JZ4770

Mobile Application Processor

Cores/Systems Programming Manual

Release Date: Aug. 17, 2011

北京君正集成电路股份有限公司
Ingenic Semiconductor Co.,Ltd.

JZ4770 Mobile Application Processor

Cores/Systems Programming Manual

Disclaimer

This documentation is provided for use with Ingenic products. No license to Ingenic property rights is granted. Ingenic assumes no liability, provides no warranty either expressed or implied relating to the usage, or intellectual property right infringement except as provided for by Ingenic Terms and Conditions of Sale.

Ingenic products are not designed for and should not be used in any medical or life sustaining or supporting equipment.

All information in this document should be treated as preliminary. Ingenic may make changes to this document without notice. Anyone relying on this documentation should contact Ingenic for the current documentation and errata.

Ingenic Semiconductor Co., Ltd.

Room 108, Building A, Information Center, Zhongguancun Software Park
8 Dongbeiwang West Road, Haidian District, Beijing, China,
Tel: 86-10-82826661
Fax: 86-10-82825845
Http: //www.ingenic.cn

CONTENTS

TABLES

FIGURES

1 Overview

JZ4770 is a mobile application processor targeting for multimedia rich and mobile devices like smartphone, tablet computer, mobile digital TV, and GPS. This SOC introduces a kind of innovative architecture to fulfill both high performance mobile computing and high quality video decoding requirements addressed by mobile multimedia devices. JZ4770 provides high-speed CPU computing power, good 3D experience and fluent 1080p video replay.

The CPU (Central Processing Unit) core, equipped with 16kB instruction and 16kB data level 1 cache, and 256kB level 2 cache, operating at 1000MHz, and full feature MMU function performs OS related tasks. At the heart of the CPU core is XBurst processor engine. XBurst is an industry leading microprocessor core which delivers superior high performance and best-in-class low power consumption. A hardware floating-point unit which compatible with IEEE754 is also included.

The VPU (Video Processing Unit) core is powered with another XBurst processor engine. The SIMD instruction set implemented by XBurst engine, in together with the on chip video accelerating engine and post processing unit, delivers high video performance. The maximum resolution of 1080p in the formats of H.264, VC-1, MPEG-2, MPEC-4, RealVideo and VP8 are supported in decoding, the maximum resolution of 720p in the format of H.264 are supported in encoding.

The GPU (Graph Processing Unit) core supports numerous 2D/3D graphics applications. It delivers hardware acceleration for 2D and 3D graphics displays, and supports screen sizes range from the smallest cell phones to full HD 1080p displays. It supports the standard APIs such as OpenGL ES2.0 and 1.1, and Open VG. The OS of Android, Linux and Windows are supported. The GPU provides high performance, high quality graphics and low power consumption.

The memory interface supports a variety of memory types that allow flexible design requirements, including glueless connection to SLC NAND flash memory or 4-bit/8-bit/12-bit/16-bit/24-bit ECC MLC/TLC NAND flash memory for cost sensitive applications. It provides the interface to DDR2, DDR and LPDDR memory chips with lower power consumption.

On-chip modules such as audio CODEC, multi-channel SAR-ADC, AC97/I2S controller and camera interface offer designers a rich suite of peripherals for multimedia application. GPS baseband is embedded. TV encoder unit 10-bits DAC provide composite TV signal output in PAL or NTSC format. The LCD controller support up to 1920x1080 output, LVDS as well as plain RGB output which support external HDMI transmitter. The EPD controller supports mainstream vendors' EPD panels in market, up to 5-bit grayscale and 8-zone concurrent updating. WLAN, Bluetooth and expansion options are supported through high-speed SPI and MMC/SD/SDIO host controllers. The TS (Transport stream) interface provides enough bandwidth to connect to an external mobile digital TV demodulator. Other peripherals such as USB OTG and USB 1.1 host, Ethernet MAC with MII and RMII interface, UART and SPI as well as general system resources provide enough computing and connectivity capability for many applications.

