd_ack signal.

DDR_PCTL_DFITPHYUPDTYPE3

Address: Operational Base + offset (0x027c)

DFI tphyupd_type3 Register

Bit	Attr	Reset Value	Description
31:12	RO	0x0	reserved
11:0	RW	0x010	tphyupd_type3 Specifies the maximum number of DFI clock cycles that the dfi_phyupd_req signal may remain asserted after the assertion of the dfi_phyupd_ack signal for dfi_phyupd_type = 0x3. The dfi_phyupd_req signal may de-assert at any cycle after the assertion of the dfi_phyupd_ack signal.

DDR_PCTL_DFITCTRLUPDMIN

Address: Operational Base + offset (0x0280)

DFI tctrlupd_min Register

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0010	tctrlupd_min Specifies the minimum number of DFI clock cycles that the dfi_ctrlupd_req signal must be asserted.

DDR_PCTL_DFITCTRLUPDMAX

Address: Operational Base + offset (0x0284)

DFI tctrlupd_max Register

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0040	tctrlupd_max Specifies the maximum number of DFI clock cycles that the dfi_ctrlupd_req signal can assert.

DDR_PCTL_DFITCTRLUPDDLY

Address: Operational Base + offset (0x0288)

DFI tctrlupddly Register

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved

Bit	Attr	Reset Value	Description
3:0	RW	0x8	tctrlupd_dly Delay in DFI clock cycles between time a controller-initiated update could be started and time controller-initiated update actually starts (dfi_ctrlupd_req going high).

DDR_PCTL_DFIUPDCFG

Address: Operational Base + offset (0x0290)

DFI Update Configuration Register

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1	RW	0x1	dfi_phyupd_en Enables the support for acknowledging PHY-initiated updates: 1'b0 = Disabled 1'b1 = Enabled
0	RW	0x1	dfi_ctrlupd_en Enables the generation of controller-initiated updates: 1'b0 = Disabled 1'b1 = Enabled

DDR_PCTL_DFITREFMSKI

Address: Operational Base + offset (0x0294)

DFI Masked Refresh Interval

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RW	0x00	trefmski Time period of the masked Refresh interval This value is only used if TREFI==0. Defines the time period (in 100ns units) of the masked Refresh (REFMSK) interval. The actual time period defined is DFITREFMSKI* TOGCNT100N * internal timers clock period.

DDR_PCTL_DFITCTRLUPDI

Address: Operational Base + offset (0x0298)

DFI tctrlupd_interval Register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	tctrlupd_interval DFI controller-initiated updates interval, measured in terms of Refresh interval units. If TREFI!=0, the time period is defined as DFITCTRLUPDI*TREFI * TOGCNT100N * internal timers clock period. If TREFI==0 and DFITREFMSKI!=0, the period changes to DFITCTRLUPDI * DFITREFMSKI * TOGCNT100N * internal timers clock period. Programming a value of 0 is the same as programming a value of 1; for instance, a controller-initiated update occurs every Refresh interval.

DDR_PCTL_DFITRCFG0

Address: Operational Base + offset (0x02ac)

DFI Training Configuration 0 Register

Bit	Attr	Reset Value	Description
31:20	RO	0x0	reserved
19:16	RW	0x0	dfi_wrlvl_rank_sel Determines the value to drive on the output signal dfi_wrlvl_cs_n. The value on dfi_wrlvl_cs_n is the inverse of the setting in this field.
15:13	RO	0x0	reserved
12:4	RW	0x000	dfi_rdlvl_edge Determines the value to drive on the output signal dfi_rdlvl_edge. The value on dfi_rdlvl_edge is the same as the setting in this field.
3:0	RW	0x0	dfi_rdlvl_rank_sel Determines the value to drive on the output signal dfi_rdlvl_cs_n. The value on dfi_rdlvl_cs_n is the inverse of the setting in this field.

DDR_PCTL_DFITRSTAT0

Address: Operational Base + offset (0x02b0)

DFI Training Status 0 Register

Bit	Attr	Reset Value	Description
31:18	RO	0x0	reserved
17:16	RO	0x0	dfi_wrlvl_mode Reports the value of the input signal dfi_wrlvl_mode.
15:10	RO	0x0	reserved
9:8	RO	0x0	dfi_rdlvl_gate_mode Reports the value of the input signal dfi_rdlvl_gate_mode.
7:2	RO	0x0	reserved
1:0	RO	0x0	dfi_rdlvl_mode Reports the value of the input signal dfi_rdlvl_mode.

DDR_PCTL_DFITRWRLVLEN

Address: Operational Base + offset (0x02b4)

DFI Training dfi_wrlvl_en Register

Bit	Attr	Reset Value	Description
31:9	RO	0x0	reserved
8:0	RW	0x000	dfi_wrlvl_en Determines the value to drive on the output signal dfi_wrlvl_en.

DDR_PCTL_DFITRRDLVLEN

Address: Operational Base + offset (0x02b8)

DFI Training dfi_rdlvl_en Register

Bit	Attr	Reset Value	Description
31:9	RO	0x0	reserved
8:0	RW	0x000	dfi_rdlvl_en Determines the value to drive on the output signal dfi_rdlvl_en.

DDR_PCTL_DFITRRDLVLGATEEN

Address: Operational Base + offset (0x02bc)

DFI Training dfi_rdlvl_gate_en Register

Bit	Attr	Reset Value	Description
31:9	RO	0x0	reserved
8:0	RW	0x000	dfi_rdlvl_gate_en Determines the value to drive on the output signal dfi_rdlvl_gate_en.

DDR_PCTL_DFISTSTAT0

Address: Operational Base + offset (0x02c0)

DFI Status Status 0 Register

Bit	Attr	Reset Value	Description
31:25	RO	0x0	reserved
24:16	RO	0x000	dfi_data_byte_disable Reports the value of the output signal dfi_data_byte_disable.
15:6	RO	0x0	reserved
5:4	RO	0x0	dfi_freq_ratio Reports the value of the output signal dfi_freq_ratio.
3:2	RO	0x0	reserved
1	RO	0x0	dfi_init_start Reports the value of the output signal dfi_init_start.
0	RO	0x0	dfi_init_complete Reports the value of the input signal dfi_init_complete.

DDR_PCTL_DFISTCFG0

Address: Operational Base + offset (0x02c4)

DFI Status Configuration 0 Register

Bit	Attr	Reset Value	Description
31:3	RO	0x0	reserved
2	RW	0x0	dfi_data_byte_disable_en Enables the driving of the dfi_data_byte_disable signal. The value driven on dfi_data_byte_disable is dependent on the setting of PPCFG register. 1'b0 - Drive dfi_data_byte_disable to default value of all zeroes. 1'b1 - Drive dfi_data_byte_disable according to value as defined by PPCFG register setting. Note: should be set to 1'b1 only after PPCFG is correctly set.
1	RO	0x0	reserved
0	RW	0x0	dfi_init_start Sets the value of the dfi_init_start signal. 1'b0 - dfi_init_start is driven low 1'b1 - dfi_init_start is driven high

DDR_PCTL_DFISTCFG1

Address: Operational Base + offset (0x02c8)

DFI Status Configuration 1 Register

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	dfi_dram_clk_disable_en Enables support of the dfi_dram_clk_disable signal with Self-Refresh (SR). 1'b0 - Disable dfi_dram_clk_disable support in relation to SR 1'b1 - Enable dfi_dram_clk_disable support in relation to SR

DDR_PCTL_DFITDRAMCLKEN

Address: Operational Base + offset (0x02d0)

DFI tdram_clk_enable Register

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3:0	RW	0x2	tdram_clk_enable Specifies the number of DFI clock cycles from the de-assertion of the dfi_dram_clk_disable signal on the DFI until the first valid rising edge of the clock to the DRAM memory devices, at the PHY-DRAM boundary. If the DFI clock and the memory clock are not phase-aligned, this timing parameter should be rounded up to the next integer value.

DDR_PCTL_DFITDRAMCLKDIS

Address: Operational Base + offset (0x02d4)

DFI tdram_clk_disable Register

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3:0	RW	0x2	tdram_clk_disable Specifies the number of DFI clock cycles from the assertion of the dfi_dram_clk_disable signal on the DFI until the clock to the DRAM memory devices, at the PHY-DRAM boundary, maintains a low value. If the DFI clock and the memory clock are not phase-aligned, this timing parameter should be rounded up to the next integer value.

DDR_PCTL_DFISTCFG2

Address: Operational Base + offset (0x02d8)

DFI Status Configuration 2 Register

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved

Bit	Attr	Reset Value	Description
1	RW	0x0	parity_en Enables the DFI parity generation feature (driven on output signal dfi_parity_in) 1'b0 - Disable DFI parity generation 1'b1 - Enable DFI parity generation
0	RW	0x0	parity_intr_en Enable interrupt generation for DFI parity error (from input signal dfi_parity_error). 1'b0 - Disable interrupt 1'b1 - Enable interrupt

DDR_PCTL_DFISTPARCLR

Address: Operational Base + offset (0x02dc)

DFI Status Parity Clear Register

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1	R/W SC	0x0	parity_log_clr Set this bit to 1'b1 to clear the DFI Status Parity Log register (DFISTPARLOG). 1'b0 = Do not clear DFI status Parity Log register 1'b1 = Clear DFI status Parity Log register
0	R/W SC	0x0	parity_intr_clr Set this bit to 1'b1 to clear the interrupt generated by a DFI parity error (as enabled by DFISTCFG2.parity_intr_en). It also clears the INTRSTAT.parity_intr register field. It is automatically cleared by hardware when the interrupt has been cleared.

DDR_PCTL_DFISTPARLOG

Address: Operational Base + offset (0x02e0)

DFI Status Parity Log Register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	parity_err_cnt Increments any time the DFI parity logic detects a parity error(s) (on dfi_parity_error).

DDR_PCTL_DFLPCFG0

Address: Operational Base + offset (0x02f0)

DFI Low Power Configuration 0 Register

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:28	RW	0x0	<p>dfi_lp_wakeup_dpd</p> <p>Value to drive on dfi_lp_wakeup signal when Deep Power Down mode is entered. Determines the DFI's tlp_wakeup time:</p> <ul style="list-style-type: none"> 4'b0000 - 16 cycles 4'b0001 - 32 cycles 4'b0010 - 64 cycles 4'b0011 - 128 cycles 4'b0100 - 256 cycles 4'b0101 - 512 cycles 4'b0110 - 1024 cycles 4'b0111 - 2048 cycles 4'b1000 - 4096 cycles 4'b1001 - 8192 cycles 4'b1010 - 16384 cycles 4'b1011 - 32768 cycles 4'b1100 - 65536 cycles 4'b1101 - 131072 cycles 4'b1110 - 262144 cycles 4'b1111 - Unlimited
27:25	RO	0x0	reserved
24	RW	0x0	<p>dfi_lp_en_dpd</p> <p>Enables DFI Low Power interface handshaking during Deep Power Down Entry/Exit.</p> <ul style="list-style-type: none"> 1'b0 - Disabled 1'b1 - Enabled
23:20	RO	0x0	reserved
19:16	RW	0x7	<p>dfi_tlp_resp</p> <p>Setting for tlp_resp time. Same value is used for both Power Down and Self-refresh and Deep Power Down modes. DFI 2.1 specification, recommends using value of 7 always.</p>

Bit	Attr	Reset Value	Description
15:12	RW	0x0	<p>dfi_lp_wakeup_sr Value to drive on dfi_lp_wakeup signal when Self-Refresh mode is entered. Determines the DFI's tlp_wakeup time:</p> <p>4'b0000 - 16 cycles 4'b0001 - 32 cycles 4'b0010 - 64 cycles 4'b0011 - 128 cycles 4'b0100 - 256 cycles 4'b0101 - 512 cycles 4'b0110 - 1024 cycles 4'b0111 - 2048 cycles 4'b1000 - 4096 cycles 4'b1001 - 8192 cycles 4'b1010 - 16384 cycles 4'b1011 - 32768 cycles 4'b1100 - 65536 cycles 4'b1101 - 131072 cycles 4'b1110 - 262144 cycles 4'b1111 - Unlimited</p>
11:9	RO	0x0	reserved
8	RW	0x0	<p>dfi_lp_en_sr Enables DFI Low Power interface handshaking during Self-Refresh Entry/Exit.</p> <p>1'b0 - Disabled 1'b1 - Enabled</p>
7:4	RW	0x0	<p>dfi_lp_wakeup_pd Value to drive on dfi_lp_wakeup signal when Power Down mode is entered. Determines the DFI's tlp_wakeup time:</p> <p>4'b0000 - 16 cycles 4'b0001 - 32 cycles 4'b0010 - 64 cycles 4'b0011 - 128 cycles 4'b0100 - 256 cycles 4'b0101 - 512 cycles 4'b0110 - 1024 cycles 4'b0111 - 2048 cycles 4'b1000 - 4096 cycles 4'b1001 - 8192 cycles 4'b1010 - 16384 cycles 4'b1011 - 32768 cycles 4'b1100 - 65536 cycles 4'b1101 - 131072 cycles 4'b1110 - 262144 cycles 4'b1111 - Unlimited</p>
3:1	RO	0x0	reserved

Bit	Attr	Reset Value	Description
0	RW	0x0	dfi_lp_en_pd Enables DFI Low Power interface handshaking during Power Down Entry/Exit. 1'b0 - Disabled 1'b1 - Enabled

DDR_PCTL_DFITRWRLVLRESP0

Address: Operational Base + offset (0x0300)

DFI Training dfi_wrlvl_resp Status 0 Register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	dfi_wrlvl_resp0 Reports the status of the dif_wrlvl_resp[31:0] signal.

DDR_PCTL_DFITRWRLVLRESP1

Address: Operational Base + offset (0x0304)

DFI Training dfi_wrlvl_resp Status 1 Register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	dfi_wrlvl_resp1 Reports the status of the dif_wrlvl_resp[63:32] signal.

DDR_PCTL_DFITRWRLVLRESP2

Address: Operational Base + offset (0x0308)

DFI Training dfi_wrlvl_resp Status 2 Register

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RO	0x00	dfi_wrlvl_resp2 Reports the status of the dif_wrlvl_resp[71:64] signal.

DDR_PCTL_DFITRRDLVLRESP0

Address: Operational Base + offset (0x030c)

DFI Training dfi_rdlvl_resp Status 0 Register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	dfi_rdlvl_resp0 Reports the status of the dif_rdlvl_resp[31:0] signal.

DDR_PCTL_DFITRRDLVLRESP1

Address: Operational Base + offset (0x0310)

DFI Training dfi_rdlvl_resp Status 1 Register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	dfi_rdlvl_resp1 Reports the status of the dif_rdlvl_resp[63:32] signal.

DDR_PCTL_DFITRRDLVLRESP2

Address: Operational Base + offset (0x0314)

DFI Training dfi_rdlvl_resp Status 2 Register

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RO	0x00	dfi_rdlvl_resp2 Reports the status of the dif_rdlvl_resp[71:64] signal.

DDR_PCTL_DFITRWRLVLDELAY0

Address: Operational Base + offset (0x0318)

DFI Training dfi_wrlvl_delay Configuration 0 Register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	dfi_wrlvl_delay0 Sets the value to be driven on the signal dfi_wrlvl_delay_x[31:0].

DDR_PCTL_DFITRWRLVLDELAY1

Address: Operational Base + offset (0x031c)

DFI Training dfi_wrlvl_delay Configuration 1 Register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	dfi_wrlvl_delay1 Sets the value to be driven on the signal dfi_wrlvl_delay_x[63:32].

DDR_PCTL_DFITRWRLVLDELAY2

Address: Operational Base + offset (0x0320)

DFI Training dfi_wrlvl_delay Configuration 2 Register

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RW	0x00	dfi_wrlvl_delay2 Sets the value to be driven on the signal dfi_wrlvl_delay_x[71:64].

DDR_PCTL_DFITRRDLVLDELAY0

Address: Operational Base + offset (0x0324)

DFI Training dfi_rdlvl_delay Configuration 0 Register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	dfi_rdlvl_delay0 Sets the value to be driven on the signal dfi_rdlvl_delay_x[31:0].

DDR_PCTL_DFITRRDLVLDELAY1

Address: Operational Base + offset (0x0328)

DFI Training dfi_rdlvl_delay Configuration 1 Register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	dfi_rdlvl_delay1 Sets the value to be driven on the signal dfi_rdlvl_delay_x[63:32].

DDR_PCTL_DFITRRDLVLDELAY2

Address: Operational Base + offset (0x032c)

DFI Training dfi_rdlvl_delay Configuration 2 Register

Bit	Attr	Reset Value	Description
------------	-------------	--------------------	--------------------

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RW	0x00	dfi_rdlvl_delay2 Sets the value to be driven on the signal dfi_rdlvl_delay_x[71:64].

DDR_PCTL_DFITRRDLVLGATEDELAY0

Address: Operational Base + offset (0x0330)

DFI Training dfi_rdlvl_gate_delay Configuration 0

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	dfi_rdlvl_gate_delay0 Sets the value to be driven on the signal dfi_rdlvl_gate_delay_x[31:0].

DDR_PCTL_DFITRRDLVLGATEDELAY1

Address: Operational Base + offset (0x0334)

DFI Training dfi_rdlvl_gate_delay Configuration 1

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	dfi_rdlvl_gate_delay1 Sets the value to be driven on the signal dfi_rdlvl_gate_delay_x[63:32].

DDR_PCTL_DFITRRDLVLGATEDELAY2

Address: Operational Base + offset (0x0338)

DFI Training dfi_rdlvl_gate_delay Configuration 2

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RW	0x00	dfi_rdlvl_gate_delay2 Sets the value to be driven on the signal dfi_rdlvl_gate_delay_x[71:64].

DDR_PCTL_DFITRCMD

Address: Operational Base + offset (0x033c)

DFI Training Command Register

Bit	Attr	Reset Value	Description
31	R/W SC	0x0	dfitrcmd_start DFI Training Command Start. When this bit is set to 1, the command operation defined in the dfitrcmd_opcode field is started. This bit is automatically cleared by the controller after the command is finished. The application can poll this bit to determine when controller is ready to accept another command. This bit cannot be cleared to 1b0 by software.
30:13	RO	0x0	reserved
12:4	RW	0x000	dfitrcmd_en DFI Training Command Enable. Selects which bits of chosen DFI Training command to drive to 1'b1.
3:2	RO	0x0	reserved

Bit	Attr	Reset Value	Description
1:0	RW	0x0	dfitrcmd_opcode DFI Training Command Opcode. Select which DFI Training command to generate for one n_clk cycle: 2'b00 - dfi_wrlvl_load 2'b01 - dfi_wrlvl_strobe 2'b10 - dfi_rdlvl_load 2'b11 - Reserved.

DDR_PCTL_IPVR

Address: Operational Base + offset (0x03f8)

IP Version Register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	ip_version ASCII value for each number in the version.

DDR_PCTL_IPTR

Address: Operational Base + offset (0x03fc)

IP Type Register

Bit	Attr	Reset Value	Description
31:0	RO	0x44574300	ip_type Contains the IP's identification code, which is an ASCII value to identify the component and it is currently set to the string "DWC". This value never changes.

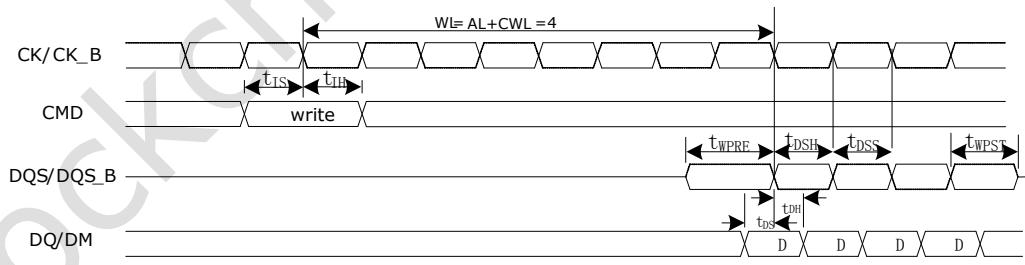
6.5 Timing Diagram**6.5.1 DDR3 Read/Write Access Timing**

Fig. 6-4 DDR3 burst write operation: AL=0, CWL=4, BC4

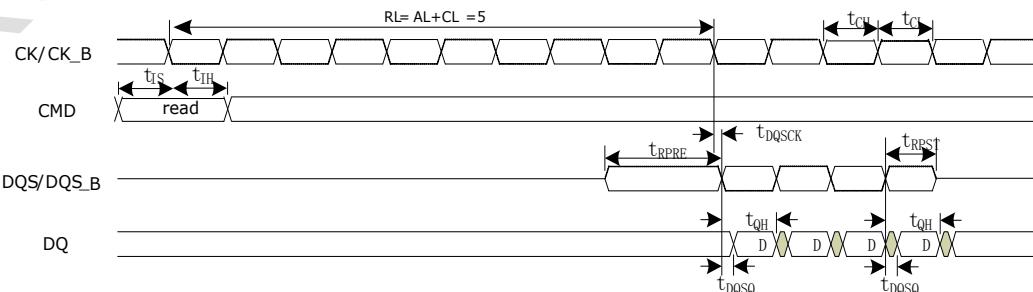


Fig. 6-5 DDR3 burst read operation: AL=0, CL=5, BC4

Table 6-1 meaning of the parameter

Parameter	Description	DDR3-800	unit
-----------	-------------	----------	------

		min	max	
t _{CH}	CK HIGH pulse width	0.43	-	tCK
t _{CL}	CK LOW pulse width	0.43	-	tCK
t _{DS}	DQ and DM input setup time (differential strobe)	75	-	ps
t _{DH}	DQ and DM input hold time (differential strobe)	150	-	ps
t _{DSS}	DQS falling edge to CK setup time	0.2	-	tCK
t _{DSH}	DQS falling edge hold time from CK	0.2	-	tCK
t _{IS}	Address and control input setup time	200	-	ps
t _{IH}	Address and control input hold time	275	-	ps
t _{WPRE}	Write preamble	0.9	-	tCK
t _{WPST}	Write postamble	0.3	-	tCK
t _{RPRE}	Read preamble	0.9	1.1	tCK
t _{RPST}	Read postamble	0.3	0.5	tCK
t _{DQSCK}	DQS output access time from CK/CK_n	-400	+400	ps
t _{DQSQ}	DQS-DQ skew for DQS and associated DQ signals	-	200	ps
t _{QH}	DQ/DQS output hold time from DQS	0.38	-	tCK

6.6 Interface Description

DDR IOs are listed as following Table.