1.1 Block Diagram

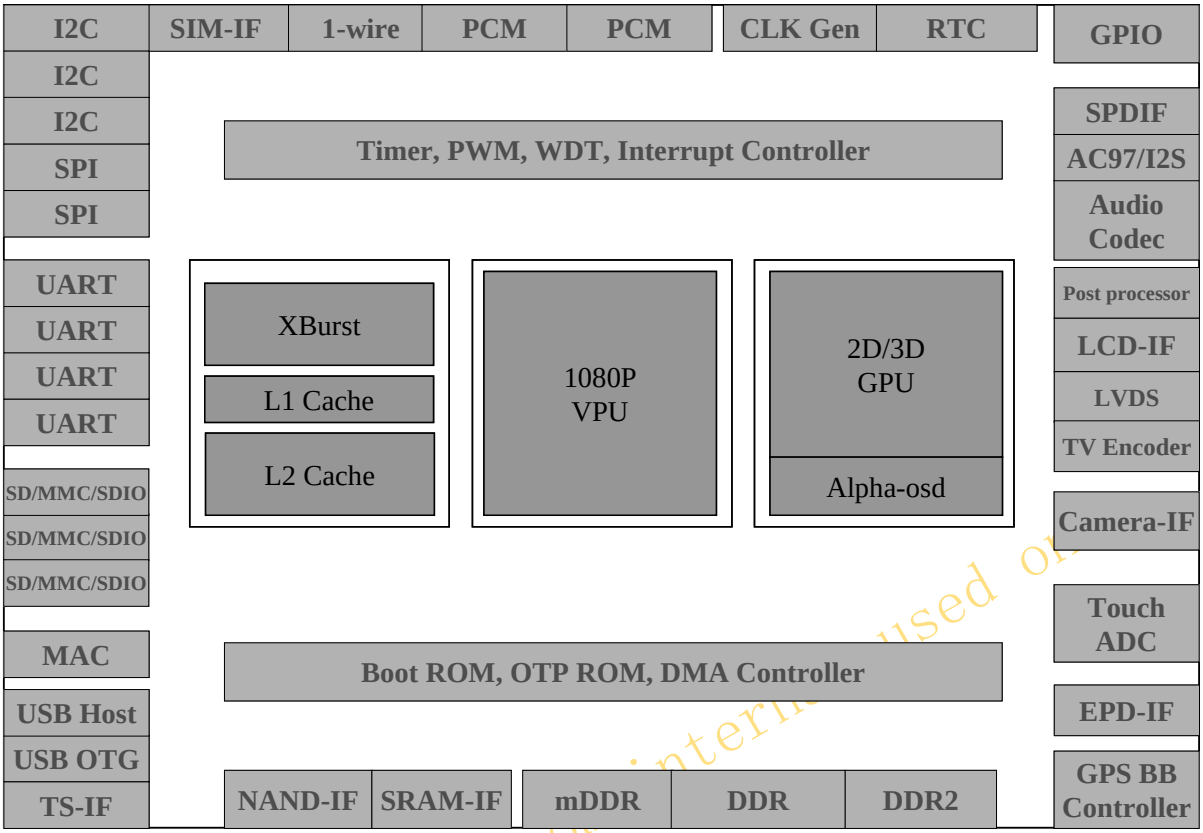


Figure 1-1 JZ4770 Diagram

1.2 Features

1.2.1 CPU Core

- XBurst CPU
  - XBurst® RISC instruction set
  - XBurst® SIMD instruction set
  - XBurst® FPU instruction set supporting both single and double floating point format which are IEEE745 compatible
  - XBurst® 8-stage pipeline micro-architecture up to 1000MHz
- MMU
  - 32-entry joint-TLB
  - 4 entry Instruction TLB
  - 4 entry data TLB
- L1 Cache
  - 16kB instruction cache
  - 16kB data cache
- Hardware debug support
- 16kB tight coupled memory
- L2 Cache
  - 256kB unify cache

1.2.2 VPU Core

- XBurst CPU for video processing
  - XBurst® RISC instruction set
  - XBurst® SIMD instruction set
  - XBurst® 8-stage pipeline micro-architecture up to 500MHz
- Video acceleration engine
  - Motion compensation
  - Motion estimation
  - De-block
  - DCT/IDCT for 4x4 block
  - Parser
- 48kB tight coupled memory
- 28kB scratch RAM

1.2.3 GPU Core

- 2D graphic
  - Bit BLT and stretch BLT
  - Line/Rectangle
  - ROP2, ROP3, ROP4/Alpha blending/scaling/Filter
  - Rotation (90/180/270 degree)/Mirror/Transparency/Rendering

  - Pixel rate up to 200M pix/s
- 3D graphic
  - OpenGL ES2.0 compliance, including extensions
  - OpenGL ES1.1/OpenVG 1.1 compliance
  - DirectFB/GDI/DirectDraw compliance
  - Geometry rate up to 20M tri/s
  - Pixel rate up to 200M pix/s
- Alpha-osd
  - Support ARGB8888, RGB565, RGB555
  - Each layer has an alpha value for all pixels
  - Up to 800*480
  - Software can change overlay orders
  - The level of overlay can be set by software
  - Software must make sure the address of source and destination are 64-word aligned
  - Support 64-burst in AHB bus
  - In RGB656 & RGB555mode, software must make sure each line aligned in word

1.2.4 Memory Sub-systems

- DDR Controller
  - Support DDR2, DDR, mobile DDR (LPDDR) memory
  - Support x16 and x32 external DDR data width
  - Support clock frequency ratio – (BUS clock) : (DDR clock) = 2:1
  - Support clock frequency ratio – (BUS clock) : (DDR clock) = 1:1
  - Support clock-stop mode
  - Support auto-refresh and self-refresh
  - Support power-down mode and deep-power-down mode
  - Programmable DDR timing parameters
  - Programmable DDR row and column address width
- Static memory interface
  - Direct interface to SRAM, ROM, Burst ROM, and NOR Flash
  - Six chip-select pins for static memory, each can be configured separately
  - Support 8 or 16 bits data width
  - 6 bits address
- NAND flash interface
  - Support 4-bit/8-bit/12-bit/16-bit/24-bit MLC/TLC NAND as well as SLC NAND
  - Support all 8-bit/16-bit NAND Flash devices regardless of density and organization
  - Support automatic boot up from NAND Flash devices
- BCH Controller
  - Support 4-bit/8-bit/12-bit/16-bit/20-bit/24-bit ECC encoding and decoding for NAND
- Direct memory access controllers
  - BDMA controller
    - 3 independent DMA channels

  - Support data transfer between normal memory (NAND, SRAM, etc.) / BCH and system memory (DDR)
- General purpose DMA
  - 12 independent DMA channels
  - Support data transfer between On-chip Peripherals (e.g. I2C, MSC, etc.) and system memory (DDR)
  - APB bus bridge
- Common features
  - Descriptor supported
  - Transfer data units: byte, 2-byte (half word), 4-byte (word), 16-byte, 32-byte or 64-byte
  - Transfer number of data unit: 1 ~ 224
  - Independent source and target port width: 8-bit, 16-bit, 32-bit

- The XBurst processor system supports little endian only

1.2.5 AHB Bus Arbiter

- Provide a fair chance for each AHB master to possess the AHB bus
- Fulfill the back-to-back feature of AHB protocol
- Automatic privilege for some masters and programmable privilege for others. Round-robin possession for masters in the same privilege

1.2.6 System Devices

- Clock generation and power management
  - On-chip oscillator circuit for an 32768Hz clock and an 12MHz clock
  - On-chip phase-locked loops (PLL) with programmable multiple-ratio. Internal counter are used to ensure PLL stabilize time
  - PLL on/off is programmable by software
  - ICLK, PCLK, HCLK, HHCLK, MCLK and LCLK frequency can be changed separately for software by setting division ratio
  - Supports six low-power modes and function: NORMAL mode; DOZE mode; IDLE mode; SLEEP mode; HIBERNATE mode; and MODULE-STOP function
  - Support module power-down
- RTC (Real Time Clock)
  - 32-bit second counter
  - 1Hz from 32768hz
  - Alarm interrupt
  - Independent power
  - A 32-bits scratch register used to indicate whether power down happens for RTC power
- Interrupt controller
  - Total 32 maskable interrupt sources from on-chip peripherals and external request through GPIO ports