Table 6-2 DDR IO description

Pin Name	Description
CK	Active-high clock signal to the memory device.
CK_B	Active-low clock signal to the memory device.
CKE	Active-high clock enable signal to the memory device for two chip select.
CS_Bi (i=0,1)	Active-low chip select signal to the memory device. There are two chip select.
RAS_B	Active-low row address strobe to the memory device.
CAS_B	Active-low column address strobe to the memory device.
WE_B	Active-low write enable strobe to the memory device.
BA[2:0]	Bank address signal to the memory device.
A[15:0]	Address signal to the memory device.
DQ[15:0]	Bidirectional data line to the memory device.
DQS[1:0]	Active-high bidirectional data strobes to the memory device.
DQS_B[1:0]	Active-low bidirectional data strobes to the memory device.
DM[1:0]	Active-low data mask signal to the memory device.
ODTi (i=0,1)	On-Die Termination output signal for two chip select.
RESET	DDR3 reset signal.

6.7 Application Notes

6.7.1 State transition of PCTL

To operate PCTL, the programmer must be familiar with the available operational states and how to transition to each state from the current state.

Every software programmable register is accessible only during certain operational states. For information about what registers are accessible in each state, refer to "Software Registers," which provides this information in each register description. The general rule is that the PCTL must be in the Init_mem or ConFig states to successfully write most of the registers.

The following tables provide the programming sequences for moving to the various states of the state machine.

Moving to the Init_mem State

Step	Application	PCTL
1	Read STAT register	Returns the current PCTL state.
2	If STAT.ctl_stat = Init_mem, go to END.	
3	If STAT.ctl_stat = Config, go to Step9.	
4	If STAT.ctl_stat = Access, go to Step8.	
5	If STAT.ctl_stat = Low_power, go to Step7.	
6	Goto Step1.	PCTL is in a Transitional state and not in any of the previous operational states.
7	Write WAKEUP to SCTL.state_cmd and poll STAT.ctl_stat = Access.	Issues SRX, moves to the Access state, updates STAT.ctl_stat = Access when complete.
8	Write CFG to SCTL.state_cmd and poll STAT.ctl_stat = ConFig	PCTL stalls the NIF; completes any pending transaction; issues PREA if required; moves into the ConFig state; updates STAT.ctl_stat = ConFig when complete.
9	Write INIT to SCTL.state_cmd and poll STAT.ctl_stat = Init_mem	Moves into the Init_mem state and updates STAT.ctl_stat = Init_mem.
END		PCTL is in Init_mem state.

Moving to ConFig State

Step	Application	PCTL
1	Read STAT register.	Returns the current PCTL state.
2	If STAT.ctl_stat = Config, goto END.	
3	If STAT.ctl_stat = Low_power, go to Step6.	
4	If STAT.ctl_stat = Init_mem or Access, go to Step7.	
5	Go to Step1.	PCTL is in a transitional state and is not in any of the previous operational states.
6	Write WAKEUP to CTL.state_cmd and poll STAT.ctl_stat = Access.	Issues SRX, moves to the Access state, and updates STAT.ctl_stat = Access when complete.
7	Write CFG to SCTL.state_cmd and poll STAT.ctl_stat = ConFig	PCTL stalls the NIF; completes any pending transaction; issues PREA if required; moves into the ConFig state; and updates STAT.ctl_stat = ConFig when complete.
END		PCTL is in ConFig state.

Moving to Access State

Step	Application	PCTL
1	Read STAT register	Returns the current PCTL state.
2	If STAT.ctl_stat = Access, go to END.	
3	If STAT.ctl_stat = Config, go to Step9	
4	If STAT.ctl_stat = Init_mem, go to Step8	

Step	Application	PCTL
5	If STATctl_stat = Low_power, go to Step7.	
6	Goto Step1.	PCTL is in a transitional state and is not in any of the previous operational states.
7	Write WAKEUP to SCTL.state_cmd and poll STATctl_stat = Access. Goto END	Issues SRX, moves to the Access state, updates STATctl_stat = Access when complete.
8	Write CFG to SCTL.state_cmd and poll STATctl_stat = ConFig	Moves into the ConFig state, updates STATctl_stat = ConFig when complete.
9	Write GO to SCTL.state_cmd and poll STATctl_stat = Access.	Moves into the Access state, updates STATctl_stat = Access when complete.
END		PCTL is in Access state.

Moving to Low Power State

Step	Application	PCTL
1	Read STAT register.	Returns current PCTL state.
2	If STATctl_stat = Low_power, go to END.	
3	If STATctl_stat = Access, go to Step9	
4	If STATctl_stat = Config, go to Step8	
5	If STATctl_stat = Init_mem, go to Step7.	
6	Goto Step1.	PCTL is in transitional state and is not in any of the previous operational states.
7	Write CFG to SCTL.state_cmd and poll STATctl_stat = ConFig	Moves into the ConFig state, updates STATctl_stat = ConFig when complete.
8	Write GO to SCTL.state_cmd and poll STATctl_stat = Access.	Moves into the Access state, updates STATctl_stat = Access when complete.
9	Write SLEEP to SCTL.state_cmd and poll STATctl_stat = Low_power.	Issues PDX if necessary; completes any pending transactions; issues PREA command; finally, issues SRE and updates STATctl_stat = Low_power.
END		PCTL is in Low Power state

6.7.2 Initialization**PHY Initialization**

DDR PHY power-up reset sequence:

1. PHY Register Reset: reset the PHY register block through presetn;
2. Configure registers: AL, CL etc. (register address 0x38)
3. System reset: reset the PHY through system_rstn;
4. Soft reset (optional): after system reset, execute the soft reset by changing PHY_REG0 values. Soft reset function is on when PHY_REG0[3:2] = 2'b11.
5. Start PHY initialization: after pclk, dfi_clk1x and dfi_clk2x clock signals are stable, initialize the PHY.
6. Start PHY Calibration: after ddr sdram initialization done, you can set PHY calibration start.
7. Close PHY Calibration: after step 5us, you can set close PHY calibration.
8. Start Write and Read.

DDR3 Initialization Sequence

The initialization steps for DDR3 SDRAMs are as follows:

1. Optionally maintain RESET# low for a minimum of either 200 us (power-up initialization) or

100ns (power-on initialization). The DDR PHY drives RESET# low from the beginning of reset assertion and therefore this step may be skipped when DRAM initialization is triggered if enough time may already have expired to satisfy the RESET# low time.

2. After RESET# is de-asserted, wait a minimum of 500 us with CKE low.
3. Apply NOP and drive CKE high.
4. Wait a minimum of tXPR.
5. Issue a load Mode Register 2 (MR2) command.
6. Issue a load Mode Register 3 (MR3) command.
7. Issue a load Mode Register (MR1) command (to set parameters and enable DLL).
8. Issue a load Mode Register (MR0) command to set parameters and reset DLL.
9. Issue ZQ calibration command.
10. Wait 512 SDRAM clock cycles for the DLL to lock (tDLLK) and ZQ calibration (tZQinit) to finish. This wait time is relative to Step 8, i.e. relative to when the DLL reset command was issued onto the SDRAM command bus.

6.7.3 Low Power Operation

Low_power state can be entered/exited via following ways:

- Software control of PCTL State machine (highest priority)
- Hardware Low Power Interface (middle priority)
- Auto Self-Refresh feature (lowest priority)

Note the priority of requests from Access to Low_power is highlighted above. The STAT.ip_trig register field reports which of the 3 requests caused the entry to Low_power state.

Software control of PCTL State

The application can request via software to enter the memories into Self-Refresh state by issuing the SLEEP command by programming SCTL.PCTL responds to the software request by moving into the Low_power operational state and issuing the SRE command to the memories. Note that the Low_power state can only be reached from the Access state.

In a similar fashion, the application requests to exit the memories from Self-Refresh by issuing a WAKEUP command by programming SCTL. PCTL responds to the WAKEUP command issuing SRX and restoring normal NIF address channel operation.

Hardware Low Power Interface

The hardware low power interface can also be used to enter/exit Self-Refresh. The functionality is enabled by setting SCFG.hw_low_power_en=1. Once that bit is set, the input c_sysreq has the ability to trigger entry into the Low Power configuration state just like the software methodology (SCTL.state_cmd = SLEEP). A hardware Low Power entry trigger will be ignored/denied if the input c_active_in=1 or n_valid=1. It may be accepted if c_active_in=0 and n_valid=0, depending on the current state of the PCTL. When SCFG.hw_low_power_en=1, the outputs c_sysack and c_active provide feedback as required by the AXI low power interface specification (this interface's operation is defined by the AXI specification). c_sysack acknowledges the request to go into the Low_power state, and c_active indicates when the PCTL is actually in the Low_power state.

The c_active output could also be used by an external Low Power controller to decide when to request a transition to low power. When MCFG1.hw_idle > 0, c_active = 1'b0 indicates that the NIF has been idle for at least MCFG1.hw_idle * 32 * n_clk cycles while in the Access state. When in low power the c_active output can be used by an external Low Power controller to trigger a low power exit. c_active will be driven high when either c_active_in or n_valid are high. The path from c_active_in and n_valid to c_active is asynchronous so even if the clocks have been removed c_active will assert. The Low Power controller should re-enable the clocks when c_active is driven high while in the Low_power state.

Auto Power Down/Self-Refresh

The Power Down and/or Self-Refresh sequence is automatically started by PCTL when the NIF address channel is idle for a number of cycles, depending on the programmed value in MCFG.pd_idle and MCFG1.sr_idle.

Following table outlines the effect of these settings in conjunction with NIF being idle.

pd_idle	sr_idle	Memory modes	Memory Type
0	0	none	All
>0	0	Power Down	All

pd_idle	sr_idle	Memory modes	Memory Type
0	>0	Self-Refresh	All
>0	>0	Power Down -> Self Refresh ³	All

Note:

1. Power Down is entered if NIF is idle for pd_idle. Following on from that, if NIF continues to be idle for a further sr_idle*32 cycles, Power Down is exited and Self-Refresh is entered.
2. Following on from that, if NIF continues to be idle for a further sr_idle*32 cycles, Power Down is exited and Self-Refresh is entered.

Removing PCTL's n_clk

In DDR3, the relationship between SRE/SRX and stopping/starting the memory clock (CK) are formalized and are accounted for automatically by PCTL. With DDR3, CK should only be stopped after PCTL has reached the Low_power state. The current operational state can be verified by reading STAT.ctl_stat. The CK must be started and stable before the Software or Hardware Low Power Interface attempts to take the memory out of Self-Refresh.

PCTL's n_clk can be safely removed when PCTL is in Low Power state. The sequences outlined in following two tables should be followed for safe operation:

Step	Application	PCTL
1	Write SLEEP to SCTL.state_cmd and poll STAT.ctl_stat = LOW_POWER.	Tells PCTL to move memories into Self-Refresh and waits until this completes.
2	Write TREFI=0. Also, write DFITCRLUPDI=0 and DFIREFMSKI=0, if they are not already 0.	Stops any MC-driven DFI updates occurring internally with PCTL
3	Wait a minimum interval which is equivalent to the PCTL's Refresh Interval (previous value of TREFI * TOGCNT100N * internal timers clock period;	Ensures any already scheduled PHY/PVT updates have completed successfully.
4	Stop toggling n_clk to PCTL.	n_clk logic inside PCTL is stopped.
end		

Step	Application	PCTL
1	Drive c_active_in low	Confirms that system external to PCTL can accept a Low-power request
2	Drive c_sysreq low	System Low-power request
3	Wait for PCTL to drive c_sysack low	PCTL Low-power request acknowledgement
4	Check value of c_active when Step 3 occurs. - if c_active=1, request denied. Cannot remove n_clk. Go to END. - if c_active=0, request accepted.	PCTL low-power request status response
5	Stop toggling n_clk to PCTL	n_clk logic inside PCTL is stopped
end		

6.7.4 TX DLLs

All high speed IO signals' phase can be adjusted by TX DLLs. Table 1-3 illustrates these DLLs.

Table 6-3 DDR PHY TX DLLs Delay Step

Offset	Bit	Control Signal Phase	Default	Description
0x60	2~0	CMD	0x0	CMD DLL delay step
0x68	2~0	CK	0x0	CK DLL delay step
0xd4	2~0	DM0, DQ7~DQ0	0x4	DM and DQ DLL
0x114	2~0	DM1, DQ15~DQ8	0x4	Signal delay step
0xd8	2~0	DQS0, DQSBO	0x0	TX DQS DLL Signal

Offset	Bit	Control Signal Phase	Default	Description
0x118	2~0	DQS1, DQSB1	0x0	delay step

Step 0x0 values means no phase delay, and 0x4 increases delay phase to 90 deg, 0x7 values corresponds to maximum phase delay. All DLLs having 8 delay steps which can get 90 deg phase delay by setting 0x4.

6.7.5 RX DLLs

The RX DLLs are used for sample RX DQS signals with proper phase delay and pulse edges. The DQS squelch (Rx Squelch) signal opens a window for passing RX DQS pulses, both RX DQS and DQS squelch signal phase can be adjusted by corresponding DLLs.

Table 6-4 DDR PHY RX DQS Delay Step

Offset	Bit	Control Signal Phase	Default	Description
0xe0	5~3	Left DQS squelch	0x0	Left DQS squelch delay step
	1~0	DQS0, DQSB0	0x0	Left Rx DQS delay step
0x120	5~3	Right DQS squelch	0x0	Right DQS squelch delay step
	1~0	DQS1, DQSB1	0x0	Right Rx DQS delay step

6.7.6 High Speed IO Drive Strength

The tuning range of driver resistance is 28ohm to 138ohm. By default, 0x5 is 46ohm for DDR3 CMD driver. When the control bit is set to be larger, the drive strength becomes stronger.

Table 6-5 DM/DQ/DQS/CMD Driver output resistance

Offset	Bit	Default	Description
0x54	7~4	0x7	adjustable CMD pull-up resistance
	3~0	0x7	adjustable CMD pull-down resistance
0x70	7~4	0x7	adjustable CK pull-up resistance
	3~0	0x7	adjustable CK pull-down resistance

Table 6-6 DM, DQ Signal Drive Strength Register

Offset	Bit	Default	Description
0x80	7~4	0x7	Low 8bit adjustable DQ/DQS pull-up resistance
	3~0	0x7	Low 8bit adjustable DQ/DQS pull-down resistance
0x84	7~4	0x7	High 8bit adjustable DQ/DQS pull-up resistance
	3~0	0x7	High 8bit adjustable DQ/DQS pull-down resistance
0x88	7~4	0x7	Low 8bit adjustable DQ/DQS RX ODT pull-up resistance
	3~0	0x7	Low 8bit adjustable DQ/DQS RX ODT pull-down resistance
0x8c	7~4	0x7	High 8bit adjustable DQ/DQS RX ODT pull-up resistance
	3~0	0x7	High 8bit adjustable DQ/DQS RX ODT pull-down resistance

The value is larger, the drive strength is stronger.

Table 6-7 DM/DQ/DQS/CMD Driver output resistance with control bit

Control bit	4'b0000	4'b0001	4'b0010	4'b0011	4'b0100	4'b0101	4'b0110	4'b0111
Pull-up resistance	+∞	309ohm	155ohm	103ohm	77ohm	63ohm	52ohm	45ohm
Pull-down resistance	+∞	309ohm	155ohm	103ohm	77ohm	63ohm	52ohm	45ohm
Control bit	4'b1000	4'b1001	4'b1010	4'b1011	4'b1100	4'b1101	4'b1110	4'b1111
Pull-up resistance	77ohm	62ohm	52ohm	44ohm	39ohm	34ohm	31ohm	28ohm
Pull-down resistance	77ohm	62ohm	52ohm	44ohm	39ohm	34ohm	31ohm	28ohm

Table 6-8 DM/DQ/DQS RX ODT resistance with control bit

Control bit	4'b0000	4'b0001	4'b0010	4'b0011	4'b0100	4'b0101	4'b0110	4'b0111
Pull-up resistance	+∞	861ohm	431ohm	287ohm	216ohm	172ohm	145ohm	124ohm
Pull-down resistance	+∞	861ohm	431ohm	287ohm	216ohm	172ohm	145ohm	124ohm
Control bit	4'b1000	4'b1001	4'b1010	4'b1011	4'b1100	4'b1101	4'b1110	4'b1111
Pull-up resistance	215ohm	172ohm	144ohm	123ohm	108ohm	96ohm	86ohm	78ohm
Pull-down resistance	215ohm	172ohm	144ohm	123ohm	108ohm	96ohm	86ohm	78ohm

6.7.7 PHY Low Speed Mode (200MHz)

DDR PHY supports low speed to high speed DDR3 by using two operating mode: normal delay line mode up to 800Mbps or more, low power mode where we support any speed up to 533Mbps. If all TX DLLs are bypassed, the PHY will enter low power state.

The DDR PHY enters low power mode when setting DLLs into Bypass mode. Table 10-9 illustrates related register settings.

Table 6-9 Low Power DLL Setting

Offset	Bit	Default	Low power Setting	Description
0x2b0	4	0x0	0x1	CMD DLL in bypass mode
	3	0x0	0x1	DQS1/DQSB1 DM1, DQ15~DQ8 DLL in bypass mode
	2	0x0	0x1	DQS0/DQSB0 DM0 DQ7~DQ0 DLL in bypass mode
0x60	4	0x0	0x0	CMD DLL not inversion mode
	3	0x1	0x1	CMD DLL enable, active HIGH
	2~0	0x0	0x0	N/A
0xd4	4	0x0	0x1	DM1, DQ15~DQ8 inversion mode
	3	0x1	0x1	DM1, DQ15~DQ8 DLL enable
	2~0	0x4	0x0	N/A
0x114	4	0x0	0x1	DM0, DQ7~DQ0 inversion mode
	3	0x1	0x1	DM1, DQ7~DQ0 DLL enable
	2~0	0x4	0x0	N/A
0xd8	3	0x0	0x0	DQS0/DQSB0 not inversion mode
	2~0	0x0	0x0	N/A
0x118	3	0x0	0x0	DQS1/DQSB1 not inversion mode
	2~0	0x0	0x0	N/A

6.7.8 Per bit de-skew tuning

Per-bit de-skew is designed for compensating PCB trace mismatch, DDR PHY support skew individually adjustable for all PHY signals. There are eight steps for each bit de-skew adjusting, and the adjust resolution under different corners is shown below:

Table 6-10 per-bit de-skew tuning resolution

	ff	tt	ss
de-skew resolution	15ps	20ps	30ps

Pre-bit de-skew is realized with inverter chain delay, per-bit de-skew control signals select how much inverters are connected to data path, the minimum resolution is determined by the two inverters minimum delay.

TX path deskew and RX path deskew employ same delay line, and they have same deskew tuning resolution. Minimum RX deskew tuning resolution can be about 28ps with SMIC55II tt corner process, and we can re-design tuning resolution according to system and customer requirement.

6.7.9 DDR PHY Calibration

DDR PHY auto-adjustment function has been implemented in the PHY. The entire training processes only need to modify the register to start and complete enough.

The entire training process is as follows:

1. PHY's register is reset, the setup is complete.
2. Controller to send the initial command and to complete initialization.
3. Set the PHY's register beginning calibration.

Offset	Bit	Default	Description
0x8	7~2	0x0	Others register
	1	0x0	set calibration bypass mode(1:bypass mode; 0:nomal)
	0	0x0	set calibration start (1: start; 0: stop)

4. After setting register in 5us complete calibration, see the calibration register can be read at this time is complete.

5. Normal read and writes operation can begin.

The software training process is as follows:

1. PHY's register is reset, the setup is complete.
2. Controller to send the initial command and to complete initialization.
3. Set the PHY's register calibration bypass mode.
4. Send 4 consecutive burst8/4 read command.
5. Read register address 0x3c4 value idqs.
6. According to the state diagram shown below to change the cycle ophse dll value.
7. Repeat 4, 5, 6 until you have completed.

Rockchip Confidential

Chapter 7 Nand Flash Controller (NandC)

7.1 Overview

Nand Flash Controller (NandC) is used to control data transmission from host to flash device or from flash device to host. NandC is connected to AHB BUS through an AHB Master and an AHB Slave. The data transmission between host and external memory can be done through AHB Master interface or AHB Slave interface.

NandC supports the following features:

- Software Interface Type
 - Support directly mode
 - Support LLP(Linked List Pointer) mode
- Flash Interface Type
 - Support Asynchronous Flash Interface with 8bits data width ("Asyn8x" for short)
 - Support 1 flash devices
- Flash Type
 - Support Managed NAND Flash(LBA) and Raw NAND Flash(NO-LBA)
 - Support SLC/MLC/TLC Flash
- Flash Interface Timing
 - Asyn8x: configurable timing, one byte per two host clocks at the fastest speed
- Randomizer Ability
 - Support two randomizer mode with different polynomial
 - Support two randomizer width, 8bit and 16bit parallel
- BCH/ECC Ability
 - 16bit/1KB BCH/ECC: support 16bitBCH/ECC, which can detect and correct up to 16 error bits in every 1K bytes data
 - 24bit/1KB BCH/ECC: support 24bitBCH/ECC, which can detect and correct up to 24 error bits in every 1K bytes data
 - 40bit/1KB BCH/ECC: support 40bitBCH/ECC, which can detect and correct up to 40 error bits in every 1K bytes data
 - 60bit/1KB BCH/ECC: support 60bitBCH/ECC, which can detect and correct up to 60 error bits in every 1K bytes data
 - 8bit/512B BCH/ECC: support 8bitBCH/ECC, which can detect and correct up to 8 error bits in every 512 bytes data
 - 12bit/512B BCH/ECC: support 12bitBCH/ECC, which can detect and correct up to 12 error bits in every 512 bytes data
 - 20bit/512B BCH/ECC: support 20bitBCH/ECC, which can detect and correct up to 20 error bits in every 512 bytes data
 - 30bit/512B BCH/ECC: support 30bitBCH/ECC, which can detect and correct up to 30 error bits in every 512 bytes data
 - 16bit/512B BCH/ECC: support 16bitBCH/ECC, which can detect and correct up to 16 error bits in every 512 bytes data
 - 24bit/512B BCH/ECC: support 24bitBCH/ECC, which can detect and correct up to 24 error bits in every 512 bytes data
 - 40bit/512B BCH/ECC: support 40bitBCH/ECC, which can detect and correct up to 40 error bits in every 512 bytes data
 - 60bit/512B BCH/ECC: support 60bitBCH/ECC, which can detect and correct up to 60 error bits in every 512 bytes data
- Transmission Ability
 - Support 16K bytes data transmission at a time at most
 - Support two transfer working modes: Bypass or DMA
 - Support two transfer codeword size for Managed NAND Flash: 1024 bytes/codeword or 512 bytes/codeword
- Internal Memory
 - 2 built-in SRAMs, and the size is 1k bytes respectively
 - Can be accessed by other masters
 - Can be operated in pingpong mode by other masters

7.2 Block Diagram

NandC comprises with:

- MIF: AHB Master Interface
- SIF : AHB Slave Interface
- SRIF : SRAM Interface
- TRANSC : Transfer Controller
- LLPC : LLP Controller
- BCHENC : BCH Encoder
- BCHDEC : BCH Decoder
- RANDMZ : Randomizer
- FIF_GEN : Flash Interface Generation
- DLC : Delay Line Controller

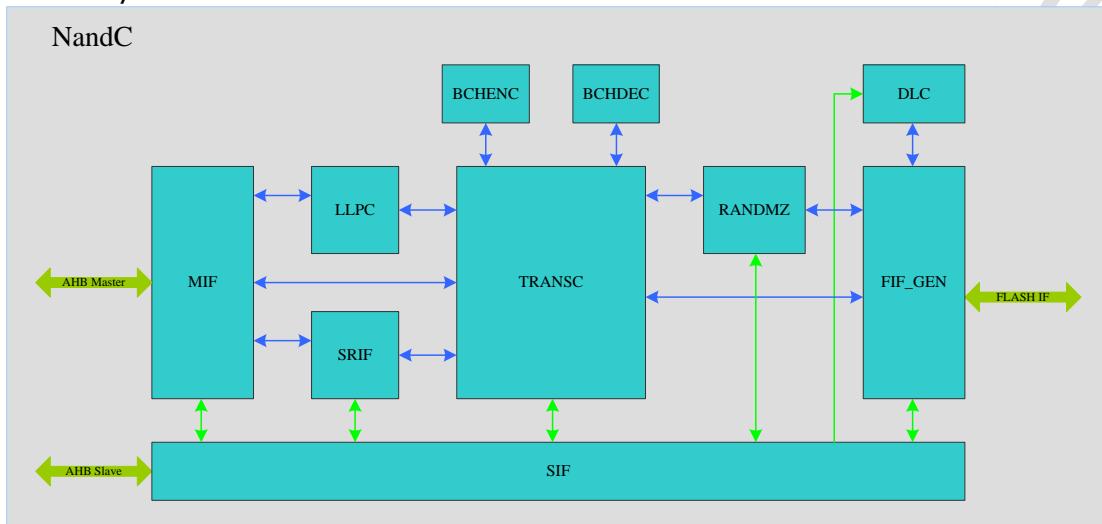


Fig. 7-1 NandC Block Diagram

7.3 Function Description

7.3.1 AHB Interface

There is an AHB master interface in NandC, which is selectable and configurable. It is responsible for transferring data from external memory to internal memory when flash program, or inverse when flash read; and transferring LLP data from external memory to internal register file when LLP is active.

There is an AHB slave interface in NandC. It is responsible for accessing registers and internal memories. The addresses of these registers and memories are listed in Register Description section.

7.3.2 Flash Type/Flash Interface

Only asynchronous 8bits flash interface is supported in this design. You can use it by software (configure FMCTL) to suit for these devices. Also you can configure its timing parameters by software (configure FMWAIT_ASYN) to have your desired rate.

7.3.3 Linked List Pointer Mode (LLP)

To save the software resource and improve the performance, a LLP is add, which is selectable. When LLP is selected, the flash operation instructions stored in external memory with specific format should be loaded for flash working. The detailed format and working flow are referred to LLP Application section.

7.3.4 BCH Encoder/BCH Decoder

The BCH Encoder is responsible for encoding data to be written into flash device. The max encoded length is 1133bytes,in which the data length is 1024bytes, system information is 4bytes, BCH code is 105bytes.

The BCH Decoder is responsible for decoding data read from flash device. The max decoded

length is 1133bytes, in which the data length is 1024bytes, spare length is 109bytes.

7.3.5 Randomizer

To improve device lifetime, a randomizer is added in NandC. It includes two parts: Scrambler and Descrambler, which is responsible for scrambling data to be written into flash after bch encoding, and descrambling data read from flash before bch decoding.

7.4 Register Description

7.4.1 Internal Address Mapping

Slave address can be divided into different length for different usage, which is shown as follows.

Table 7-1 NandC Address Mapping

Base Address[12:8]	Device	Address Length	Offset Address Range
5'b00_00x(x=0, 1)	FLR	512 BYTE	0x0000 ~ 0x01ff
5'b00_01x(x=0, 1)	SPR	512 BYTE	0x0200 ~ 0x03ff
5'b00_10x(x=0, 1)	FLR1	512 BYTE	0x0400 ~ 0x05ff
5'b01_000	Flash0	256 BYTE	0x0800 ~ 0x08ff
5'b01_001	Flash1	256 BYTE	0x0900 ~ 0x09ff
5'b01_010	Flash2	256 BYTE	0x0a00 ~ 0x0aff
5'b01_011	Flash3	256 BYTE	0x0b00 ~ 0x0bff
5'b01_100	Flash4	256 BYTE	0x0c00 ~ 0x0cff
5'b01_101	Flash5	256 BYTE	0x0d00 ~ 0x0dff
5'b01_110	Flash6	256 BYTE	0x0e00 ~ 0x0eff
5'b01_111	Flash7	256 BYTE	0x0f00 ~ 0x0fff
5'b10_0xx(x=0, 1)	Sram0	1K BYTE	0x1000 ~ 0x13ff
5'b10_1xx(x=0, 1)	Sram1	1K BYTE	0x1400 ~ 0x17ff

7.4.2 Registers Summary

Name	Offset	Size	Reset Value	Description
NANDC_FMCTL	0x0000	W	0x000000200	Flash interface control register
NANDC_FMWAIT_ASYN	0x0004	W	0x3f03f7ff	Flash timing control register for asynchronous timing
NANDC_FLCTL	0x0008	W	0x00100000	Internal transfer control register
NANDC_BCHCTL	0x000c	W	0x00000008	BCH control register
NANDC_MTRANS_CFG	0x0010	W	0x000001d0	Bus transfer configuration register
NANDC_MTRANS_SADDR0	0x0014	W	0x000000000	Start address register for page data transfer
NANDC_MTRANS_SADDR1	0x0018	W	0x000000000	Start address register for spare data transfer
NANDC_MTRANS_STAT	0x001c	W	0x000000000	Bus transfer status register
NANDC_BCHST0	0x0020	W	0x040000000	BCH status register for codeword 0~1
NANDC_BCHST1	0x0024	W	0x000000000	BCH status register for codeword 2~3
NANDC_BCHST2	0x0028	W	0x000000000	BCH status register for codeword 4~5
NANDC_BCHST3	0x002c	W	0x000000000	BCH status register for codeword 6~7

Name	Offset	Size	Reset Value	Description
NANDC_BCHST4	0x0030	W	0x00000000	BCH status register for codeword 8~9
NANDC_BCHST5	0x0034	W	0x00000000	BCH status register for codeword 10~11
NANDC_BCHST6	0x0038	W	0x00000000	BCH status register for codeword 12~13
NANDC_BCHST7	0x003c	W	0x00000000	BCH status register for codeword 14~15
NANDC_BCHLOC0	0x0040	W	0x00000000	BCH error bit location number register for codeword 0~5
NANDC_BCHLOC1	0x0044	W	0x00000000	BCH error bit location number register for codeword 6~11
NANDC_BCHLOC2	0x0048	W	0x00000000	BCH error bit location number register for codeword 12~15
NANDC_BCHLOC3	0x004c	W	0x00000000	Highest bit for BCH error bit location number register
NANDC_BCHDE0_0	0x0070	W	0x00000000	BCH decode result of 0th error bit for codeword 0
NANDC_BCHDE0_1	0x0074	W	0x00000000	BCH decode result of 1th error bit for codeword 0
NANDC_BCHDE0_2	0x0078	W	0x00000000	BCH decode result of 2th error bit for codeword 0
NANDC_BCHDE0_3	0x007c	W	0x00000000	BCH decode result of 3th error bit for codeword 0
NANDC_BCHDE0_4	0x0080	W	0x00000000	BCH decode result of 4th error bit for codeword 0
NANDC_BCHDE0_5	0x0084	W	0x00000000	BCH decode result of 5th error bit for codeword 0
NANDC_BCHDE0_6	0x0088	W	0x00000000	BCH decode result of 6th error bit for codeword 0
NANDC_BCHDE0_7	0x008c	W	0x00000000	BCH decode result of 7th error bit for codeword 0
NANDC_BCHDE0_8	0x0090	W	0x00000000	BCH decode result of 8th error bit for codeword 0
NANDC_BCHDE0_9	0x0094	W	0x00000000	BCH decode result of 9th error bit for codeword 0
NANDC_BCHDE0_10	0x0098	W	0x00000000	BCH decode result of 10th error bit for codeword 0
NANDC_BCHDE0_11	0x009c	W	0x00000000	BCH decode result of 11th error bit for codeword 0
NANDC_BCHDE0_12	0x00a0	W	0x00000000	BCH decode result of 12th error bit for codeword 0
NANDC_BCHDE0_13	0x00a4	W	0x00000000	BCH decode result of 13th error bit for codeword 0

Name	Offset	Size	Reset Value	Description
NANDC_BCHDE0_14	0x00a8	W	0x00000000	BCH decode result of 14th error bit for codeword 0
NANDC_BCHDE0_15	0x00ac	W	0x00000000	BCH decode result of 15th error bit for codeword 0
NANDC_BCHDE0_16	0x00b0	W	0x00000000	BCH decode result of 16th error bit for codeword 0
NANDC_BCHDE0_17	0x00b4	W	0x00000000	BCH decode result of 17th error bit for codeword 0
NANDC_BCHDE0_18	0x00b8	W	0x00000000	BCH decode result of 18th error bit for codeword 0
NANDC_BCHDE0_19	0x00bc	W	0x00000000	BCH decode result of 19th error bit for codeword 0
NANDC_BCHDE0_20	0x00c0	W	0x00000000	BCH decode result of 20th error bit for codeword 0
NANDC_BCHDE0_21	0x00c4	W	0x00000000	BCH decode result of 21th error bit for codeword 0
NANDC_BCHDE0_22	0x00c8	W	0x00000000	BCH decode result of 22th error bit for codeword 0
NANDC_BCHDE0_23	0x00cc	W	0x00000000	BCH decode result of 23th error bit for codeword 0
NANDC_BCHDE1_0	0x00d0	W	0x00000000	BCH decode result of 0th error bit for codeword 1
NANDC_BCHDE1_1	0x00d4	W	0x00000000	BCH decode result of 1th error bit for codeword 1
NANDC_BCHDE1_2	0x00d8	W	0x00000000	BCH decode result of 2th error bit for codeword 1
NANDC_BCHDE1_3	0x00dc	W	0x00000000	BCH decode result of 3th error bit for codeword 1
NANDC_BCHDE1_4	0x00e0	W	0x00000000	BCH decode result of 4th error bit for codeword 1
NANDC_BCHDE1_5	0x00e4	W	0x00000000	BCH decode result of 5th error bit for codeword 1
NANDC_BCHDE1_6	0x00e8	W	0x00000000	BCH decode result of 6th error bit for codeword 1
NANDC_BCHDE1_7	0x00ec	W	0x00000000	BCH decode result of 7th error bit for codeword 1
NANDC_BCHDE1_8	0x00f0	W	0x00000000	BCH decode result of 8th error bit for codeword 1
NANDC_BCHDE1_9	0x00f4	W	0x00000000	BCH decode result of 9th error bit for codeword 1
NANDC_BCHDE1_10	0x00f8	W	0x00000000	BCH decode result of 10th error bit for codeword 1
NANDC_BCHDE1_11	0x00fc	W	0x00000000	BCH decode result of 11th error bit for codeword 1

Name	Offset	Size	Reset Value	Description
NANDC_BCHDE1_12	0x0100	W	0x00000000	BCH decode result of 12th error bit for codeword 1
NANDC_BCHDE1_13	0x0104	W	0x00000000	BCH decode result of 13th error bit for codeword 1
NANDC_BCHDE1_14	0x0108	W	0x00000000	BCH decode result of 14th error bit for codeword 1
NANDC_BCHDE1_15	0x010c	W	0x00000000	BCH decode result of 15th error bit for codeword 1
NANDC_BCHDE1_16	0x0110	W	0x00000000	BCH decode result of 16th error bit for codeword 1
NANDC_BCHDE1_17	0x0114	W	0x00000000	BCH decode result of 17th error bit for codeword 1
NANDC_BCHDE1_18	0x0118	W	0x00000000	BCH decode result of 18th error bit for codeword 1
NANDC_BCHDE1_19	0x011c	W	0x00000000	BCH decode result of 19th error bit for codeword 1
NANDC_BCHDE1_20	0x0120	W	0x00000000	BCH decode result of 20th error bit for codeword 1
NANDC_BCHDE1_21	0x0124	W	0x00000000	BCH decode result of 21th error bit for codeword 1
NANDC_BCHDE1_22	0x0128	W	0x00000000	BCH decode result of 22th error bit for codeword 1
NANDC_BCHDE1_23	0x012c	W	0x00000000	BCH decode result of 23th error bit for codeword 1
NANDC_DLL_CTL_REG0	0x0130	W	0x00000000	DLL control register 0
NANDC_DLL_CTL_REG1	0x0134	W	0x00000000	DLL control register 1
NANDC_DLL_OBS_REG0	0x0138	W	0x00000002	DLL status register
NANDC_RANDMZ_CFG	0x0150	W	0x00000000	Randomizer configure register
NANDC_FMWAIT_SYN	0x0158	W	0x00000000	Flash timing control register for synchronous timing
NANDC_NANDC_VER	0x0160	W	0x56363030	NandC version register
NANDC_LLPCONTROL	0x0164	W	0x00000000	LLP control register
NANDC_LLPSSTATUS	0x0168	W	0x00000001	LLP status register
NANDC_INTEN	0x016c	W	0x00000000	Interrupt enable register
NANDC_INTCLR	0x0170	W	0x00000000	Interrupt clear register
NANDC_INTST	0x0174	W	0x00000000	Interrupt status register
NANDC_SPARE0_0	0x0200	W	0xffffffff	System information for codeword 0
NANDC_SPARE0_1	0x0204	W	0x00000000	Spare data and BCH encode information for codeword 0
NANDC_SPARE0_2	0x0208	W	0x00000000	Spare data and BCH encode information for codeword 0
NANDC_SPARE0_3	0x020c	W	0x00000000	Spare data and BCH encode information for codeword 0

Name	Offset	Size	Reset Value	Description
NANDC_SPARE0_4	0x0210	W	0x00000000	Spare data and BCH encode information for codeword 0
NANDC_SPARE0_5	0x0214	W	0x00000000	Spare data and BCH encode information for codeword 0
NANDC_SPARE0_6	0x0218	W	0x00000000	Spare data and BCH encode information for codeword 0
NANDC_SPARE0_7	0x021c	W	0x00000000	Spare data and BCH encode information for codeword 0
NANDC_SPARE0_8	0x0220	W	0x00000000	Spare data and BCH encode information for codeword 0
NANDC_SPARE0_9	0x0224	W	0x00000000	Spare data and BCH encode information for codeword 0
NANDC_SPARE0_10	0x0228	W	0x00000000	Spare data and BCH encode information for codeword 0
NANDC_SPARE0_11	0x022c	W	0x00000000	Spare data and BCH encode information for codeword 0
NANDC_SPARE1_0	0x0230	W	0xffffffff	System information for codeword 1
NANDC_SPARE1_1	0x0234	W	0x00000000	Spare data and BCH encode information for codeword 1
NANDC_SPARE1_2	0x0238	W	0x00000000	Spare data and BCH encode information for codeword 1
NANDC_SPARE1_3	0x023c	W	0x00000000	Spare data and BCH encode information for codeword 1
NANDC_SPARE1_4	0x0240	W	0x00000000	Spare data and BCH encode information for codeword 1
NANDC_SPARE1_5	0x0244	W	0x00000000	Spare data and BCH encode information for codeword 1
NANDC_SPARE1_6	0x0248	W	0x00000000	Spare data and BCH encode information for codeword 1
NANDC_SPARE1_7	0x024c	W	0x00000000	Spare data and BCH encode information for codeword 1
NANDC_SPARE1_8	0x0250	W	0x00000000	Spare data and BCH encode information for codeword 1
NANDC_SPARE1_9	0x0254	W	0x00000000	Spare data and BCH encode information for codeword 1
NANDC_SPARE1_10	0x0258	W	0x00000000	Spare data and BCH encode information for codeword 1
NANDC_SPARE1_11	0x025c	W	0x00000000	Spare data and BCH encode information for codeword 1
NANDC_SPARE0_12	0x0260	W	0x00000000	Spare data and BCH encode information for codeword 0
NANDC_SPARE0_13	0x0264	W	0x00000000	Spare data and BCH encode information for codeword 0
NANDC_SPARE0_14	0x0268	W	0x00000000	Spare data and BCH encode information for codeword 0

Name	Offset	Size	Reset Value	Description
NANDC_SPARE0_15	0x026c	W	0x00000000	Spare data and BCH encode information for codeword 0
NANDC_SPARE0_16	0x0270	W	0x00000000	Spare data and BCH encode information for codeword 0
NANDC_SPARE0_17	0x0274	W	0x00000000	Spare data and BCH encode information for codeword 0
NANDC_SPARE0_18	0x0278	W	0x00000000	Spare data and BCH encode information for codeword 0
NANDC_SPARE0_19	0x027c	W	0x00000000	Spare data and BCH encode information for codeword 0
NANDC_SPARE0_20	0x0280	W	0x00000000	Spare data and BCH encode information for codeword 0
NANDC_SPARE0_21	0x0284	W	0x00000000	Spare data and BCH encode information for codeword 0
NANDC_SPARE0_22	0x0288	W	0x00000000	Spare data and BCH encode information for codeword 0
NANDC_SPARE0_23	0x028c	W	0x00000000	Spare data and BCH encode information for codeword 0
NANDC_SPARE0_24	0x0290	W	0x00000000	Spare data and BCH encode information for codeword 0
NANDC_SPARE0_25	0x0294	W	0x00000000	Spare data and BCH encode information for codeword 0
NANDC_SPARE0_26	0x0298	W	0x00000000	Spare data and BCH encode information for codeword 0
NANDC_SPARE0_27	0x029c	W	0x00000000	Spare data and BCH encode information for codeword 0
NANDC_SPARE1_12	0x02a0	W	0x00000000	Spare data and BCH encode information for codeword 1
NANDC_SPARE1_13	0x02a4	W	0x00000000	Spare data and BCH encode information for codeword 1
NANDC_SPARE1_14	0x02a8	W	0x00000000	Spare data and BCH encode information for codeword 1
NANDC_SPARE1_15	0x02ac	W	0x00000000	Spare data and BCH encode information for codeword 1
NANDC_SPARE1_16	0x02b0	W	0x00000000	Spare data and BCH encode information for codeword 1
NANDC_SPARE1_17	0x02b4	W	0x00000000	Spare data and BCH encode information for codeword 1
NANDC_SPARE1_18	0x02b8	W	0x00000000	Spare data and BCH encode information for codeword 1
NANDC_SPARE1_19	0x02bc	W	0x00000000	Spare data and BCH encode information for codeword 1
NANDC_SPARE1_20	0x02c0	W	0x00000000	Spare data and BCH encode information for codeword 1

Name	Offset	Size	Reset Value	Description
NANDC_SPARE1_21	0x02c4	W	0x00000000	Spare data and BCH encode information for codeword 1
NANDC_SPARE1_22	0x02c8	W	0x00000000	Spare data and BCH encode information for codeword 1
NANDC_SPARE1_23	0x02cc	W	0x00000000	Spare data and BCH encode information for codeword 1
NANDC_SPARE1_24	0x02d0	W	0x00000000	Spare data and BCH encode information for codeword 1
NANDC_SPARE1_25	0x02d4	W	0x00000000	Spare data and BCH encode information for codeword 1
NANDC_SPARE1_26	0x02d8	W	0x00000000	Spare data and BCH encode information for codeword 1
NANDC_SPARE1_27	0x02dc	W	0x00000000	Spare data and BCH encode information for codeword 1
NANDC_BCHDE0_24	0x0400	W	0x00000000	BCH decode result of 24th error bit for codeword 0
NANDC_BCHDE0_25	0x0404	W	0x00000000	BCH decode result of 25th error bit for codeword 0
NANDC_BCHDE0_26	0x0408	W	0x00000000	BCH decode result of 26th error bit for codeword 0
NANDC_BCHDE0_27	0x040c	W	0x00000000	BCH decode result of 27th error bit for codeword 0
NANDC_BCHDE0_28	0x0410	W	0x00000000	BCH decode result of 28th error bit for codeword 0
NANDC_BCHDE0_29	0x0414	W	0x00000000	BCH decode result of 29th error bit for codeword 0
NANDC_BCHDE0_30	0x0418	W	0x00000000	BCH decode result of 30th error bit for codeword 0
NANDC_BCHDE0_31	0x041c	W	0x00000000	BCH decode result of 31th error bit for codeword 0
NANDC_BCHDE0_32	0x0420	W	0x00000000	BCH decode result of 32th error bit for codeword 0
NANDC_BCHDE0_33	0x0424	W	0x00000000	BCH decode result of 33th error bit for codeword 0
NANDC_BCHDE0_34	0x0428	W	0x00000000	BCH decode result of 34th error bit for codeword 0
NANDC_BCHDE0_35	0x042c	W	0x00000000	BCH decode result of 35th error bit for codeword 0
NANDC_BCHDE0_36	0x0430	W	0x00000000	BCH decode result of 36th error bit for codeword 0
NANDC_BCHDE0_37	0x0434	W	0x00000000	BCH decode result of 37th error bit for codeword 0
NANDC_BCHDE0_38	0x0438	W	0x00000000	BCH decode result of 38th error bit for codeword 0