  - Interrupt source and pending registers for software handling
  - Unmasked interrupts can wake up the chip in sleep or standby mode
- Timer and counter unit with PWM output and/or input edge counter
  - Provide eight separate channels, six of them have input signal transition edge counter
  - 16-bit A counter and 16-bit B counter with auto-reload function every channel
  - Support interrupt generation when the A counter underflows
  - Three clock sources: RTCLK (real time clock), EXCLK (external clock input), PCLK (APB Bus clock) selected with 1, 4, 16, 64, 256 and 1024 clock dividing selected
  - Every channel has PWM output
- OS timer
  - One channel
  - 32-bit counter and 32-bit compare register
  - Support interrupt generation when the counter matches the compare register
  - Three clock sources: RTCLK (real time clock), EXCLK (external clock input), PCLK (APB Bus clock) selected with 1, 4, 16, 64, 256 and 1024 clock dividing selected
- Watchdog timer
  - 16-bit counter in RTC clock with 1, 4, 16, 64, 256 and 1024 clock dividing selected
  - Generate power-on reset

1.2.7 Audio/Display/UI Interfaces

- LCD controller
  - Single-panel display in active mode, and single- or dual-panel displays in passive mode
  - 2, 4, 16 grayscales and up to 4096 colors in STN mode
  - 2, 4, 16, 256, 4K, 32K, 64K, 256K and 16M colors in TFT mode
  - 24-bit data bus
  - Support 1,2,4,8 pins STN panel, 16bit, 18bit and 24bit TFT and 8bit I/F TFT
  - Display size up to 1920x1080 pixels
  - 256×16 bits internal palette RAM
  - Support ITU601/656 data format
  - Support smart LCD (SRAM-like interface LCD module)
  - Support delta RGB
  - One single color background and two foreground OSD
  - Compressed frame supported
  - Support LVDS signal output
- TV encoder
  - Support NTSC or PAL
  - Support CVBS signal
  - 10 bits DAC
- EPD controller
  - Supports Electro-Phoretic Display and compatible devices
  - Supports different size of display panel
  - Supports different width of pixel data

- Supports internal DMA operation and register operation

- Image post processor
  - Video frame resize
  - Color space conversion: 420/444/422 YUV to RGB convert
  - Bi-cubic algorithm supported
  - Video enhancement
- Camera interface module
  - Input image size up to 4096×4096 pixels
  - Supports CCIR656 data format
  - YCbCr 4:2:2 and YCbCr 4:4:4 data format
  - Raw data input
  - 64×32 image data receive FIFO with DMA support
- On-chip audio CODEC
  - 24-bit DAC, SNR: 95dB
  - 24-bit ADC, SNR: 90dB
  - Sample rate: 8/9.6/11.025/12/16/22.05/24/32/44.1/48/96kHz
  - L/R channels line input
  - 2 MICs input, differential or single-ended
  - L/R channels headphone output amplifier support up to 16ohm load
  - Capacitor-coupled
  - Mono differential line out
  - Mono 450mW amplifier for speaker out for 8ohm load
- AC97/I2S/SPDIF controller
  - Supports 8, 16, 18, 20 and 24 bit for sample for AC-link and I2S/MSB-Justified format
  - Support 2/4/6/8 channels data out for I2S
  - Support compress data format for SPDIF
  - DMA transfer mode support
  - Support variable sample rate mode for AC-link format
  - Power down mode and two wake-up mode support for AC-link format
  - Programmable Interrupt function support
  - Support the on-chip CODEC
  - Support off-chip CODEC
  - Support off-chip HDMI transmitter audio
- Two PCM interfaces
  - Data starts with the frame PCMSYN or one PCMCLK later
  - Support three modes of operation for PCM: Short frame sync mode, Long frame sync mode, Multi-slot mode
  - Data is transferred and received with the MSB first
  - Support master mode and slave mode
  - The PCM serial output data, PCMDOUT, is clocked out using the rising edge of the PCMSCLK
  - The PCM serial input data, PCMDIN, is clocked in on the falling edge of the PCMSCLK.
  - 8/16 bit sample data sizes supported

- DMA transfer mode supported
- Two FIFOs for transmit and receive respectively with 16 samples capacity in every direction

- SADC
  - 12-bit, 1Msps/200ksps
  - XP/XN, YP/YN inputs for touch screen
  - Battery voltage inputs for internal/external resistor divider respectively
  - 2 generic input channels
  - 5mW@1Msps, 2.2mW@200ksps

1.2.8 On-chip Peripherals

- General-Purpose I/O ports
  - Total GPIO pin number is 181, where 5 are dedicated and all others are shared
  - Each pin can be configured as general-purpose input or output or multiplexed with internal chip functions
  - Each pin can act as a interrupt source and has configurable rising/falling edge or high/low level detect manner, and can be masked independently
  - Each pin can be configured as open-drain when output
  - Each pin can be configured as internal resistor pull-up/down on or off
- Three I2C bus interfaces
  - Only supports single master mode
  - Supports I2C standard-mode and F/S-mode up to 400 kHz
  - Double-buffered for receiver and transmitter
  - Supports general call address and START byte format after START condition
- Two Synchronous serial interfaces (SSI0, SSI1)
  - Up to 50MHz speed
  - Supports three formats: TI's SSP, National Microwire, and Motorola's SPI
  - Configurable 2 - 17 (or multiples of them) bits data transfer
  - Full-duplex/transmit-only/receive-only operation
  - Supports normal transfer mode or Interval transfer mode
  - Programmable transfer order: MSB first or LSB first
  - 17-bit width, 128-level deep transmit-FIFO and receive-FIFO
  - Programmable divider/prescaler for SSI clock
  - Back-to-back character transmission/reception mode
- One-wire bus interface
  - Overdrive and regular speed
  - Master only
  - LSB first
  - Bit or byte operate modes
- USB 1.1 host interface
  - Open Host Controller Interface (OHCI)-compatible and USB Revision 1.1-compatible
  - Full speed and low speed

  - Embedded USB 1.1 PHY
- USB 2.0 OTG interface
  - Compliant with USB protocol revision 2.0 OTG
  - High speed and full speed supported for device role
  - High speed, full speed and low speed supported for host role
  - Embedded USB OTG PHY
- Ethernet MAC interface
  - Compliant with IEEE802.3
  - 10/100 Mbps data transfer rate with full and half duplex modes
  - MII/RMII interface to talk to an external PHY
- Three MMC/SD/SDIO controllers (MSC0, MSC1, MSC2)
  - Support automatic boot up from MSC0, which has 4-bit data bus
  - MSC1 with 4-bit data bus
  - Compliant with "The MultiMediaCard System Specification version 4.2"
  - Compliant with "SD Memory Card Specification version 2.0" and "SDIO Card Specification version 1.0" with 1 command channel and 4 data channels
  - Up to 320 Mbps data rate in MSC0
  - Up to 320 Mbps data rate in MSC1
  - Supports up to 10 cards (including one SD card)
  - Maskable hardware interrupt for SD I/O interrupt, internal status, and FIFO status
- Four UARTs (UART0, UART1, UART2, UART3)
  - 5, 6, 7 or 8 data bit operation with 1 or 1.5 or 2 stop bits, programmable parity (even, odd, or none)
  - 32x8bit FIFO for transmit and 32x11bit FIFO for receive data
  - Interrupt support for transmit, receive (data ready or timeout), and line status
  - Supports DMA transfer mode
  - Provide complete serial port signal for modem control functions
  - Support slow infrared asynchronous interface (IrDA)
  - IrDA function up to 115200bps baudrate
  - UART function up to 3.7Mbps baudrate
  - Hardware flow control
- SIM IF
  - Supports normal card and UIM card
  - 8-bit, 16-level receive-/transmit- FIFO
  - Supports asynchronous character (T=0) communication modes
  - Supports asynchronous block (T=1) communication modes
  - Supports setting of clock-rate conversion factor F (372, 512, 558, etc.), and bit-rate adjustment factor D (1, 2, 4, 8, 16, 32, 12, 20, etc.)
  - Supports extra guard time waiting
  - Auto-error detection in T=0 receive mode
  - Auto-character repeat in T=0 transmit mode
  - Transforms inverted format to regular format and vice versa
  - Support stop clock function in some power consuming sensitive applications

- Transport stream slave interface
  - 8-bit or 1-bit data bus selectable
  - Support PID filtering
- OTP Slave Interface
  - Total 256 bits. Lower 128bits are read-able and write-able, Higher 128bits are read only