Name	Offset	Size	Reset Value	Description
NANDC_BCHDE0_39	0x043c	W	0x00000000	BCH decode result of 39th error bit for codeword 0
NANDC_BCHDE0_40	0x0440	W	0x00000000	BCH decode result of 40th error bit for codeword 0
NANDC_BCHDE0_41	0x0444	W	0x00000000	BCH decode result of 41th error bit for codeword 0
NANDC_BCHDE0_42	0x0448	W	0x00000000	BCH decode result of 42th error bit for codeword 0
NANDC_BCHDE0_43	0x044c	W	0x00000000	BCH decode result of 43th error bit for codeword 0
NANDC_BCHDE0_44	0x0450	W	0x00000000	BCH decode result of 44th error bit for codeword 0
NANDC_BCHDE0_45	0x0454	W	0x00000000	BCH decode result of 45th error bit for codeword 0
NANDC_BCHDE0_46	0x0458	W	0x00000000	BCH decode result of 46th error bit for codeword 0
NANDC_BCHDE0_47	0x045c	W	0x00000000	BCH decode result of 47th error bit for codeword 0
NANDC_BCHDE0_48	0x0460	W	0x00000000	BCH decode result of 48th error bit for codeword 0
NANDC_BCHDE0_49	0x0464	W	0x00000000	BCH decode result of 49th error bit for codeword 0
NANDC_BCHDE0_50	0x0468	W	0x00000000	BCH decode result of 50th error bit for codeword 0
NANDC_BCHDE0_51	0x046c	W	0x00000000	BCH decode result of 51th error bit for codeword 0
NANDC_BCHDE0_52	0x0470	W	0x00000000	BCH decode result of 52th error bit for codeword 0
NANDC_BCHDE0_53	0x0474	W	0x00000000	BCH decode result of 53th error bit for codeword 0
NANDC_BCHDE0_54	0x0478	W	0x00000000	BCH decode result of 54th error bit for codeword 0
NANDC_BCHDE0_55	0x047c	W	0x00000000	BCH decode result of 55th error bit for codeword 0
NANDC_BCHDE0_56	0x0480	W	0x00000000	BCH decode result of 56th error bit for codeword 0
NANDC_BCHDE0_57	0x0484	W	0x00000000	BCH decode result of 57th error bit for codeword 0
NANDC_BCHDE0_58	0x0488	W	0x00000000	BCH decode result of 58th error bit for codeword 0
NANDC_BCHDE0_59	0x048c	W	0x00000000	BCH decode result of 59th error bit for codeword 0
NANDC_BCHDE1_24	0x0490	W	0x00000000	BCH decode result of 24th error bit for codeword 1

Name	Offset	Size	Reset Value	Description
NANDC_BCHDE1_25	0x0494	W	0x00000000	BCH decode result of 25th error bit for codeword 1
NANDC_BCHDE1_26	0x0498	W	0x00000000	BCH decode result of 26th error bit for codeword 1
NANDC_BCHDE1_27	0x049c	W	0x00000000	BCH decode result of 27th error bit for codeword 1
NANDC_BCHDE1_28	0x04a0	W	0x00000000	BCH decode result of 28th error bit for codeword 1
NANDC_BCHDE1_29	0x04a4	W	0x00000000	BCH decode result of 29th error bit for codeword 1
NANDC_BCHDE1_30	0x04a8	W	0x00000000	BCH decode result of 30th error bit for codeword 1
NANDC_BCHDE1_31	0x04ac	W	0x00000000	BCH decode result of 31th error bit for codeword 1
NANDC_BCHDE1_32	0x04b0	W	0x00000000	BCH decode result of 32th error bit for codeword 1
NANDC_BCHDE1_33	0x04b4	W	0x00000000	BCH decode result of 33th error bit for codeword 1
NANDC_BCHDE1_34	0x04b8	W	0x00000000	BCH decode result of 34th error bit for codeword 1
NANDC_BCHDE1_35	0x04bc	W	0x00000000	BCH decode result of 35th error bit for codeword 1
NANDC_BCHDE1_36	0x04c0	W	0x00000000	BCH decode result of 36th error bit for codeword 1
NANDC_BCHDE1_37	0x04c4	W	0x00000000	BCH decode result of 37th error bit for codeword 1
NANDC_BCHDE1_38	0x04c8	W	0x00000000	BCH decode result of 38th error bit for codeword 1
NANDC_BCHDE1_39	0x04cc	W	0x00000000	BCH decode result of 39th error bit for codeword 1
NANDC_BCHDE1_40	0x04d0	W	0x00000000	BCH decode result of 40th error bit for codeword 1
NANDC_BCHDE1_41	0x04d4	W	0x00000000	BCH decode result of 41th error bit for codeword 1
NANDC_BCHDE1_42	0x04d8	W	0x00000000	BCH decode result of 42th error bit for codeword 1
NANDC_BCHDE1_43	0x04dc	W	0x00000000	BCH decode result of 43th error bit for codeword 1
NANDC_BCHDE1_44	0x04e0	W	0x00000000	BCH decode result of 44th error bit for codeword 1
NANDC_BCHDE1_45	0x04e4	W	0x00000000	BCH decode result of 45th error bit for codeword 1
NANDC_BCHDE1_46	0x04e8	W	0x00000000	BCH decode result of 46th error bit for codeword 1

Name	Offset	Size	Reset Value	Description
NANDC_BCHDE1_47	0x04ec	W	0x00000000	BCH decode result of 47th error bit for codeword 1
NANDC_BCHDE1_48	0x04f0	W	0x00000000	BCH decode result of 48th error bit for codeword 1
NANDC_BCHDE1_49	0x04f4	W	0x00000000	BCH decode result of 49th error bit for codeword 1
NANDC_BCHDE1_50	0x04f8	W	0x00000000	BCH decode result of 50th error bit for codeword 1
NANDC_BCHDE1_51	0x04fc	W	0x00000000	BCH decode result of 51th error bit for codeword 1
NANDC_BCHDE1_52	0x0500	W	0x00000000	BCH decode result of 52th error bit for codeword 1
NANDC_BCHDE1_53	0x0504	W	0x00000000	BCH decode result of 53th error bit for codeword 1
NANDC_BCHDE1_54	0x0508	W	0x00000000	BCH decode result of 54th error bit for codeword 1
NANDC_BCHDE1_55	0x050c	W	0x00000000	BCH decode result of 55th error bit for codeword 1
NANDC_BCHDE1_56	0x0510	W	0x00000000	BCH decode result of 56th error bit for codeword 1
NANDC_BCHDE1_57	0x0514	W	0x00000000	BCH decode result of 57th error bit for codeword 1
NANDC_BCHDE1_58	0x0518	W	0x00000000	BCH decode result of 58th error bit for codeword 1
NANDC_BCHDE1_59	0x051c	W	0x00000000	BCH decode result of 59th error bit for codeword 1

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

7.4.3 Detail Register Description

NANDC_FMCTL

Address: Operational Base + offset (0x0000)

Flash interface control register

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15	RW	0x0	<p>syn_mode Toggle enable signal, 1 active. 1'b0: ONFI synchronous flash 1'b1: Toggle synchronous flash</p>
14	RW	0x0	<p>syn_ciken Synchronous flash clock enable signal, 1 active. Only available in Synchronous Mode. 1'b0: flash clock is disabled 1'b1: flash clock is enabled</p>

Bit	Attr	Reset Value	Description
13	RW	0x0	tm Timing mode indication. 1'b0: Asynchronous Mode 1'b1: Synchronous Mode (Toggle or ONFI Synchronous)
12	RW	0x0	dwidth Flash data bus width indication. 1'b0: 8bits, active in both Asynchronous Mode flash and Synchronous Mode flash 1'b1: 16bits, active only in Asynchronous Mode flash
11:10	RO	0x0	reserved
9	RO	0x1	frdy Flash ready/busy indicate signal. 1'b0: flash is busy 1'b1: flash is ready This bit is the sample of the pin of R/Bn.
8	RW	0x0	wp Flash write protect. 1'b0: flash program/erase disabled. 1'b1: flash program/erase enabled. This bit is output to the pin of WPn.
7	RW	0x0	fcs7 Flash memory chip 7 select control. 1'b1: hold flash memory chip select activity 1'b0: flash memory chip select activity free
6	RW	0x0	fcs6 Flash memory chip 6 select control. 1'b1: hold flash memory chip select activity. 1'b0: flash memory chip select activity free.
5	RW	0x0	fcs5 Flash memory chip 5 select control. 1'b1: hold flash memory chip select activity 1'b0: flash memory chip select activity free
4	RW	0x0	fcs4 Flash memory chip 4 select control. 1'b1: hold flash memory chip select activity 1'b0: flash memory chip select activity free
3	RW	0x0	fcs3 Flash memory chip 3 select control. 1'b1: hold flash memory chip select activity 1'b0: flash memory chip select activity free
2	RW	0x0	fcs2 Flash memory chip 2 select control. 1'b1: hold flash memory chip select activity 1'b0: flash memory chip select activity free

Bit	Attr	Reset Value	Description
1	RW	0x0	fcs1 Flash memory chip 1 select control. 1'b1: hold flash memory chip select activity 1'b0: flash memory chip select activity free
0	RW	0x0	fcs0 Flash memory chip 0 select control. 1'b1: hold flash memory chip select activity 1'b0: flash memory chip select activity free

NANDC_FMWAIT_ASYN

Address: Operational Base + offset (0x0004)

Flash timing control register for asynchronous timing

Bit	Attr	Reset Value	Description
31	RO	0x0	reserved
30	RW	0x0	fmw_dly_en fmw_dly enable signal,1 active
29:24	RW	0x3f	fmw_dly The number of delay cycle between two codeword transmission
23:18	RO	0x0	reserved
17:12	RW	0x3f	csr_w When in Asynchronous mode or Toggle address/command mode, this field specifies the number of processor clock cycles from the falling edge of CSn to the falling edge of RDn or WRn. The min value of csr_w is 0.
11	RW	0x0	hard_rdy Hardware handshaking controller bit. When asserted, an external device asserts signal "RDY" to extend a wait-state access and the rest bits in this register will be ignored
10:5	RW	0x3f	rwpw When in Asynchronous mode or Toggle address/command mode, this field specifies the width of RDn or WRn in processor clock cycles, $0x0 \leq rwpw \leq 0x3f$.
4:0	RW	0x1f	rwcs When in Asynchronous mode or Toggle address/command mode, this field specifies the number of processor clock cycles from the rising edge of RDn or WRn to the rising edge of CSn, $0x0 \leq rwcs \leq 0x1f$.

NANDC_FLCTL

Address: Operational Base + offset (0x0008)

Internal transfer control register

Bit	Attr	Reset Value	Description
31:27	RO	0x0	reserved

Bit	Attr	Reset Value	Description
26:22	RW	0x00	<p>page_num Transmission codeword number in internal DMA mode when bus-mode is master-mode. 1~16: 1~16 codeword default: not support Only active in internal DMA mode and when bus-mode is master-mode</p>
21	RW	0x0	<p>page_size Transmission codeword size in internal DMA mode. 1'b0: 1024bytes/codeword 1'b1: 512bytes/codeword</p>
20	RO	0x1	<p>tr_rdy Internal DMA transmission ready indication. 1'b0: internal DMA transmission is busy 1'b1: internal DMA transmission is ready When reading flash, tr_rdy should not be set to 1 until all data transmission and correct finished. When programming flash, tr_rdy should not be set to 1 until all data transmission finished. Only active in internal DMA mode.</p>
19	RO	0x0	reserved
18:12	RW	0x00	<p>spare_size Spare byte number when lba_en=1. 0<= spare_size<=109. When spare_size>=109, it is treated as 0. The spare_size must be even number when flash is ONFI Synchronous Flash or Asynchronous Flash with 16bits data width.</p>
11	RW	0x0	<p>lba_en LBA mode indication, 1 active. 1'b0: NO-LBA mode, NandC should transfer both page data and spare data in every codeword 1'b1: LBA mode, NandC should transfer both page data and spare data in every codeword When lba_en is active, BCH CODEC should be disabled, spare_size and page_size are configurable. When lba_en is active, cor_able is inactive.</p>
10	RW	0x0	<p>cor_able Auto correct enable indication, 1 active. 1'b0: auto correct disable 1'b1: auto correct enable Only active in internal DMA mode. lba_en is prior to cor_able. When lba_en=1, cor_able is ignored.</p>
9:8	RO	0x0	reserved

Bit	Attr	Reset Value	Description
7	RW	0x0	<p>flash_st_mod Mode for NandC to start internal data transmission in internal DMA mode.</p> <p>1'b0: busy mode, hardware should not start internal data transmission until flash is ready even flash_st is asserted</p> <p>1'b1: ready mode, hardware should start internal data transmission directly when flash_st is asserted</p> <p>Only active in internal DMA mode.</p>
6:5	RW	0x0	<p>tr_count Transmission codeword number in internal DMA mode when bus-mode is slave-mode.</p> <p>2'b00: 0 codeword need transferred</p> <p>2'b01: 1 codeword need transferred</p> <p>2'b10: 2 codeword need transferred</p> <p>2'b11: not supported</p> <p>Only active in internal DMA mode and when bus-mode is slave-mode.</p>
4	RW	0x0	<p>st_addr Start buffer address.</p> <p>1'b0: start transfer from sram0</p> <p>1'b1: start transfer from sram1</p> <p>Bear in mind only active in internal DMA mode.</p>
3	RW	0x0	<p>bypass NandC internal DMA bypass indication.</p> <p>1'b0: bypass the internal DMA, data are transferred to/from flash by direct path</p> <p>1'b1: internal DMA active, data are transferred to/from flash by internal DMA</p>
2	RW	0x0	<p>flash_st Start signal for NandC to transfer data between flash and internal buffer in internal DMA mode. When asserted, it will auto cleared.</p> <p>1'b0: not start transmission</p> <p>1'b1: start transmission</p> <p>Bear in mind that it is only active in internal DMA mode</p>
1	RW	0x0	<p>flash_rdn Indicate data flow direction.</p> <p>1'b0: NandC read data from flash</p> <p>1'b1: NandC write data to flash</p>
0	RW	0x0	<p>flash_RST NandC software reset indication. When asserted, it will auto cleared.</p> <p>1'b0: not software reset</p> <p>1'b1: software reset</p> <p>Bear in mind that flash_RST is prior to flash_st</p>

NANDC_BCHCTL

Address: Operational Base + offset (0x000c)

BCH control register

Bit	Attr	Reset Value	Description
31:27	RO	0x0	reserved
26:19	RW	0x00	bchthres BCH error number threshold
18	RW	0x0	bchmode1 BCH mode selection bit 1. BchMode=bchmode1, bchmode0: 2'b00: 16bitBCH 2'b01: 24bitBCH 2'b10: 40bitBCH 2'b11: 60bitBCH
17	RO	0x0	reserved
16	RW	0x0	bchpage The data size indication when BCH is active. 1'b0: 1024 bytes, all the 1024 bytes data in codeword are valid data to be transferred 1'b1: 512 bytes, 1th~512th bytes in codeword are valid data to be transferred, and 513th~1024th bytes in codeword are invalid data stuffed with 0xff Only active when data transferred in internal DMA mode and for asynchronous flash.
15:8	RW	0x00	addr BCH active range selection. BCH should be active when access in range address.
7:5	RW	0x0	region BCH active region selection indication. 3'b000: Flash memory 0 region (flash 0) 3'b001: Flash memory 1 region (flash 1) 3'b010: Flash memory 2 region (flash 2) 3'b011: Flash memory 3 region (flash 3) 3'b100: Flash memory 4 region (flash 4) 3'b101: Flash memory 5 region (flash 5) 3'b110: Flash memory 6 region (flash 6) 3'b111: Flash memory 7 region (flash 7)
4	RW	0x0	bchmode0 BCH mode selection bit 0.
3	RW	0x1	bchepd BCH encoder/decoder power down indication. 1'b0: BCH encoder/decoder working 1'b1: BCH encoder/decoder not working

Bit	Attr	Reset Value	Description
2	RW	0x0	mode_addrctrl BCH address care mode selection indication. 1'b0: address care 1'b1: address not care This bit is just active for data transmission in bypass mode, but not for command and address transmission.
1	RO	0x0	reserved
0	RW	0x0	bchrst BCH software reset indication, When asserted, it will auto cleared. 1'b0: not software reset 1'b1: software reset BCH Decoder should be software reset before decode begin. BCH software reset should be used with nandc software reset at the same time.

NANDC_MTRANS_CFG

Address: Operational Base + offset (0x0010)

Bus transfer configuration register

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15	W1C	0x0	ahb_rst ahb master interface software reset, auto cleared
14	RW	0x0	fl_pwd Flash power down indication, 1 active. 1'b0: Flash power on, data transferred through master interface is data that to be written into or read from flash 1'b1: Flash power down, data transferred through master interface is not data that to be written into or read from flash. NandC is just used as DMA for external memory and internal memory.
13:9	RW	0x00	incr_num AHB Master incr num indication. When burst=001, software should configure incr_num. Only active for master-mode.
8:6	RW	0x7	burst AHB Master burst type indication: 3'b000: Single transfer 3'b011: 4-beat burst 3'b101: 8-beat Burst 3'b111: 16-beat burst default: not supported Only active for master-mode.

Bit	Attr	Reset Value	Description
5:3	RW	0x2	hsize AHB Master data size indication: 3'b000: 8 bits 3'b001: 16 bits 3'b010: 32 bits default: not supported Only active for master-mode.
2	RW	0x0	bus_mode Bus interface selection. 1'b0: Slave interface, flash data is transferred through slave interface 1'b1: Master interface, flash data is transferred through master interface
1	RW	0x0	ahb_wr Data transfer direction through master interface. 1'b0: read direction(internal memory ->external memory) 1'b1: write direction (internal memory->external memory) Only active for master-mode. When read flash(flash_rdn=0), ahb_wr=1; when program flash(flash_rdn=1), ahb_wr=0.
0	W1C	0x0	ahb_wr_st Start indication for loading data from external memory to internal memory or storing data from internal memory to external memory through master. When asserted, it will auto cleared. Notes: a. Only active for master-mode and fl_pwd=1. b. When fl_pwd=0, flash is active, NandC start to transfer data through master interface if flash_st=1 c. When fl_pwd=1, flash is not active, NandC start to transfer data through master interface if ahb_wr_st=1

NANDC_MTRANS_SADDR0

Address: Operational Base + offset (0x0014)

Start address register for page data transfer

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	saddr0 Start address for page data transmission. Only active for master-mode, should be word aligned.

NANDC_MTRANS_SADDR1

Address: Operational Base + offset (0x0018)

Start address register for spare data transfer

Bit	Attr	Reset Value	Description
-----	------	-------------	-------------

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	saddr1 Start address for spare data. Only active for master-mode, and should be word aligned.

NANDC_MTRANS_STAT

Address: Operational Base + offset (0x001c)

Bus transfer status register

Bit	Attr	Reset Value	Description
31:21	RO	0x0	reserved
20:16	RO	0x00	mtrans_cnt finished counter for codeword transmission through Master interface. Only active for master-mode.
15:0	RO	0x0000	bus_err Bus error indication for codeword0~15. [0] : bus error for codeword 0 [15] : bus error for codeword 15 Only active for master-mode.

NANDC_BCHST0

Address: Operational Base + offset (0x0020)

BCH status register for codeword 0~1

Bit	Attr	Reset Value	Description
31	RO	0x0	reserved
30	RO	0x0	err_hnum1_h1 Highest bit of err_hnum1.
29	RO	0x0	err_tnum1_h1 Highest bit of err_tnum1.
28	RO	0x0	err_hnum0_h1 Highest bit of err_hnum0.
27	RO	0x0	err_tnum0_h1 Highest bit of err_tnum0.
26	RO	0x1	bchrdy Ready indication for bch encoder/decoder, 1 active. 1'b0: bch encoder/decoder is busy 1'b1: bch encoder/decoder is ready
25:21	RO	0x00	err_hnum1_l5 Lower 5 bits of number of error bits found in first 512bytes of 1st backup codeword.
20:16	RO	0x00	err_tnum1_l5 Lower 5 bits of number of error bits found in 1st backup codeword.

Bit	Attr	Reset Value	Description
15	RO	0x0	fail1 Indication for the 1st backup codeword decoded failed or not. 1'b0: decode successfully 1'b1: decode fail
14	RO	0x0	done1 Indication for finishing decoding the 1st backup codeword. 1'b0: not finished 1'b1: finished
13	RO	0x0	errf1 Indication for error found in 1st backup codeword. 1'b0: no error 1'b1: error found
12:8	RO	0x00	err_hnum0_l5 Lower 5 bits of number of error bits found in first 512bytes of current backup codeword.
7:3	RO	0x00	err_tnum0_l5 Lower 5 bits of number of error bits found in current backup codeword.
2	RO	0x0	fail0 Indication for current backup codeword decode failed or not. 1'b0: decode successfully 1'b1: decode fail
1	RO	0x0	done0 Indication for finishing decoding the current backup codeword. 1'b0: not finished 1'b1: finished
0	RO	0x0	errf0 Indication for error found in current backup codeword. 1'b0: no error 1'b1: error found

NANDC_BCHST1

Address: Operational Base + offset (0x0024)

BCH status register for codeword 2~3

Bit	Attr	Reset Value	Description
31	RO	0x0	reserved
30	RO	0x0	err_hnum3_h1 Highest bit of err_hnum3.
29	RO	0x0	err_tnum3_h1 Highest bit of err_tnum3.
28	RO	0x0	err_hnum2_h1 Highest bit of err_hnum2.
27	RO	0x0	err_tnum2_h1 Highest bit of err_tnum2.
26	RO	0x0	reserved

Bit	Attr	Reset Value	Description
25:21	RO	0x00	err_hnum3_l5 Lower 5 bits of number of error bits found in first 512bytes of 3th backup codeword.
20:16	RO	0x00	err_tnum3_l5 Lower 5 bits of number of error bits found in 3th backup codeword.
15	RO	0x0	fail3 Indication for the 3th backup codeword decoded failed or not. 1'b0: decode successfully 1'b1: decode fail
14	RO	0x0	done3 Indication for finishing decoding the 3th backup codeword. 1'b0: not finished 1'b1: finished
13	RO	0x0	errf3 Indication for error found in 3th backup codeword. 1'b0: no error 1'b1: error found
12:8	RO	0x00	err_hnum2_l5 Lower 5 bits of number of error bits found in first 512bytes of 2th backup codeword.
7:3	RO	0x00	err_tnum2_l5 Lower 5 bits of number of error bits found in 2th backup codeword.
2	RO	0x0	fail2 Indication for 2th backup codeword decode failed or not. 1'b0: decode successfully 1'b1: decode fail
1	RO	0x0	done2 Indication for finishing decoding the 2th backup codeword. 1'b0: not finished 1'b1: finished
0	RO	0x0	errf2 Indication for error found in 2th backup codeword. 1'b0: no error 1'b1: error found

NANDC_BCHST2

Address: Operational Base + offset (0x0028)

BCH status register for codeword 4~5

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	bchst_cwd4_cwd5 BCHST information for 4th and 5th backup codeword. Bit assignment is similar to BCHST1 register. For more description, please refer to BCHST1 register.