1.2.9 Bootrom

- 8kB Boot ROM memory

1.3 Characteristic

Item	Characteristic
Process Technology	65nm CMOS low power
Power supply voltage	General purpose I/O: 1.6~3.6V DDR I/O for mDDR: 1.8V± 0.2V DDR I/O for DDR: 2.5V± 0.2V DDR I/O for DDR2: 1.8V± 0.2V RTC I/O: 3.0V~3.6V EFUSE programming: 2.5V± 10% Analog power suppy 1: 2.5V± 10% Analog power suppy 2: 3.3V± 10% Core: 1.2 -0.1/+0.2 V
Package	BGA379 14mm x 14mm x 1.1mm, 0.65mm pitch
Operating frequency	1000MHz

2 CPU Core

Enhanced features of CPU core include:

- Enhanced MXU implements XBurst SIMD instruction set release I and release II
- Full implementation of MIPS32 integer instruction release II
- TCSM, tightly coupled shared memory with physical address scope 0x132B0000 ~ 0x132BFFFF
- PMON, processor performance monitor
- FPU, floating point unit implemented to improve floating point number processing ability
- Unified level 2 cache that is transparent for programmer

2.1 Block Diagram

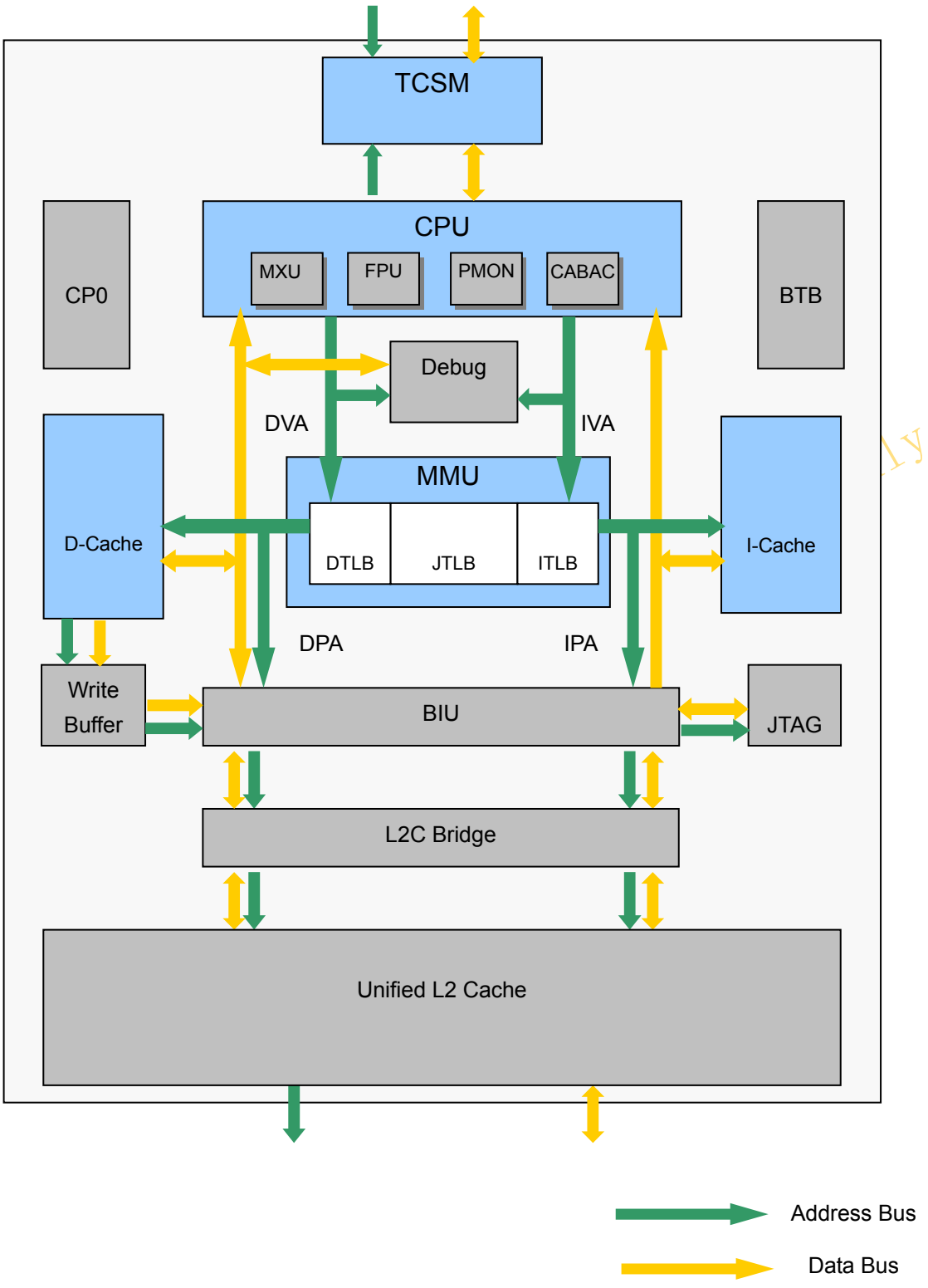


Figure 2-1 Structure of CPU core

2.2 Extra Features of the CPU core

Item	Features
Media Extension Unit (MXU)	• XBurst SIMD instruction set release I and release II • fully pipelined
Integer Unit with MIPS32 integer instruction release II	• non full pipelined implementation for most of MIP32 integer instruction release II, need 2 ~ 4 interlock cycles
Tightly Coupled Sharing Memory (TCSM)	• Size: 16K bytes • Same clock frequency as L1 cache • AHB slave interface • Four banks support up to four simultaneous accesses
Floating Point Unit (FPU)	• Comply with IEEE754 standard • Support single and double format • not fully pipelined implementation
CABAC interface	• Part of bitstream processing cooperating CABAC in VPU • Dedicated CP0 interface is CP0 register number 21, select0~7
Performance Monitor (PMON)	• Real-time monitor • Dedicated CP0 interface
Unified Level 2 Cache	• Size: 256K bytes • 4 way set association with LRU replacement • Write from Level 1 data cache always write through to memory • Programmer transparent, that is, those CACHE instructions managing L1 cache can manipulate L2 cache automatically