NANDC_BCHST3

Address: Operational Base + offset (0x002c)

BCH status register for codeword 6~7

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	bchst_cwd6_cwd7 BCHST information for 6th and 7th backup codeword. Bit assignment is similar to BCHST1 register. For more description, please refer to BCHST1 register.

NANDC_BCHST4

Address: Operational Base + offset (0x0030)

BCH status register for codeword 8~9

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	bchst_cwd8_cwd9 BCHST information for 8th and 9th backup codeword. Bit assignment is similar to BCHST1 register. For more description, please refer to BCHST1 register.

NANDC_BCHST5

Address: Operational Base + offset (0x0034)

BCH status register for codeword 10~11

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	bchst_cwd10_cwd11 BCHST information for 10th and 11th backup codeword. Bit assignment is similar to BCHST1 register. For more description, please refer to BCHST1 register.

NANDC_BCHST6

Address: Operational Base + offset (0x0038)

BCH status register for codeword 12~13

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	bchst_cwd12_cwd13 BCHST information for 12th and 13th backup codeword. Bit assignment is similar to BCHST1 register. For more description, please refer to BCHST1 register.

NANDC_BCHST7

Address: Operational Base + offset (0x003c)

BCH status register for codeword 14~15

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	bchst_cwd14_cwd15 BCHST information for 14th and 15th backup codeword. Bit assignment is similar to BCHST1 register. For more description, please refer to BCHST1 register.

NANDC_BCHLOCO

Address: Operational Base + offset (0x0040)

BCH error bit location number register for codeword 0~5

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved
29:25	RO	0x00	err_loc5_I5 Lower 5 bits of number of 8bit error location in 5th backup codeword
24:20	RO	0x00	err_loc4_I5 Lower 5 bits of number of 8bit error location in 4th backup codeword
19:15	RO	0x00	err_loc3_I5 Lower 5 bits of number of 8bit error location in 3rd backup codeword
14:10	RO	0x00	err_loc2_I5 Lower 5 bits of number of 8bit error location in 2nd backup codeword
9:5	RO	0x00	err_loc1_I5 Lower 5 bits of number of 8bit error location in 1st backup codeword
4:0	RO	0x00	err_loc0_I5 Lower 5 bits of number of 8bit error location in current backup codeword

NANDC_BCHLOC1

Address: Operational Base + offset (0x0044)

BCH error bit location number register for codeword 6~11

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved
29:25	RO	0x00	err_loc11_I5 Lower 5 bits of number of 8bit error location in 11th backup codeword
24:20	RO	0x00	err_loc10_I5 Lower 5 bits of number of 8bit error location in 10th backup codeword
19:15	RO	0x00	err_loc9_I5 Lower 5 bits of number of 8bit error location in 9th backup codeword
14:10	RO	0x00	err_loc8_I5 Lower 5 bits of number of 8bit error location in 8th backup codeword
9:5	RO	0x00	err_loc7_I5 Lower 5 bits of number of 8bit error location in 7th backup codeword
4:0	RO	0x00	err_loc6_I5 Lower 5 bits of number of 8bit error location in 6th backup codeword

NANDC_BCHLOC2

Address: Operational Base + offset (0x0048)

BCH error bit location number register for codeword 12~15

Bit	Attr	Reset Value	Description
31:20	RO	0x0	reserved
19:15	RO	0x00	err_loc15_l5 Lower 5 bits of number of 8bit error location in 15th backup codeword
14:10	RO	0x00	err_loc14_l5 Lower 5 bits of number of 8bit error location in 4th backup codeword
9:5	RO	0x00	err_loc13_l5 Lower 5 bits of number of 8bit error location in 13th backup codeword
4:0	RO	0x00	err_loc12_l5 Lower 5 bits of number of 8bit error location in 12th backup codeword

NANDC_BCHLOC3

Address: Operational Base + offset (0x004c)

Highest bit for BCH error bit location number register

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15	RO	0x0	err_loc15_h1 High bit for numbers of 8bit error location in 15th codeword
14	RO	0x0	err_loc14_h1 High bit for numbers of 8bit error location in 14th codeword
13	RO	0x0	err_loc13_h1 High bit for numbers of 8bit error location in 13th codeword
12	RO	0x0	err_loc12_h1 High bit for numbers of 8bit error location in 12th codeword
11	RO	0x0	err_loc11_h1 High bit for numbers of 8bit error location in 11th codeword
10	RO	0x0	err_loc10_h1 High bit for numbers of 8bit error location in 10th codeword
9	RO	0x0	err_loc9_h1 High bit for numbers of 8bit error location in 9th codeword
8	RO	0x0	err_loc8_h1 High bit for numbers of 8bit error location in 8th codeword
7	RO	0x0	err_loc7_h1 High bit for numbers of 8bit error location in 7th codeword
6	RO	0x0	err_loc6_h1 High bit for numbers of 8bit error location in 6th codeword
5	RO	0x0	err_loc5_h1 High bit for numbers of 8bit error location in 5th codeword

Bit	Attr	Reset Value	Description
4	RO	0x0	err_loc4_h1 High bit for numbers of 8bit error location in 4th codeword
3	RO	0x0	err_loc3_h1 High bit for numbers of 8bit error location in 3th codeword
2	RO	0x0	err_loc2_h1 High bit for numbers of 8bit error location in 2th codeword
1	RO	0x0	err_loc1_h1 High bit for numbers of 8bit error location in 1th codeword
0	RO	0x0	err_loc0_h1 High bit for numbers of 8bit error location in 0th codeword

NANDC_BCHDE0_0

Address: Operational Base + offset (0x0070)

BCH decode result of 0th error bit for codeword 0

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved
18:8	RO	0x000	offset The offset byte address of the error bit. The value is 11bit, which is the byte offset address in the codeword. The address can be divided into different part for different use, showed as follows. 0 ~1023: page data 1024~1027: system information 1028~1055: bch information for 16bitBCH 1028~1069: bch information for 24bitBCH 1028~1097: bch information for 40bitBCH 1028~1132: bch information for 60bitBCH
7:0	RO	0x00	err_val The error value of corresponding error byte

NANDC_BCHDE0_1

Address: Operational Base + offset (0x0074)

BCH decode result of 1th error bit for codeword 0

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved
18:0	RO	0x00000	bchde0_1 Decode result of 1th error bit for codeword 0. Bit assignment is similar to BCHDE0_0 register. For more description, please refer to BCHDE0_0 register.

NANDC_BCHDE0_2

Address: Operational Base + offset (0x0078)

BCH decode result of 2th error bit for codeword 0

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved

Bit	Attr	Reset Value	Description
18:0	RO	0x00000	bchde0_2 Decode result of 2th error bit for codeword 0. Bit assignment is similar to BCHDE0_0 register. For more description, please refer to BCHDE0_0 register.

NANDC_BCHDE0_3

Address: Operational Base + offset (0x007c)
BCH decode result of 3th error bit for codeword 0

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved
18:0	RO	0x00000	bchde0_3 Decode result of 3th error bit for codeword 0. Bit assignment is similar to BCHDE0_0 register. For more description, please refer to BCHDE0_0 register.

NANDC_BCHDE0_4

Address: Operational Base + offset (0x0080)
BCH decode result of 4th error bit for codeword 0

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved
18:0	RO	0x00000	bchde0_4 Decode result of 4th error bit for codeword 0. Bit assignment is similar to BCHDE0_0 register. For more description, please refer to BCHDE0_0 register.

NANDC_BCHDE0_5

Address: Operational Base + offset (0x0084)
BCH decode result of 5th error bit for codeword 0

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved
18:0	RO	0x00000	bchde0_5 Decode result of 5th error bit for codeword 0. Bit assignment is similar to BCHDE0_0 register. For more description, please refer to BCHDE0_0 register.

NANDC_BCHDE0_6

Address: Operational Base + offset (0x0088)
BCH decode result of 6th error bit for codeword 0

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved
18:0	RO	0x00000	bchde0_6 Decode result of 6th error bit for codeword 0. Bit assignment is similar to BCHDE0_0 register. For more description, please refer to BCHDE0_0 register.

NANDC_BCHDE0_7

Address: Operational Base + offset (0x008c)

BCH decode result of 7th error bit for codeword 0

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved
18:0	RO	0x00000	bchde0_7 Decode result of 7th error bit for codeword 0. Bit assignment is similar to BCHDE0_0 register. For more description, please refer to BCHDE0_0 register.

NANDC_BCHDE0_8

Address: Operational Base + offset (0x0090)

BCH decode result of 8th error bit for codeword 0

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved
18:0	RO	0x00000	bchde0_8 Decode result of 8th error bit for codeword 0. Bit assignment is similar to BCHDE0_0 register. For more description, please refer to BCHDE0_0 register.

NANDC_BCHDE0_9

Address: Operational Base + offset (0x0094)

BCH decode result of 9th error bit for codeword 0

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved
18:0	RO	0x00000	bchde0_9 Decode result of 9th error bit for codeword 0. Bit assignment is similar to BCHDE0_0 register. For more description, please refer to BCHDE0_0 register.

NANDC_BCHDE0_10

Address: Operational Base + offset (0x0098)

BCH decode result of 10th error bit for codeword 0

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved
18:0	RO	0x00000	bchde0_10 Decode result of 10th error bit for codeword 0. Bit assignment is similar to BCHDE0_0 register. For more description, please refer to BCHDE0_0 register.

NANDC_BCHDE0_11

Address: Operational Base + offset (0x009c)

BCH decode result of 11th error bit for codeword 0

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved

Bit	Attr	Reset Value	Description
18:0	RO	0x00000	bchde0_11 Decode result of 11th error bit for codeword 0. Bit assignment is similar to BCHDE0_0 register. For more description, please refer to BCHDE0_0 register.

NANDC_BCHDE0_12

Address: Operational Base + offset (0x00a0)

BCH decode result of 12th error bit for codeword 0

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved
18:0	RO	0x00000	bchde0_12 Decode result of 12th error bit for codeword 0. Bit assignment is similar to BCHDE0_0 register. For more description, please refer to BCHDE0_0 register.

NANDC_BCHDE0_13

Address: Operational Base + offset (0x00a4)

BCH decode result of 13th error bit for codeword 0

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved
18:0	RO	0x00000	bchde0_13 Decode result of 13th error bit for codeword 0. Bit assignment is similar to BCHDE0_0 register. For more description, please refer to BCHDE0_0 register.

NANDC_BCHDE0_14

Address: Operational Base + offset (0x00a8)

BCH decode result of 14th error bit for codeword 0

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved
18:0	RO	0x00000	bchde0_14 Decode result of 14th error bit for codeword 0. Bit assignment is similar to BCHDE0_0 register. For more description, please refer to BCHDE0_0 register.

7.5 Interface Description

Table 7-2 NandC Interface Description

Module Pin	Direction	Pad Name	IOMUX Setting
flash_ale	O	FLASH_ALE	GRF_GPIO2A_IOMUX[1:0]=2'b01
flash_cle	O	FLASH_CLE	GRF_GPIO2A_IOMUX[3:2]=2'b01
flash_wrn	O	FLASH_WRN	GRF_GPIO2A_IOMUX[5:4]=2'b01
flash_rdn	O	FLASH_RDN	GRF_GPIO2A_IOMUX[7:6]=2'b01
flash_data[0]	I/O	FLASH_DATA[0]	GRF_GPIO1D_IOMUX[1:0]=2'b01
flash_data[1]	I/O	FLASH_DATA[1]	GRF_GPIO1D_IOMUX[3:2]=2'b01
flash_data[2]	I/O	FLASH_DATA[2]	GRF_GPIO1D_IOMUX[5:4]=2'b01
flash_data[3]	I/O	FLASH_DATA[3]	GRF_GPIO1D_IOMUX[7:6]=2'b01

Module Pin	Direction	Pad Name	IOMUX Setting
flash_data[4]	I/O	FLASH_DATA[4]	GRF_GPIO1D_IOMUX[9:8]=2'b01
flash_data[5]	I/O	FLASH_DATA[5]	GRF_GPIO1D_IOMUX[11:10]=2'b01
flash_data[6]	I/O	FLASH_DATA[6]	GRF_GPIO1D_IOMUX[13:12]=2'b01
flash_data[7]	I/O	FLASH_DATA[7]	GRF_GPIO1D_IOMUX[15:14]=2'b01
flash_rdy	I	FLASH_RDY	GRF_GPIO2A_IOMUX[9:8]=2'b01
flash_csn0	O	FLASH0_CSN	GRF_GPIO2A_IOMUX[12]=1'b1

Notes: I=input, O=output, I/O=input/output, bidirectional

Furthermore, different IOs are selected and connected to flash interface, which is shown as follows.

Table 7-3 NandC Interface Connection

Module Pin	Direction	Flash Interface			
		Asyn8x	Asyn16x	ONFI	Toggle
flash_csn0	O	✓	-	-	-
flash_ale	O	✓	-	-	-
flash_cle	O	✓	-	-	-
flash_wrn	O	✓	-	-	-
flash_rdn	O	✓	-	-	-
flash_data[7:0]	I/O	✓	-	-	-
flash_rdy	I	✓	-	-	-

7.6 Application Notes

7.6.1 BCHST/BCHLOC/BCHDE/SPARE Application

1. BCHST

There are 8 BCHST-registers in NandC to store 16 codeword's BCH decode status(bchst) information. Every register stores 2 codeword's bchst information except BCHST0, which not only includes bchst information, but also includes one bit for *bchrny*.

Let bchst_cwd0~bchst_cwd15 be the bchst information for 16 codewords. In BCHST-registers, the latest codeword's bchst is stored into bchst_cwd0, and the former is shifted into bchst_cwd1. That is, bchst_cwd0 → bchst_cwd1 → → bchst_cwd15. Therefore, for example, if 16 codewords are decoded, then bchst_cwd0 is the bch decode status for codeword15, and bchst_cwd15 is the bch decode status for codeword0.

bchst_cwd0 = {BCHST0[28], BCHST0[12:8], BCHST0[27], BCHST0[7:3], BCHST0[2:0]}

bchst_cwd1 = {BCHST0[30], BCHST0[25:21], BCHST0[29], BCHST0[20:16],
BCHST0[15:13]}

bchst_cwd2 = {BCHST1[28], BCHST1[12:8], BCHST1[27], BCHST1[7:3], BCHST1[2:0]}

bchst_cwd3 = {BCHST1[30], BCHST1[25:21], BCHST1[29], BCHST1[20:16],
BCHST1[15:13]}

bchst_cwd4 = {BCHST2[28], BCHST2[12:8], BCHST2[27], BCHST2[7:3], BCHST2[2:0]}

bchst_cwd5 = {BCHST2[30], BCHST2[25:21], BCHST2[29], BCHST2[20:16],
BCHST2[15:13]}

bchst_cwd6 = {BCHST3[28], BCHST3[12:8], BCHST3[27], BCHST3[7:3], BCHST3[2:0]}

bchst_cwd7 = {BCHST3[30], BCHST3[25:21], BCHST3[29], BCHST3[20:16],
BCHST3[15:13]}

bchst_cwd8 = {BCHST4[28], BCHST4[12:8], BCHST4[27], BCHST4[7:3], BCHST4[2:0]}

bchst_cwd9 = {BCHST4[30], BCHST4[25:21], BCHST4[29], BCHST4[20:16],
BCHST4[15:13]}

bchst_cwd10 = {BCHST5[28], BCHST5[12:8], BCHST5[27], BCHST5[7:3], BCHST5[2:0]}

bchst_cwd11 = {BCHST5[30], BCHST5[25:21], BCHST5[29], BCHST5[20:16],
BCHST5[15:13]}

bchst_cwd12 = {BCHST6[28], BCHST6[12:8], BCHST6[27], BCHST6[7:3], BCHST6[2:0]}

bchst_cwd13 = {BCHST6[30], BCHST6[25:21], BCHST6[29], BCHST6[20:16],
BCHST6[15:13]}

bchst_cwd14 = {BCHST7[28], BCHST7[12:8], BCHST7[27], BCHST7[7:3], BCHST7[2:0]}

bchst_cwd15 = {BCHST7[30], BCHST7[25:21], BCHST7[29], BCHST7[20:16],
BCHST7[15:13]}

2. BCHLOC

There are 4 BCHLOC-registers in NandC to store 16 codeword's bch decode location(bchloc) information.

Let bchloc_cwd0~bchloc_cwd15 be the bchloc information for the 16 codeword. In BCHLOC registers, the latest codeword's bchloc is stored into bchloc_cwd0, and the former is shifted into bchloc_cwd1. That is, bchloc_cwd0→bchloc_cwd1→.....→bchloc_cwd15. Therefore, for example, if 16 codeword are decoded, then bchloc_cwd0 is the bch decode status for codeword15, and bchloc_cwd15 is the bch decode status for codeword0.

```
bchloc_cwd0 = {BCHLOC3[0], BCHLOC0[4:0]}
bchloc_cwd1 = {BCHLOC3[1], BCHLOC0[9:5]}
bchloc_cwd2 = {BCHLOC3[2], BCHLOC0[14:10]}
bchloc_cwd3 = {BCHLOC3[3], BCHLOC0[19:15]}
bchloc_cwd4 = {BCHLOC3[4], BCHLOC0[24:20]}
bchloc_cwd5 = {BCHLOC3[5], BCHLOC0[29:25]}
bchloc_cwd6 = {BCHLOC3[6], BCHLOC1[4:0]}
bchloc_cwd7 = {BCHLOC3[7], BCHLOC1[9:5]}
bchloc_cwd8 = {BCHLOC3[8], BCHLOC1[14:10]}
bchloc_cwd9 = {BCHLOC3[9], BCHLOC1[19:15]}
bchloc_cwd10 = {BCHLOC3[10], BCHLOC1[24:20]}
bchloc_cwd11 = {BCHLOC3[11], BCHLOC1[29:25]}
bchloc_cwd12 = {BCHLOC3[12], BCHLOC2[4:0]}
bchloc_cwd13 = {BCHLOC3[13], BCHLOC2[9:5]}
bchloc_cwd14 = {BCHLOC3[14], BCHLOC2[14:10]}
bchloc_cwd15 = {BCHLOC3[15], BCHLOC2[19:15]}
```

3. BCHDE

BCHDE includes two register-groups, BCHDE0 and BCHDE1. Each group has 60 registers: BCHDE0_0~BCHDE0_59 and BCHDE1_0~BCHDE1_59. BCHDE0_n(n=0~59) is the decode information of the nth error bit for codeword in sram0, and BCHDE1_n(n=0~59) is the decode information of the nth error bit for codeword in sram1.

The needed number of BCHDE registers is determined by bchmode. That is:

- a. When 16bitBCH selected, BCHDEm_0 ~ BCHDEm_15 are available
- b. When 24bitBCH selected, BCHDEm_0 ~ BCHDEm_23 are available
- c. When 40bitBCH selected, BCHDEm_0 ~ BCHDEm_39 are available
- d. When 60bitBCH selected, BCHDEm_0 ~ BCHDEm_59 are available

4. SPARE

SPARE includes two register-groups, SPARE0 and SPARE1. Each group has 28 registers:

SPARE0_0~SPARE0_27 and SPARE1_0~SPARE1_27.

When in bch encoding, SPARE0_0 stores system information for codeword in sram0, SPARE0_n(n=1~27) stores encode information for codeword in sram0; SPARE1_0 stores system information for codeword in sram1, SPARE1_n(n=1~27) stores encode information for codeword in sram1.

When in bch decoding, SPARE0_n(n=0~27) stores the spare data read from flash for codeword in sram0; SPARE1_n(n=0~27) stores the spare data read from flash for codeword in sram1.

The needed number of BCHDE registers is determined by bchmode. That is:

- a. When 16bitBCH selected, spare data=28bytes, SPAREm_0~SPAREm_7 are available
- b. When 24bitBCH selected, spare data=42bytes, SPAREm_0~SPAREm_10 and SPAREm_11[15:0] are available
- c. When 40bitBCH selected, spare data=70bytes, SPAREm_0~SPAREm_17 and SPAREm_18[15:0] are available
- d. When 60bitBCH selected, spare data=105bytes, SPAREm_0~SPAREm_26 and SPAREm_27[7:0] are available

7.6.2 Bus Mode Application

MTRANS_CFG[2] determines whether the data load/store between internal memory and external memory is through slave interface or master interface.

1. Slave Mode

When MTRANS_CFG[2]=0, slave is selected. i.e., flash data load/store between internal

memory and external memory is through slave interface by CPU or external DMA. In this mode, software should store page data into internal memory and spare data into SPARE registers before starting flash program operation; and should load page data from internal memory and spare data from SPARE registers after finishing flash read operation. In this mode, MTRANS_CFG, MTRANS_SADDR0 and MTRANS_SADDR1 are unused. The transfer codeword number is determined by FLCTL[6:5], and the maximum number is 2. The judgment condition for finishing data transfer is FLCTL[20]. When FLCTL[20] is high, it means that data transfer is finished.

2. Master Mode

When MTRANS_CFG[2]=1, master is selected. i.e., flash data load/store between internal memory and external memory is through master interface.

In this mode, software should initialize page data and spare data into external memory, and set their addresses in MTRANS_SADDR0 and MTRANS_SADDR1 respectively before starting flash program operation. Similarly, software should configure MTRANS_SADDR0 and MTRANS_SADDR1 respectively before starting flash read operation and could read data from addresses in MTRANS_SADDR0 and MTRANS_SADDR1 after NandC transfer finish.

In this mode, MTRANS_CFG, MTRANS_SADDR0 and MTRANS_SADDR1 are used. The transfer codeword number is determined by FLCTL[26:22], and the maximum number is 16.

The judgment condition for finishing data transfer is FLCTL[20]. When FLCTL[20] is high, it means that data transmission is finished.

When MTRANS_CFG[2]=1, page data and spare data are stored in the continuous space of external memory respectively.

For page data, source address is named Saddr0, specified in MTRANS_SADDR0. The space can be divided into many continuous units, and the unit size(named PUnit) is 1024 bytes or 512 bytes determined by FLCTL[21] and FLCTL[11]:

- a.when FLCTL[11]=0, PUnit is always equal to 1024 bytes
- b.when FLCTL[11]=1 and FLCTL[21]=0, PUnit is equal to 1024 bytes
- c.when FLCTL[11]=1 and FLCTL[21]=1, PUnit is equal to 512 bytes

For spare data, source address is named Saddr1, specified in MTRANS_SADDR1. The space can be divided into many continuous units, and the unit size(named SUnit) is 64 bytes or 128 bytes determined by BCHCTL[18], FLCTL[11] and FLCTL[21]:

- a.When FLCTL[11]=0 and BCHCTL[18]=0, SUnit is equal to 64 bytes
- b.When FLCTL[11]=0 and BCHCTL[18]=1, SUnit is equal to 128 bytes
- c.When FLCTL[11]=1 and FLCTL[21]=0, SUnit is equal to 128 bytes
- d.When FLCTL[11]=1 and FLCTL[21]=1, SUnit is equal to 64 bytes

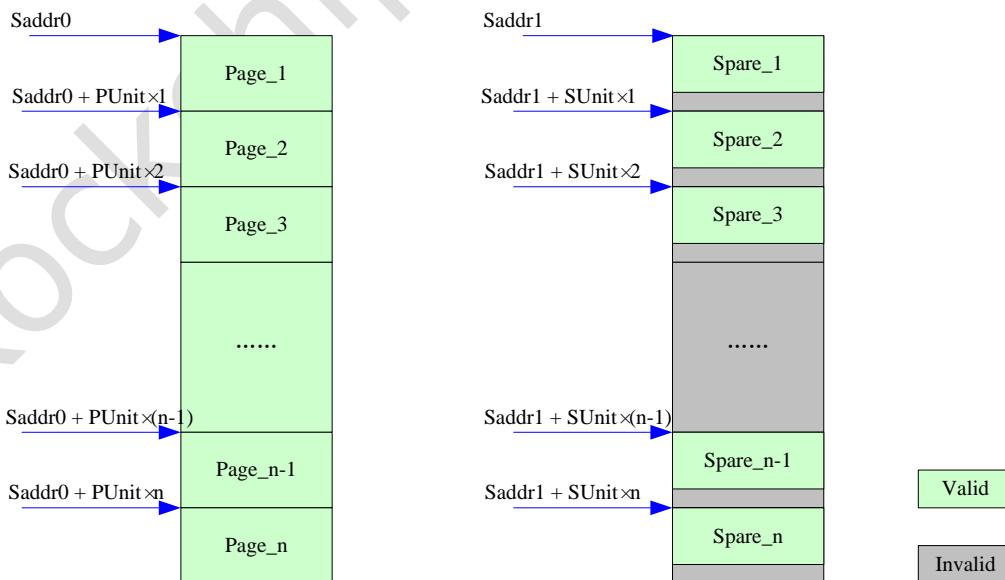


Fig. 7-2 NandC Address Assignment

The detailed format for page data and spare data in every unit is shown in following figures.

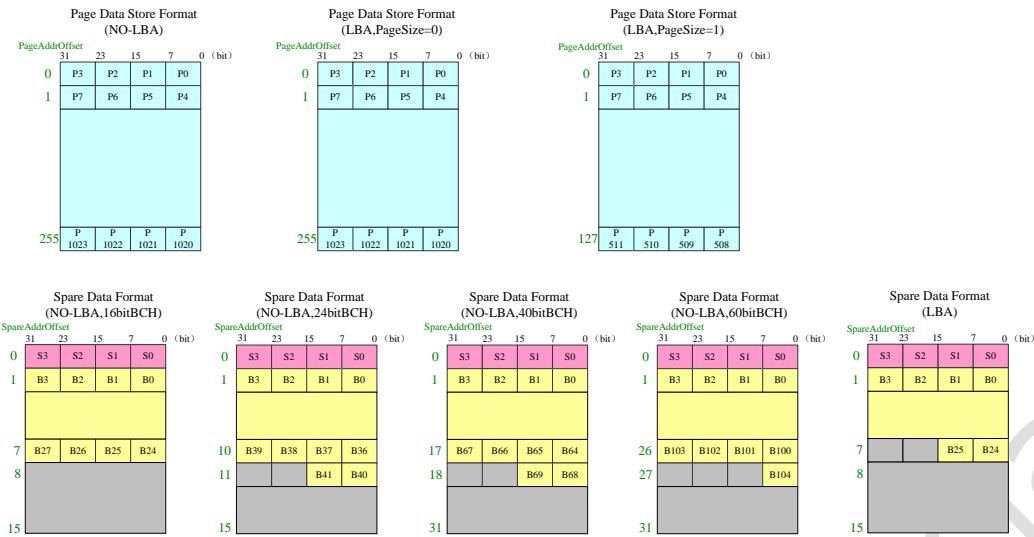


Fig. 7-3 NandC Data Format

7.6.3 BchPage Application

BCHCTL[16] determines whether codeword size for page data is 1024 bytes or 512 bytes when FLCTL[11] is 0.

1. 1024bytes

When BCHCTL[16]=0, BchPage=0, hardware needs to write 1024 bytes page data and spare data into flash or read 1024 bytes page data and spare data from flash. All the 1024 bytes page data and spare data are encoded when writing or decoded when reading.

2. 512bytes

When BCHCTL[16]=1, BchPage=1, hardware needs to write 512 bytes page data and spare data into flash or read 512 bytes page data and spare data from flash.

In this mode, the page data unit size for BCH encoder and BCH decoder still is 1024byte. So to support BCH encoder and decoder, software should configure page data as follows: 1th~512th bytes are invalid data which must be stuffed with 0xff, 513th~1024th bytes are valid page data.

However, Randomizer function is not supported under this condition.

7.6.4 PageSize/SpareSize Application

FLCTL[21] determines whether the codeword size is 1024 bytes or 512 bytes when FLCTL[11] is 1.

1. Big Page

When FLCTL[11]=0(LbaEn=0), the flash to be operated is Raw NAND Flash. Every codeword size is 1024 bytes and FLCTL[21] should always be set to 0, and the PageStep in external memory is 1024 bytes if bus mode is master mode.

At this mode, the spare size and SpareStep in external memory are determined by BCH Mode as follows:

BCH Mode=16bitBCH: spare size=(28+4)bytes , SpareStep=64bytes

BCH Mode=24bitBCH: spare size=(42+4)bytes , SpareStep=64bytes

BCH Mode=40bitBCH: spare size=(70+4)bytes , SpareStep=128bytes

BCH Mode=60bitBCH: spare size=(105+4)bytes, SpareStep=128bytes

2. Small Page

When FLCTL[11]=1, LbaEn=1, the flash to be operated is Managed NAND Flash. Every codeword size could be 1024 bytes or 512 bytes according to FLCTL[21]. If FLCTL[21]=0, codeword size is 1024 bytes, PageStep in external memory is 1024 bytes, and SpareStep is 128bytes. If FLCTL[21]=1, codeword size is 512 bytes, PageStep in external memory is 512 bytes, and SpareStep is 64 bytes.

At this mode, the spare size is configured in FLCTL[18:12], and the max available number is 109.

In the summary, the total data size in every codeword for flash or for software including page data and spare data, is determined by BCHCTL[16], FLCTL[11], FLCTL[21], BCHCTL[4], BCHCTL[18]. Their relationship is shown as follows.

Table 7-4 NandC Page/Spare size for flash

page/spare size for software	page size /codeword	spare size /codeword
FLCTL[11]=0	16bitECC	1024 byte (4+28)byte
	24bitECC	1024 byte (4+42)byte
	40bitECC	1024 byte (4+70)byte
	60bitECC	1024 byte (4+105)byte
FLCTL[11]=1	FLCTL[21]=0	1024 byte FLCTL[18:12]
	FLCTL[21]=1	512 byte FLCTL[18:12]

Notes: that "page/spare size for flash" means that hardware should transfer these numbers of bytes in every codeword to or from flash.

7.6.5 Randomizer Application

RANDMZ_CFG[31] determines whether randomizer is enable or not. When RANDMZ_CFG[31] equals to 1, randomizer is active. Data should be scrambled before written into flash, and descrambled after read from flash.

RANDMZ_CFG[30] determines the randomizer polynomial.

When RANDMZ_CFG[30]=0, Polynomial= $1+x^{18}+x^{23}$

When RANDMZ_CFG[30]=1, Polynomial= $1+x^{14}+x^{15}$

RANDMZ_CFG[22:0] is the seed for randomizer. It should be ensured that data in the same page should have the same randomizer polynomial and randomizer seed when in flash program or flash read operation.

The data unit for randomizer is one codeword(data+spare).

However, Randomizer is just available for data transfer by internal DMA mode, but not by for bypass mode. Furthermore, it should not be enable if BCHCTL[16]=0 (BchPage=512bytes).

7.6.6 NandC Interrupt Application

NandC has 1 interrupt output signal and 4 interrupt sources: DMA finish interrupt source, flash ready interrupt source, bch error interrupt source, BCH fail interrupt source. When one or more of these interrupt source are enabled, NandC interrupt is asserted if one or more interrupt source is high. Software can determine the interrupt source by reading INTST and clear interrupt by writing corresponding bit in INTCLR.

7.6.7 LLP Application

LLP is used in NandC to store and execute instruction groups configured in external memory by software. When LLPCTL[0]=1, LLP is active, NandC will load instruction groups stored in {LLPCTL[31:6], 6'h0} and execute them. Next instruction groups should not be loaded until current instruction execution finished.

1. LLP Structure

The structure of LLP is shown as follows:

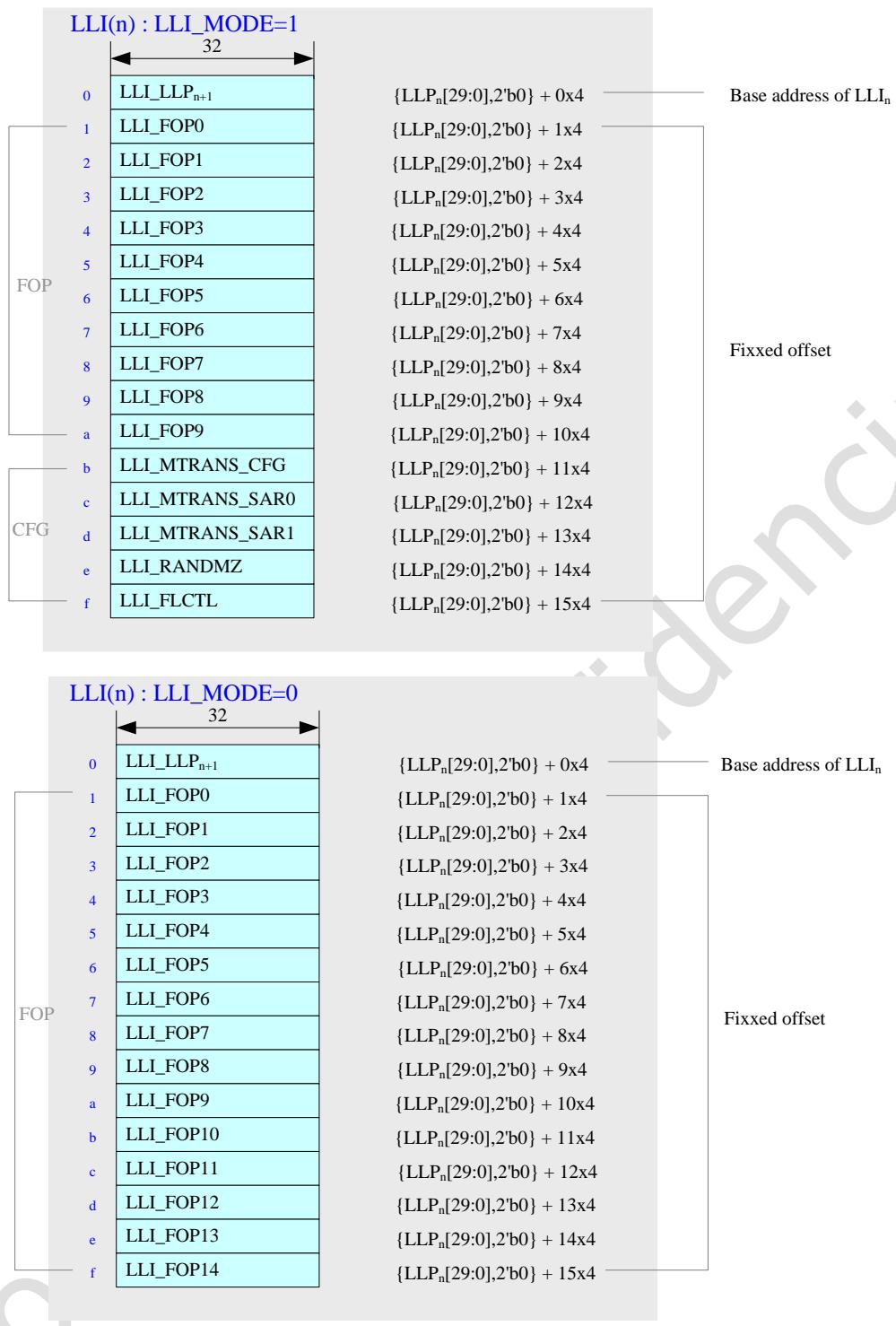
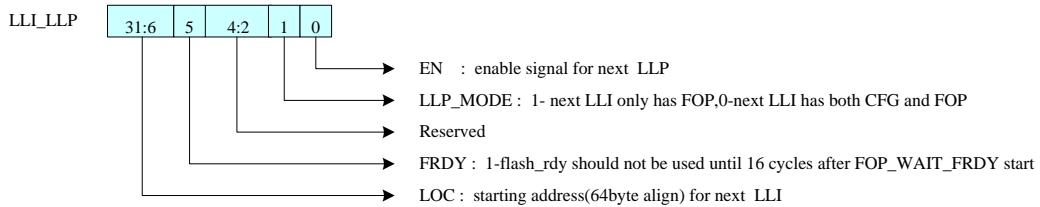


Fig. 7-4 NandC LLP Data Format

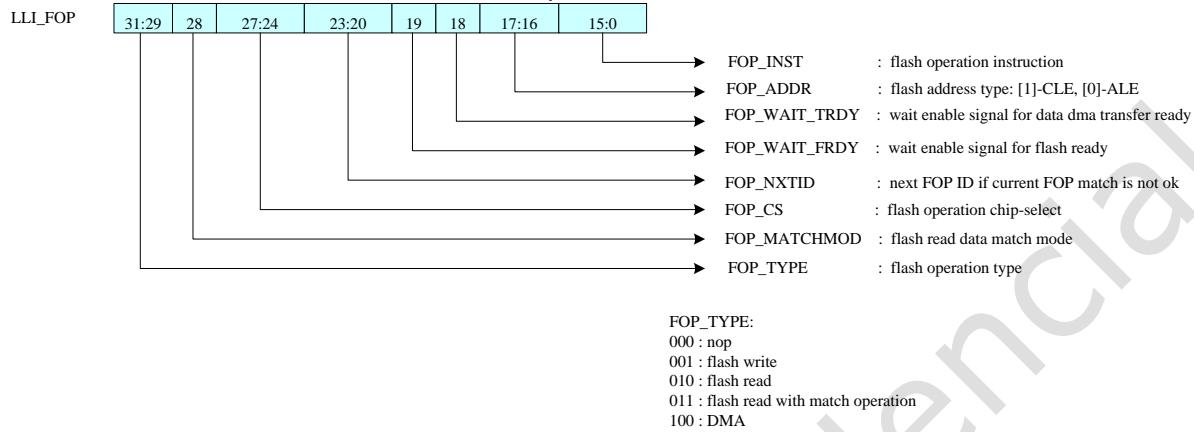
LLI_MODE is determined by LLPCTL[1]. If current operation is flash program or flash read, then LLI_MODE=1 is need; otherwise, LLI_MODE=0 is workable.
In addition, you could do more than one flash operation in one LLP group, but you should not separate one flash operation into two LLI groups.

2. LLI Format

- a. LLI_LLPO_{n+1} stores the address for next LLI group data



b. LLI_FOP0~LLI_FOP14 store the flash operation instruction



When FOP_TYPE=3'b011, match operation is active, and the PATTERN is LLI_FOP[15:0]. It is matched when "RDATA|PATTERN=PATTERN" with FOP_MATCHMOD=0, or when "RDATA&PATTERN=PATTERN" with FOP_MATCHMOD=1.

c.LLI_MTRANS_CFG/LLI_MTRANS_SADDR0/LLI_MTRANS_SADDR1/LLI_RANDMZ/LLI_FLCTL store the configuration for MTRANS_CFG/MTRANS_SADDR0/MTRANS_SADDR1/RANDMZ/FLCTL.

3. LLP Working Mode

There are two working modes for LLP:

- Normal mode: LLPCTL[0] is kept to 1 until all LLP loading and executing finished. Software can monitor the progress by LLPSTAT[31:6], LLPSTAT[0].
- Pause mode: LLPCTL[0] is changed from 1 to 0 during LLP loading or LLP executing. NandC should not stop working until current LLP executing finished. Software can monitor the progress by LLPSTAT[31:6], LLPSTAT[0].

Chapter 8 Timer

8.1 Overview

Timer is a programmable timer peripheral. This component is an APB slave device. Timer0~2 count down from a programmed value and generate an interrupt when the count reaches zero.

Timer5 counts up from zero to a programmed value and generate an interrupt when the count reaches the programmed value.

Timer supports the following features:

- One APB timer in the PERI subsystem, include four programmable 64 bits timer channels, acts as TIMER0, TIMER1, TIMER2, and TIMER5 respectively
- Two operation modes: free-running and user-defined count
- Maskable for each individual interrupt
- TIMER5 is used for CPU; others are used for normalization.

8.2 Block Diagram

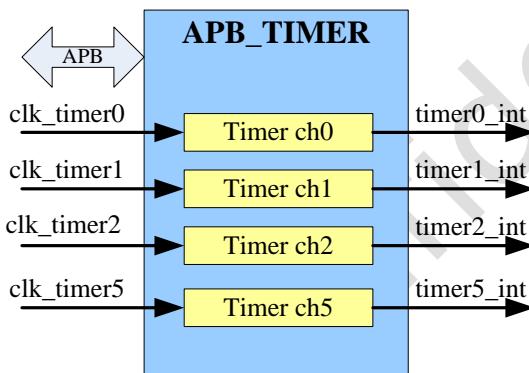


Fig. 8-1 Timer Block Diagram

The above figure shows the architecture of the APB timer (include four programmable timer channel).

8.3 Function Description

8.3.1 Timer clock

The timer clock can be selected from 24MHz OSC or pclk in PERI bus respectively for TIMER0/1/2/5 by CRU_CLKSEL2_CON[0]/[1]/[2]/[3].

8.3.2 Programming sequence

1. Initialize the timer by the TIMERn_CONTROLREG ($n=0,1,2,5$) 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_COUNT1 ($n=0,1,2,5$) and TIMERn_LOAD_COUNT0 ($n=0,1,2,5$) register.
3. Enable the timer by writing a "1" to bit 0 of TIMERn_CONTROLREG ($n=0,1,2,5$).

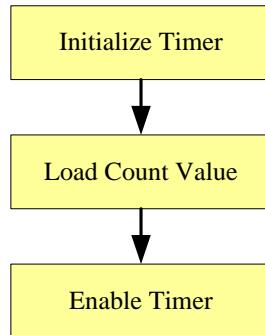


Fig. 8-2 Timer Usage Flow

8.3.3 Loading a timer count value

The initial value for each timer — that is, the value from which it counts down — is loaded into the timer using the load count register (TIMERn_LOAD_COUNT1 (n=0,1,2,5) and TIMERn_LOAD_COUNT0 (n=0,1,2,5)). Two events can cause a timer to load the initial value from its load count register:

- Timer is enabled after reset or disabled.
- Timer counts down to 0, when timer is configured into free-running mode.

8.3.4 Timer mode selection

- User-defined count mode – Timer loads TIMERn_LOAD_COUNT1 (n=0,1,2,5) and TIMERn_LOAD_COUNT0 (n=0,1,2,5) register as initial value. Timer will not automatically load the counter register, when timer counts down to 0. User need to disable timer firstly and follow the programming sequence to make timer work again.
- Free-running mode – Timer loads the TIMERn_LOAD_COUNT1 (n=0,1,2,5) and TIMERn_LOAD_COUNT0 (n=0,1,2,5) register as initial value. Timer will automatically load the counter register, when timer counts down to 0.

8.4 Register Description

This section describes the control/status registers of the design. Software should read and write these registers using 32-bits accesses.

8.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
RKTIMER_n_LOAD_COUNTL	0x0000	W	0x00000001f	Timer n Load Count Register
RKTIMER_n_LOAD_COUNTH	0x0004	W	0x000000000	Timer n Load Count Register
RKTIMER_n_CURRENT_VALUEL	0x0008	W	0x000000000	Timer n Current Value Register
RKTIMER_n_CURRENT_VALUEH	0x000c	W	0x000000000	Timer n Current Value Register
RKTIMER_n_CONTROLREG	0x0010	W	0x000000000	Timer n Control Register
RKTIMER_n_INTSTATUS	0x0014	W	0x000000000	Timer Interrupt Stauts Register

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

8.4.2 Detail Register Description

RKTIMER_n_LOAD_COUNTL

Address: Operational Base + offset (0x0000)

Timer n Load Count Register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000001f	load_count_lowbits Low 32 bits value to be loaded into Timer n. This is the value from which counting commences.

RKTIMER_n_LOAD_COUNTH

Address: Operational Base + offset (0x0004)

Timer n Load Count Register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	load_count_highbits High 32 bits value to be loaded into Timer n. This is the value from which counting commences.

RKTIMER_n_CURRENT_VALUEL

Address: Operational Base + offset (0x0008)

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.

RKTIMER_n_CURRENT_VALUEH

Address: Operational Base + offset (0x000c)

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.

RKTIMER_n_CONTROLREG

Address: Operational Base + offset (0x0010)

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

RKTIMER_n_INTSTATUS

Address: Operational Base + offset (0x0014)

Timer Interrupt Stants Register

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved

Bit	Attr	Reset Value	Description
0	W1C	0x0	int_pd This register contains the interrupt status for timer n. Write 1 to this register will clear the interrupt.

8.5 Application Notes

In the chip, the timer_clk is from 24MHz OSC, asynchronous to the pclk. When user disables the timer enables bit (bit 0 of TIMERn_CONTROLREG (n=0,1,2,5)), 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.