Processor ID	Value read from CP0.PRId is 0x2ed1024f

Please refer to documents XBurst-ISA and XBurst1_PM for ISA and programming relative details.

2.3 Instruction Cycles

Most instructions have one cycle repeat rate, that is, when the pipeline is fully filled, there is one instruction issued per clock cycle. However, some particular instructions require extra cycles. Following table lists cycle consumption of all instructions belonging to XBurst-ISA implemented.

1st Instruction	2nd Instruction	Cycles	Description
WAIT	**Anyone**	variable	WAIT instruction will be repeatedly executed until an interrupt arise.
MTC0 **TLBWI/TLBWR** **TLBP/TLBR**	**Anyone**	4	3 extra interlock cycles.
CACHE	**Anyone**	2	1 extra interlock cycles.
JMP/BC	**Anyone** **(delay slot)**	4/1	0 cycle penalty when BTB predicts taken and the branch is taken or BTB predicts untaken and the branch is untaken or BTB miss and the branch is untaken. Otherwise, extra 3 cycles penalty.
BCL	**Anyone** **(delay slot)**	5/4/2/1	0 cycle penalty when BTB predicts taken and branch is taken, otherwise: 1 BTB miss, branch is taken, 3 cycles penalty. 2 BTB miss, branch is untaken, 1 cycle penalty. 3 BTB predict taken, branch is untaken, 4 cycles penalty. 4 BTB predict untaken, branch is taken, 3 cycles penalty.
MULT/MULTU **MADD/MADDU** **MSUB/MSUBU**	**MULT/MULTU** **MADD/MADDU** **MSUB/MSUBU**	4	3 extra interlock cycles due to MDU operating hazard.
	MUL/DIV/DIVU	4	3 extra interlock cycles due to MDU operating hazard.
	MFHI/MFLO **MTHI/MTLO**	4	3 extra interlock cycles due to MDU operating hazard.
	Any other	1	No data dependency or hazards exist.
MUL	**MULT/MULTU** **MADD/MADDU** **MSUB/MSUBU**	4	3 extra interlock cycles due to MDU operating hazard.
	MUL/DIV/DIVU	4	3 extra interlock cycles due to MDU operating hazard.
	MFHI/MFLO **MTHI/MTLO**	4	3 extra interlock cycles due to MDU operating hazard.