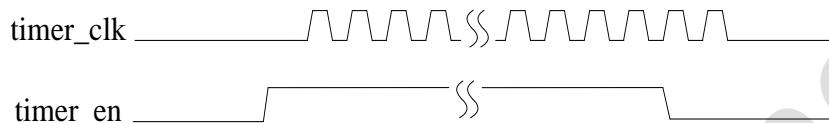


Fig. 8-3 Timing between timer_en and timer_clk

Please refer to function description section for the timer usage flow.

Chapter 9 Generic Interrupt Controller (GIC)

9.1 Overview

The generic interrupt controller (GIC400) in this device has two interfaces, the distributor interface connects to the interrupt source, and the CPU interface connects to Cortex-A7.

It supports the following features:

- Supports 128 hardware interrupt inputs
- Masking of any interrupts
- Prioritization of interrupts
- Distribution of the interrupts to the target Cortex-A7 processor(s)
- Generation of interrupts by software
- Supports Security Extensions

9.2 Block Diagram

The generic interrupt controller comprises with:

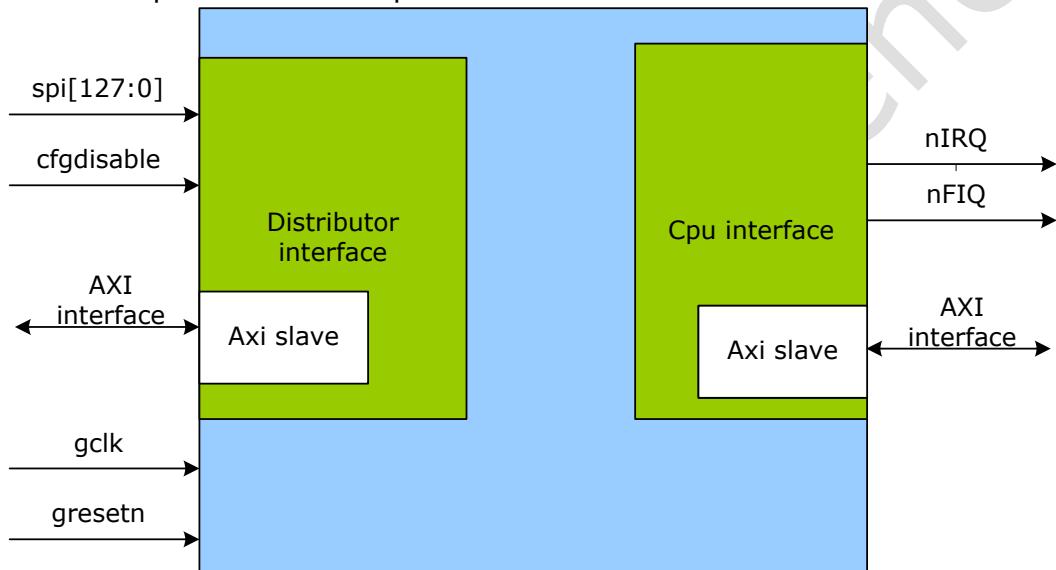


Fig. 9-1 GIC Block Diagram

Chapter 10 DMA Controller(DMAC)

10.1 Overview

DMAC is mainly used for data transfer of the following slaves: I2S, UART0, UART1, UART2, SPI, SDMMC, SDIO, EMMC, SPDIF.

Following table shows the DMAC request mapping scheme.

Table 10-1 DMAC Request Mapping Table

Req number	Source	Polarity
0	I2S tx	High level
1	I2S rx	High level
2	Uart0 tx	High level
3	Uart0 rx	High level
4	Uart1 tx	High level
5	Uart1 rx	High level
6	Uart2 tx	High level
7	Uart2 rx	High level
8	SPI tx	High level
9	SPI rx	High level
10	SDMMC	High level
11	SDIO	High level
12	EMMC	High level
13	SPDIF tx	High level

DMAC supports the following features:

- Supports 14 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 64 MFIFO depth.

10.2 Block Diagram

Following figure shows the block diagram of DMAC.

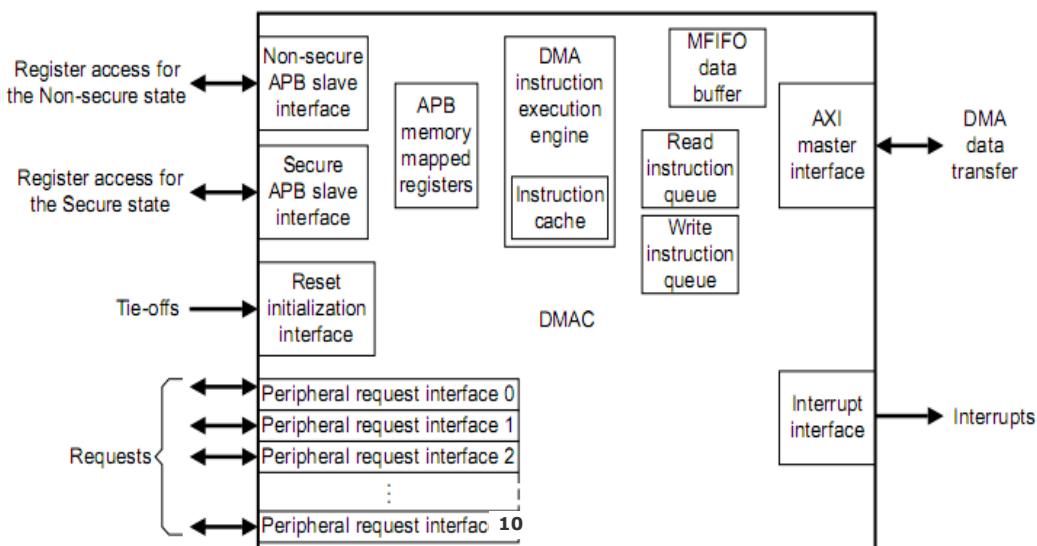


Fig. 10-1 Block diagram of DMAC

10.3 Function Description

10.3.1 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.

DMAC 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.

10.3.2 Operating states

Following figure shows the operating states for the DMA manager thread and DMA channel threads.

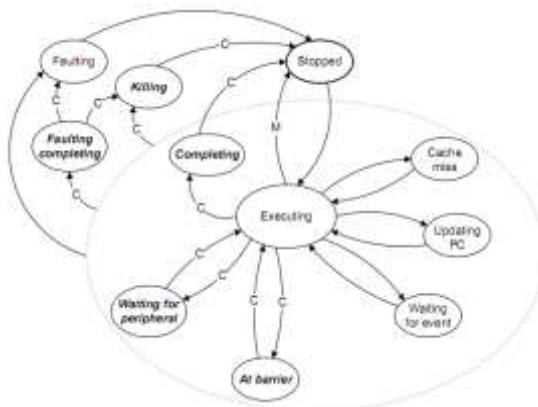


Fig. 10-2 DMAC operation states

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.

10.4 Register Description

10.4.1 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
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

Name	Offset	Size	Reset Value	Description
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
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

Name	Offset	Size	Reset Value	Description
DMAC_DBGINST0	0x0d08	W	0x00000000	Debug Instruction-0 Register
DMAC_DBGINST1	0x0d0c	W	0x00000000	Debug Instruction-1 Register
DMAC_CR0	0xe00	W	0x00047051	Configuration Register 0
DMAC_CR1	0xe04	W	0x00000057	Configuration Register 1
DMAC_CR2	0xe08	W	0x00000000	Configuration Register 2
DMAC_CR3	0xe0c	W	0x00000000	Configuration Register 3
DMAC_CR4	0xe10	W	0x00000006	Configuration Register 4
DMAC_CRDn	0xe14	W	0x02094733	DMA Configuration Register
DMAC_WD	0xe80	W	0x00000000	DMA Watchdog Register

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

10.4.2 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	<p>Program the appropriate bit to control how the DMAC responds when it executes DMASEV:</p> <p>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.</p> <p>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 interrupt request.</p>

DMAC_EVENT_RIS

Address: Operational Base + offset (0x0024)

Event-Interrupt Raw Status Register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	<p>Returns the status of the event-interrupt resources:</p> <p>Bit [N] = 0 Event N is inactive or irq[N] is LOW.</p> <p>Bit [N] = 1 Event N is active or irq[N] is HIGH.</p>

DMAC_INTMIS

Address: Operational Base + offset (0x0028)

Interrupt Status Register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	<p>Provides the status of the interrupts that are active in the DMAC:</p> <p>Bit [N] = 0 Interrupt N is inactive and therefore irq[N] is LOW.</p> <p>Bit [N] = 1 Interrupt N is active and therefore irq[N] is HIGH</p>

DMAC_INTCLR

Address: Operational Base + offset (0x002c)

Interrupt Clear Register

Bit	Attr	Reset Value	Description
31:0	WO	0x00000000	<p>Controls the clearing of the irq outputs:</p> <p>Bit [N] = 0 The status of irq[N] does not change.</p> <p>Bit [N] = 1 The DMAC sets irq[N] LOW if the INTEN Register programs the DMAC to signal an interrupt.</p> <p>Otherwise, the status of irq[N] does not change.</p>

DMAC_FSRD

Address: Operational Base + offset (0x0030)

Fault Status DMA Manager Register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	<p>Provides the fault status of the DMA manager. Read as:</p> <p>0 = the DMA manager thread is not in the Faulting state</p> <p>1 = the DMA manager thread is in the Faulting state.</p>

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

Bit	Attr	Reset Value	Description
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

Bit	Attr	Reset Value	Description
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 DMA LDs have not put enough data in the MFIFO to enable the DMAST to complete. This fault is a precise abort.
12	RO	0x0	Indicates if the MFIFO prevented the DMA channel thread from executing DMA LD or DMA ST. Depending on the instruction: DMA LD 0 = MFIFO contains sufficient space 1 = MFIFO is too small to hold the data that DMA LD requires. DMA ST 0 = MFIFO contains sufficient data 1 = MFIFO is too small to store the data to enable DMA ST 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

Bit	Attr	Reset Value	Description
6	RO	0x0	<p>Indicates if a DMA channel thread, in the Non-secure state, attempts to execute DMAWFP, DMALDP, DMASTP, or DMAFLUSHP with inappropriate security permissions:</p> <p>0 = a DMA channel thread in the Non-secure state is not violating the security permissions</p> <p>1 = a DMA channel thread in the Non-secure state attempted to execute either:</p> <ul style="list-style-type: none"> o DMAWFP to wait for a secure peripheral o DMALDP or DMASTP to notify a secure peripheral o DMAFLUSHP to flush a secure peripheral. <p>This fault is a precise abort.</p>
5	RO	0x0	<p>Indicates if the DMA channel thread attempts to execute DMAWFE or DMASEV with inappropriate security permissions:</p> <p>0 = a DMA channel thread in the Non-secure state is not violating the security permissions</p> <p>1 = a DMA channel thread in the Non-secure state attempted to execute either:</p> <ul style="list-style-type: none"> DMAWFE to wait for a secure event DMASEV to create a secure event or secure interrupt. <p>This fault is a precise abort.</p>
4:2	RO	0x0	reserved
1	RO	0x0	<p>Indicates if the DMA channel thread was attempting to execute an instruction operand that was not valid for the configuration of the DMAC:</p> <p>0 = valid operand</p> <p>1 = invalid operand.</p> <p>This fault is a precise abort.</p>
0	RO	0x0	<p>Indicates if the DMA channel thread was attempting to execute an undefined instruction:</p> <p>0 = defined instruction</p> <p>1 = undefined instruction.</p> <p>This fault is a precise abort</p>

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

Bit	Attr	Reset Value	Description
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
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

Destination Address 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

Bit	Attr	Reset Value	Description
24:22	RO	0x0	<p>Programs the state of AWPROT[2:0]a when the DMAC writes the destination data.</p> <p>Bit [24] 0 = AWPROT[2] is LOW 1 = AWPROT[2] is HIGH.</p> <p>Bit [23] 0 = AWPROT[1] is LOW 1 = AWPROT[1] is HIGH.</p> <p>Bit [22] 0 = AWPROT[0] is LOW 1 = AWPROT[0] is HIGH</p>
21:18	RO	0x0	<p>For each burst, these bits program the number of data transfers that the DMAC performs when it writes the destination data:</p> <p>b0000 = 1 data transfer b0001 = 2 data transfers b0010 = 3 data transfers ... b1111 = 16 data transfers.</p> <p>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</p>
17:15	RO	0x0	<p>For each beat within a burst, it programs the number of bytes that the DMAC writes to the destination:</p> <p>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.</p> <p>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.</p>
14	RO	0x0	<p>Programs the burst type that the DMAC performs when it writes the destination data:</p> <p>0 = Fixed-address burst. The DMAC signals AWBURST[0] LOW. 1 = Incrementing-address burst. The DMAC signals AWBURST[0] HIGH.</p>
13:11	RO	0x0	<p>Set the bits to control the state of ARCACHE[2:0]a when the DMAC reads the source data.</p> <p>Bit [13] 0 = ARCACHE[2] is LOW 1 = ARCACHE[2] is HIGH.</p> <p>Bit [12] 0 = ARCACHE[1] is LOW 1 = ARCACHE[1] is HIGH.</p> <p>Bit [11] 0 = ARCACHE[0] is LOW 1 = ARCACHE[0] is HIGH.</p>

Bit	Attr	Reset Value	Description
10:8	RO	0x0	<p>Programs the state of ARPROT[2:0]a when the DMAC reads the source data.</p> <p>Bit [10] 0 = ARPROT[2] is LOW 1 = ARPROT[2] is HIGH.</p> <p>Bit [9] 0 = ARPROT[1] is LOW 1 = ARPROT[1] is HIGH.</p> <p>Bit [8] 0 = ARPROT[0] is LOW 1 = ARPROT[0] is HIGH.</p>
7:4	RO	0x0	<p>For each burst, these bits program the number of data transfers that the DMAC performs when it reads the source data:</p> <p>b0000 = 1 data transfer b0001 = 2 data transfers b0010 = 3 data transfers ... b1111 = 16 data transfers.</p> <p>The total number of bytes that the DMAC reads into the MFIFO when it executes a DMA LD instruction is the product of src_burst_len and src_burst_size</p>
3:1	RO	0x0	<p>For each beat within a burst, it programs the number of bytes that the DMAC reads from the source:</p> <p>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.</p> <p>The total number of bytes that the DMAC reads into the MFIFO when it executes a DMA LD instruction is the product of src_burst_len and src_burst_size</p>
0	RO	0x0	<p>Programs the burst type that the DMAC performs when it reads the source data:</p> <p>0 = Fixed-address burst. The DMAC signals ARBURST[0] LOW. 1 = Incrementing-address burst. The DMAC signals ARBURST[0] HIGH</p>

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

Bit	Attr	Reset Value	Description
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
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

Bit	Attr	Reset Value	Description
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_CRO

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

Bit	Attr	Reset Value	Description
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
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

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
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

Bit	Attr	Reset Value	Description
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
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.

10.5 Timing Diagram

Following picture shows the relationship between dma_req and dma_ack.

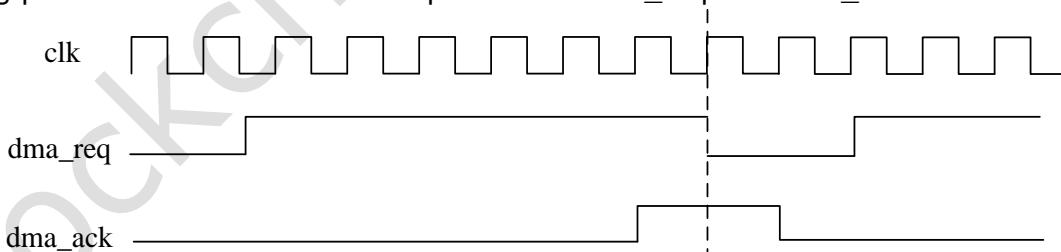


Fig. 10-3 DMAC request and acknowledge timing

10.6 Interface Description

DMAC has the following tie-off signals. It can be configured by GRF register. (Please refer to the chapter to find how to configure)

Table 10-2 DMAC boot interface

Interface	Reset value	Control source
boot_addr	0x0	GRF
boot_from_pc	0x0	GRF
boot_manager_ns	0x1	
boot_irq_ns	0xffff	

Interface	Reset value	Control source
boot_periph_ns	0x3fff	

Notes: *boot_manager_ns*, *boot_irq_ns* and *boot_periph_ns* can't be configured, so dmac can work under non-secure state only

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.

10.7 Application Notes

10.7.1 Using the APB slave interfaces

DMAC can work under non-secure state only, and the secure APB interface is not used. So only the non-secure APB interface can be used to start or restart a DMA channel.

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.

10.7.2 Security usage

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 CR3 Register, 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 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.

10.7.3 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 DMA LD and DMA ST 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.

10.7.4 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 split 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 split 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 10-3 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

10.7.5 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

10.7.6 Instruction sets

Table 10-4 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
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

10.7.7 Assembler directives

In this document, only DMMADNH instruction is took as an example to show the way the instruction assembled. For the other instructions, please refer to pl330_trm.pdf.

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 10-5 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, 0xFFFF causes the value 0xFFFF 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.

10.7.8 MFIFO usage

For MFIFO usage, please refer to pl330_trm.pdf

Chapter 11 Process-Voltage-Temperature Monitor (PVTM)

11.1 Overview

The Process-Voltage-Temperature Monitor (PVTM) is used to monitor the chip performance variance caused by chip process, voltage and temperature.

PVTM supports the following features:

- A clock oscillation ring is integrated and used to generate a clock like signal, the frequency of this clock is determined by the cell delay value of clock oscillation ring circuit.
- A frequency counter is used to measure the frequency of the clock oscillation ring.
- Three PVTM blocks are supported:
 - core_pvtm, used in pd_core power domain
 - gpu_pvtm, used in pd_gpu power domain
 - video_pvtm, used in pd_video power domain

11.2 Block Diagram

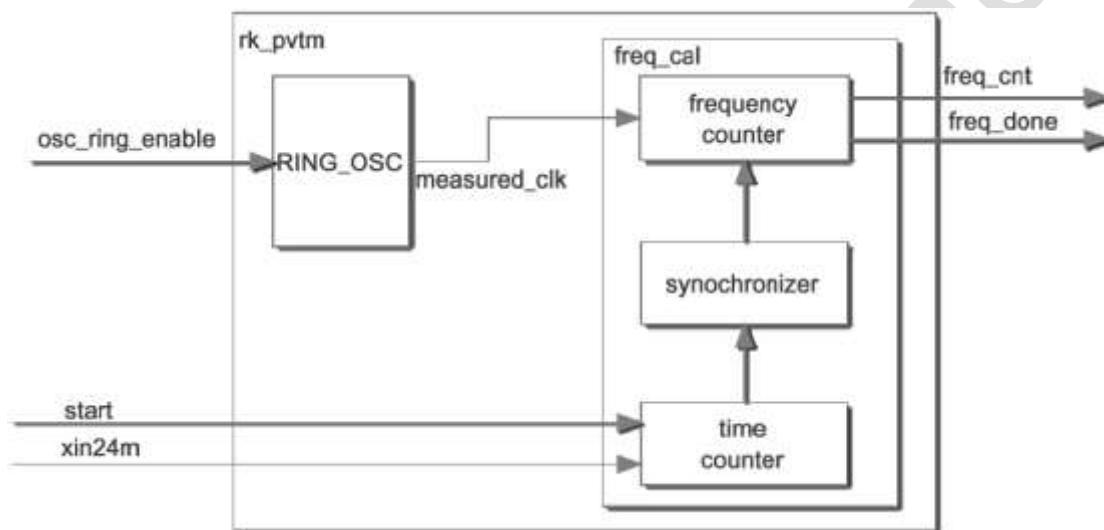


Fig. 11-1 PVTM Block Diagram

The PVTM include two main blocks:

- RING_OSC, it is composed with inverters with odd number, which is used to generate a clock.
- Freq_cal, it is used to measure the frequency of clock which generated from the RING_OSC block.

11.3 Function Description

11.3.1 Frequency Calculation

A clock is generated by the RING_OSC, and a frequency fixed clock (24MHz) is used to calculate the cycles of the clock. Suppose the time period is 1s, then the clock period of RING_OSC clock is $T = 1/\text{clock_counter(s)}$, the cell delay value is $T/2$.

11.3.2 Control Source and Result Destination

The pvtm is controlled by CRU and GRF, and the monitor result is geted by GRF. Following tables shows the three PVTM control source and result destination.

Table 11-1 core_pvtm control source and result destination

Interface	Reset value	Control Source/Result Destination
xin24m	0x0	CRU_CLKGATE10_CON[0], control the clock enable, active low
resetn	0x1	CRU_SOFT_RST1_CON[13] , reverse connect to resetn, active high

Interface	Reset value	Control Source/Result Destination
start	0x0	GRF_PVTM_CON0[0] , active high
osc_ring_enable	0x0	GRF_PVTM_CON0[1] , active high
cal_cnt	0x0	GRF_PVTM_CON1
freq_done	0x0	GRF_PVTM_STATUS0[1]
freq_cnt	0x0	GRF_PVTM_STATUS1

Table 11-2 gpu_pvtm control source and result destination

Interface	Reset value	Control Source/Result Destination
xin24m	0x0	CRU_CLKGATE10_CON[1], control the clock enable, active low
resetn	0x1	CRU_SOFT_RST1_CON[10], reverse connect to resetn, active high
start	0x0	GRF_PVTM_CON0[2], active high
osc_ring_enable	0x0	GRF_PVTM_CON0[3] , active high
cal_cnt	0x0	GRF_PVTM_CON2
freq_done	0x0	GRF_PVTM_STATUS0[0]
freq_cnt	0x0	GRF_PVTM_STATUS2

Table 11-3 video_pvtm control source and result destination

Interface	Reset value	Control Source/Result Destination
xin24m	0x0	CRU_CLKGATE10_CON[2], control the clock enable, active low
resetn	0x1	CRU_SOFT_RST1_CON[9] , reverse connect to resetn, active high
start	0x0	GRF_PVTM_CON0[4] , active high
osc_ring_enable	0x0	GRF_PVTM_CON0[5] , active high
cal_cnt	0x0	GRF_PVTM_CON3
freq_done	0x0	GRF_PVTM_STATUS0[3]
freq_cnt	0x0	GRF_PVTM_STATUS3

11.4 Application Notes

11.4.1 PVTM Usage Flow

1. Enable the frequency fixed clock xin24m.
2. Reset the pvtm.
3. Set osc_ring_enable '1' to enable the generated clock.
4. Configure the cal_cnt to an appropriate value.
4. Set start '1' to calculate the cycles of the generated clock.
5. Wait the freq_done is asserted, then get the value of freq_cnt. The period of RING_OSC clock is $T = \text{cal_cnt} * (\text{Period of 24MHz clock}) / \text{freq_cnt}$, the cell delay value is $T/2$.

Chapter 12 EFUSE

12.1 Overview

RK3036 EFUSE is a parallel-in/parallel-out Electrical Fuse Macro IP which has 256 bits internal nonvolatile one-time programmable EFUSE storage. With a serial interface, 1-bit can be programmed each time in Program (PGM) mode and 8-bit can be read at one time in read mode.

The main features are as follows:

- One-time programmable nonvolatile EFUSE storage cells organized as 32x8 bit
- 1.1V typical core voltage
- The accumulative total time of $1.21V < AVDD(\text{IO_EFUSE_VQPS}) \leq 2.75V$ must not exceed 1s
- Burning requirements:
 - 2.5V typical burning voltage($AVDD(\text{IO_EFUSE_VQPS})$), $AVDD(\text{IO_EFUSE_VQPS})$ must be high during PGM mode
 - 2us burning pulse width
 - Ambient temperature range of $10\sim40^{\circ}\text{C}$

12.2 Block Diagram

In the following diagram, all the signals except power supply AVDD (IO_EFUSE_VQPS) and DVDD are controlled by registers. For detailed description, please refer to detailed register descriptions.

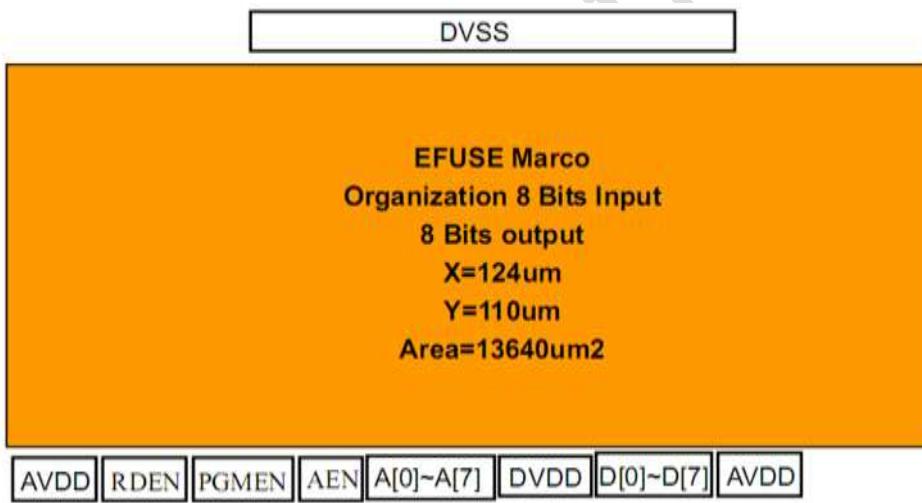


Fig. 12-1 EFUSE Block Diagram

12.3 Function Description

The EFUSE has three operation modes: PGM mode, Read mode, Inactive mode.

The EFUSE enters one of the three modes of operation determined by the logic level of read select signal (RDEN) and the program select (PGMEN). See following table for the corresponding logic levels and operation modes.

Table 12-1 EFUSE Operation Mode

Mode	PGMEN	RDEN	AVDD	DVDD
PGM mode	H	L	H	H
Read mode	L	H	L or Floating	H
Inactive mode	L	L	L or Floating	H

Program (PGM) Mode

Before burning, initial Fuse output is "0", and written to "1" after burning. The EFUSE enters PGM mode if RDEN=L, PGMEN=H. When programming, EFUSE bit specified by address A[7:0] will be burnt by high pulse of AEN.

Read Mode

The EFUSE enters read mode if RDEN=H, PGMEN=L.

Address signals A[7]~A[5] are "invalid".

Table 12-2 EFUSE Read Mode

A[4]~A[0]	D[0]	D[1]	D[2]	D[3]	D[4]	D[5]	D[6]	D[7]
00000	Fuse[0]	Fuse[32]	Fuse[64]	Fuse[96]	Fuse[128]	Fuse[160]	Fuse[192]	Fuse[224]
00001	Fuse[1]	Fuse[33]	Fuse[65]	Fuse[97]	Fuse[129]	Fuse[161]	Fuse[193]	Fuse[225]
00010	Fuse[2]	Fuse[34]	Fuse[66]	Fuse[98]	Fuse[130]	Fuse[162]	Fuse[194]	Fuse[226]
00011	Fuse[3]	Fuse[35]	Fuse[67]	Fuse[99]	Fuse[131]	Fuse[163]	Fuse[195]	Fuse[227]
.
.
.
11111	Fuse[31]	Fuse[63]	Fuse[95]	Fuse[127]	Fuse[159]	Fuse[191]	Fuse[223]	Fuse[255]

Inactive Mode

The EFUSE enters inactive mode if neither PGM mode nor read mode is active. The preferred standby conditions in inactive mode are AEN=L, RDEN=L, PGMEN=L. D[7:0] are undefined in inactive mode.

12.4 Register Description

12.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
EFUSE_CON	0x0000	W	0x00000000	EFUSE control register
EFUSE_DATA	0x0004	W	0x00000000	EFUSE data out register

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

12.4.2 Detail Register Description

EFUSE_CON

Address: Operational Base + offset (0x0000)

EFUSE control register

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:8	RW	0x00	efuse_a address input, A[0]~A[7].
7:4	RO	0x0	reserved
3	RW	0x0	efuse_pgmen program enable, low enable. The efuse_pgmen was valid only when the grf_efuse_prg(bit[13] of register GRF_EFUSE_PRG) was set to "1"
2	RW	0x0	efuse_rden high to turn on sense amplifier and load data into latch. Active-High (1.1v) for read mode; when PGM mode, must be set "low"(0v).
1	RW	0x0	efuse_aen high to turn on the array for read or program access.
0	RO	0x0	reserved

EFUSE_DATA

Address: Operational Base + offset (0x0004)

EFUSE data out register

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RO	0x00	efuse_dout EFUSE data out.

12.5 Timing Diagram

- When EFUSE is in program(PGM) mode

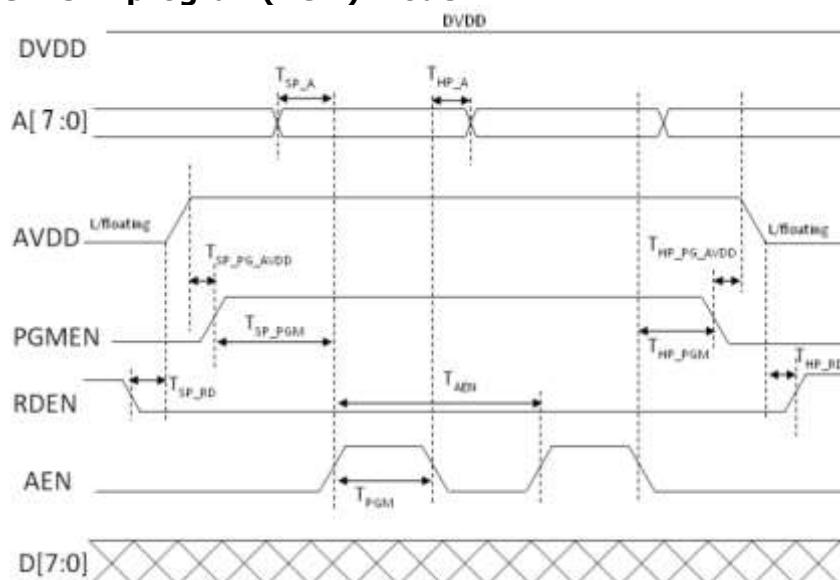


Fig. 12-2 EFUSE Timing Diagram in PGM Mode

The following table shows the detailed requirements for timing parameters in the above diagram.

Table 12-3 EFUSE Timing Requirements in PGM Mode

Parameter	Symbol	Values			Unit
		Min.	Typ.	Max.	
Burning time	T_{PGM}	2000	2000	10000	ns
Address enable cycle time	T_{AEN}	$T_{PGM} + 1900$			ns
Address to AEN setup time	T_{SP_A}	50		—	ns
Address hold time from AEN	T_{HP_A}	50		—	ns
PGMEN signal to AEN setup time	T_{SP_PGM}	100			ns
AEN to PGMEN signal hold time	T_{HP_PGM}	100			ns
RDEN signal to AVDD setup time	T_{SP_RD}	150			ns
AVDD to RDEN signal hold time	T_{HP_RD}	150			ns
AVDD to PGMEN setup time	$T_{SP_PG_AVDD}$	1000			ns
PGMEN to AVDD hold time	$T_{HP_PG_AVDD}$	1000			ns

- When EFUSE is in read mode

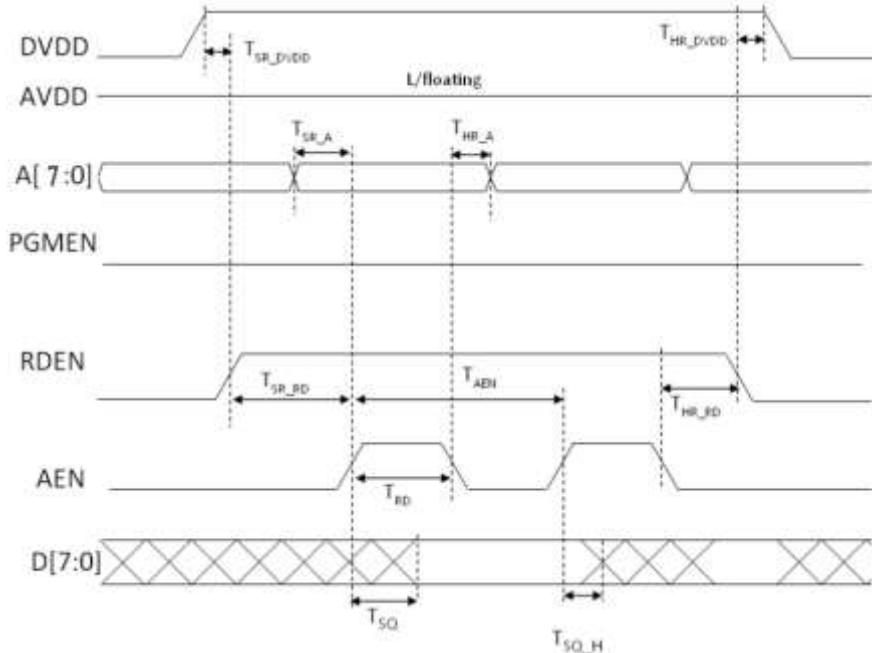


Fig. 12-3 EFUSE Timing Diagram in Read Mode

The following table shows the detailed requirements for timing parameters in the above diagram.

Table 12-4 EFUSE Timing Requirements in Read Mode

Parameter	Symbol	Values			Unit
		Min.	Typ.	Max.	
Read time	T_{RD}	40			ns
Address enable cycle time	T_{AEN}	$T_{RD} + 35$			ns
Address to AEN setup time	T_{SR_A}	10		—	ns
AEN to Address hold time	T_{HR_A}	10		—	ns
DVDD to RDEN setup time	T_{SR_DVDD}	150			ns
RDEN to DVDD hold time	T_{HR_DVDD}	150			ns
RDEN signal to AEN setup time	T_{SR_RD}	100			ns
Output data steady time with 0 loading	T_{SQ}			45	ns
Output data hold time	T_{SQ_H}	0			ns
AEN to RDEN signal hold time	T_{HR_RD}	100			ns

12.6 Application Notes

During usage of EFUSE, customers must pay more attention to the following items:

- (1) PGMEN and RDEN are not allowed to be high at the same time
- (2) AEN high is not allowed except in PGM mode or Read mode
- (3) A port (address) toggle is not allowed when AEN is high in PGM mode or Read mode
- (4) Neither Read mode transitioning to PGM mode directly nor PGM mode transitioning to Read mode directly is not allowed
- (5) For PGM mode converting to inactive mode, AVDD (IO_EFUSE_VQPS) must change from high to low or floating
- (6) AEN/PGMEN/RDEN signal must be held low until DVDD and AVDD supplies have reach their minimum values. DVDD should be set up earlier than AVDD. This prevents unintentional burning of the EFUSE macro during power-up.
- (7) The PGMEN was valid when the grf_efuse_prg (registers GRF_EFUSE_PRG[13]) was set to "1".

Chapter 13 WatchDog

13.1 Overview

Watchdog Timer (WDT) is an APB slave peripheral that can be used to prevent system lockup that may be 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

13.2 Block Diagram

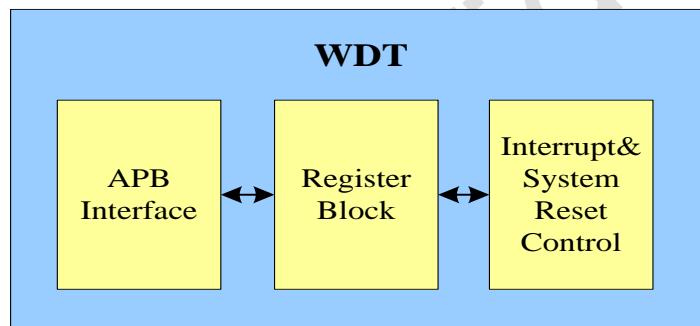


Fig. 13-1 WDT block diagram

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 reads 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.

13.3 Function Description

13.4 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 to 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

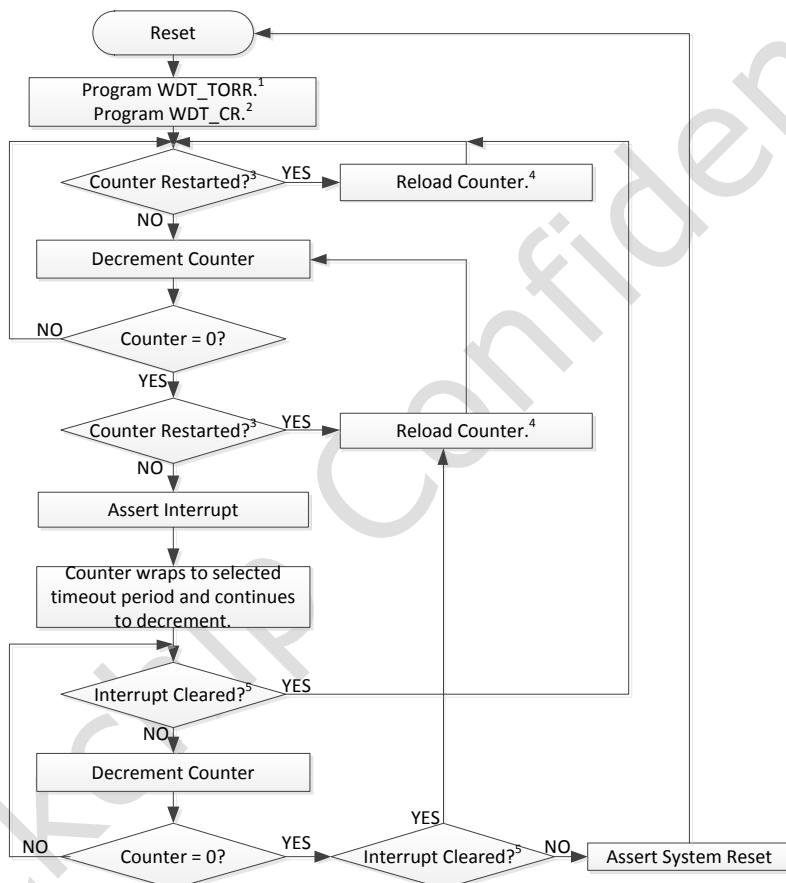
When a 0 is written to the output response mode field (RMOD, bit 1) of the Watchdog Timer Control Register (WDT_CR), the WDT generates a system reset when a timeout occurs.

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.

13.5 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. 13-2 WDT Operation Flow

13.6 Register Description

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

13.6.1 Registers Summary

Name	Offset	Size	Reset Value	Description
WDT_CR	0x0000	W	0x0000000a	Control Register
WDT_TORR	0x0004	W	0x00000000	Timeout range Register

Name	Offset	Size	Reset Value	Description
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

13.6.2 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	<p>rst_pluse_lenth Reset pulse length. This is used to select the number of pclk cycles for which the system reset stays asserted.</p> <p>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</p>
1	RW	0x1	<p>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.</p>
0	RW	0x0	<p>wdt_en WDT enable 0: WDT disabled; 1: WDT enabled.</p>

WDT_TORR

Address: Operational Base + offset (0x0004)

Timeout range Register

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved

Bit	Attr	Reset Value	Description
3:0	RW	0x0	<p>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:</p> <ul style="list-style-type: none"> 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	<p>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</p>

WDT_CRR

Address: Operational Base + offset (0x000c)

Counter restart Register

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	W1C	0x00	<p>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.</p>

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.

13.7 Application Notes

Please refer to the function description section

Chapter 14 System Debug

14.1 Overview

The chip uses the DAPLITE Technology to support real-time debug.

14.1.1 Features

- Invasive debug with core halted
- SW-DP

14.1.2 Debug components address map

The following table shows the debug components address in memory map:

Module	Base Address
DAP_ROM	0x20020000

14.2 Block Diagram

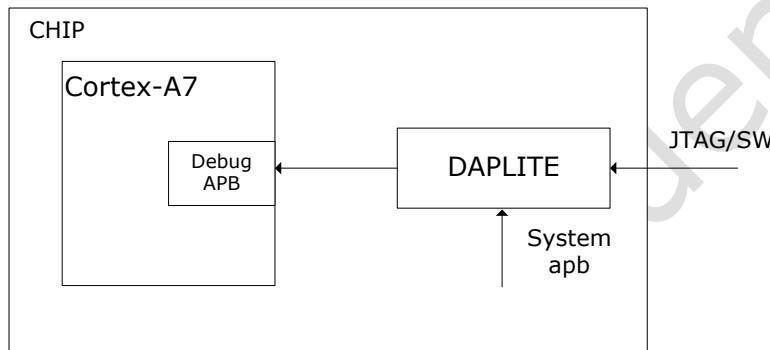


Fig. 14-1 Debug system structure

14.3 Function Description

14.3.1 DAP

The DAP has following components:

- Serial Wire JTAG Debug Port(SWJ-DP)
- APB Access Port(APB-AP)
- ROM table

The debug port is the host tools interface to access the DAP-Lite. This interface controls any access ports provided within the DAP-Lite. The DAP-Lite supports a combined debug port which includes both JTAG and Serial Wire Debug (SWD), with a mechanism that supports switching between them.

The APB-AP acts as a bridge between SWJ-DP and APB bus which translate the Debug request to APB bus.

The DAP provides an internal ROM table connected to the master Debug APB port of the APB-Mux. The Debug ROM table is loaded at address 0x00000000 and 0x80000000 of this bus and is accessible from both APB-AP and the system APB input. Bit[31] of the address bus is not connected to the ROM Table, ensuring that both views read the same value. The ROM table stores the locations of the components on the Debug APB.

More information please refer to the document CoreSight_DAPLite_TRM.pdf for the debug detail description.

14.4 Register Description

Please refer to the document CoreSight_DAPLite_TRM.pdf for the debug detail description.

14.5 Interface Description

14.5.1 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 Wire Debug(SWJ) to JTAG probe to a target.

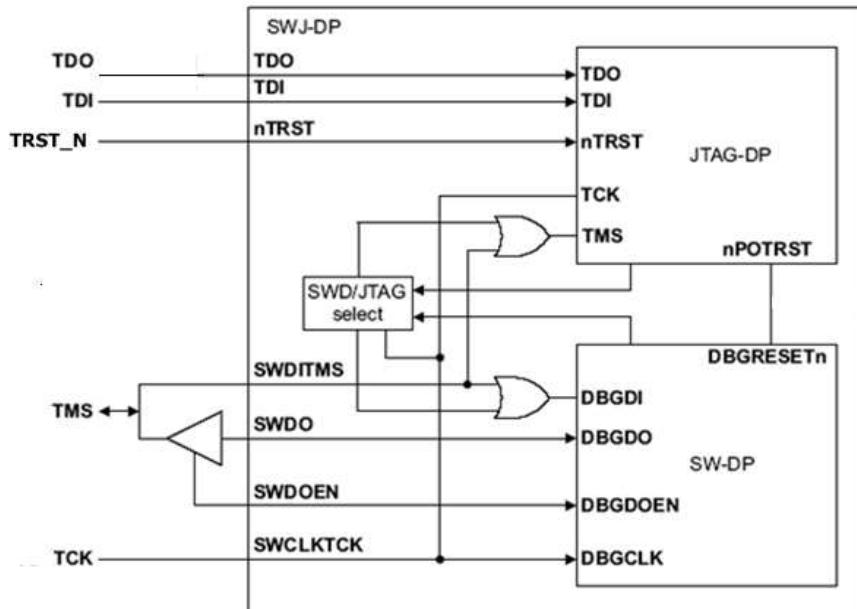


Fig. 14-2 DAP SWJ interface

14.5.2 DAP SW-DP Interface

This implementation is taken from ADIV5.1 and operates with a synchronous serial interface. This uses a single bidirectional data signal, and a clock signal.

The figure below describes the interaction between the timing of transactions on the serial wire interface, and the DAP internal bus transfers. It shows when the target responds with a WAIT acknowledgement.

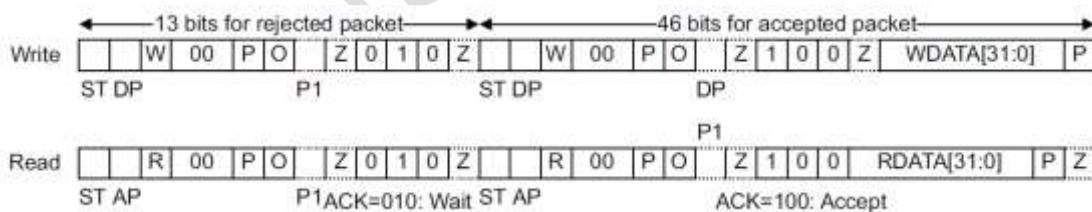


Fig. 14-3 SW-DP acknowledgement timing

Table 14-1 SW-DP Interface Description

Module pin	Direction	Pad name	IOMUX
jtag_tck	I	IO_MMC0d2_JTAGtck_GPIO1c4	GRF_GPIO1C_IOMUX[9:8]=2'b10 & mmc0_detn
jtag_tm s	I/O	IO_MMC0d3_JTAGtms1_GPIO1c5	GRF_GPIO1C_IOMUX[11:10]=2'b10 & mmc0_detn

Note : mmc0_detn, when high, no sd card is used.