	Any other	4/3/2/1	If the second instruction has RAW data dependency, 3 extra interlock cycles; similarly, 2 extra for the third RAW one and 1 extra for the forth RAW one, otherwise, 0 cycle penalty.
DIV/DIVU	**MULT/MULTU MADD/MADDU MSUB/MSUBU MUL/DIV/DIVU**	4~35	3~34 extra interlock cycles determined by characteristic value of divider and dividend.
	MFHI/MFLO	2~34	1~33 interlock cycles determined by characteristic value of divider and dividend.
	Any other	1	No data dependency or hazards exist.
MFHI/MFLO/MFC0	**Anyone**	4/3/2/1	If the second instruction has RAW data dependency, 3 extra interlock cycles, similarly, 2 extra for the third RAW one and 1 extra for the forth RAW one, otherwise, 0 cycle penalty.
LW/LL LWL/LWR LB/LBHU LH/HU LXW LXH/LXHU LXB/LXBU	**Anyone**	4/3/2/1	If the second instruction has RAW data dependency, 3 extra interlock cycles, similarly, 2 extra for the third RAW one and 1 extra for the forth RAW one, otherwise, 0 cycle penalty.
D16MUL/D16MULF D16MAC/D16MACF D16MULE/D16MACE	**SIMD instruction**	3/2/1	If the second SIMD instruction has RAW data dependency, 2 extra interlock cycles, similarly, 1 extra for the third RAW one, otherwise, 0 cycle penalty.
	Any other	1	No data dependency or hazards exist.
D32ACC/Q16ACC Q8SAD S32MAX/S32MIN D16MAX/D16MIN D32ACCM/D32ASUM Q16ACCM/D16ASUM	**SIMD instruction**	2/1	If the second SIMD instruction has RAW data dependency, 1 extra interlock cycle, otherwise, 0 cycle penalty.
	Any other	1	No data dependency or hazards exist.
S32LDD/S32LDDV S32LDI/S32LDIV S32LDDR/S32LDDVR S32LDIR/S32LDIVR S16LDD/S16DI	**SIMD instruction**	2/1	If the second SIMD instruction has RAW data dependency, 1extra interlock cycle, otherwise, 0 cycle penalty.
	Any other	1	No data dependency or hazards exist.

S8LDD/S8LDI			
S32I2M	**SIMD instruction**	2/1	If the second SIMD instruction has RAW data dependency, 1extra interlock cycle, otherwise, 0 cycle penalty.
	Any other	1	No data dependency or hazards exist.
S32M2I	**Anyone**	4/3/2/1	If the second instruction has RAW data dependency, 3 extra interlock cycles, similarly, 2 extra for the third RAW one and 1 extra for the forth RAW one, otherwise, 0 cycle penalty.
S32EXTR **S32EXTRV**	**SIMD instruction**	2/1	If the second SIMD instruction has RAW data dependency, 1extra interlock cycle, otherwise, 0 cycle penalty.
	Any other	1	No data dependency or hazards exist.
Others	**Anyone**	1	--

NOTE: JMP denotes J and JR instructions; BC denotes branch conditionally instructions; BCL denotes branch conditionally and likely instructions.

2.4 TCSM

TCSM (tightly-coupled shared memory) is a dedicated on-chip SRAM. It serves as an on-chip scratchpad memory, moreover, it acts as a high-speed SRAM for CPU. Through the TCSM, CPU and VPU's AHB masters such as DBlock can exchange data quickly and efficiently. TCSM in the CPU core has following features:

- 16K bytes
- The same clock frequency as L1 cache
- Physical address scope from 0x132B,0000 to 0x132B,FFFF
- Four banks support up to four simultaneous accesses if no bank conflicts occurs

Moreover, like the **dseg** section separated from K3 section, another **tcsm** section with 16MB capacity range from 0xF400,0000 to 0xF4FF,FFFF is separated too. This virtual address section is uncacheable and unmappable and can only be accessed by CPU core in kernel mode.

Please note the fact that the capacity of TCSM is only 16K bytes, which denotes that available virtual address range is from 0xF400,0000 to 0xF400,3FFF and available physical address range is from 0x132B,0000 to 0x132B,3FFF.

2.4.1 TCSM Occupied Available Physical Address Range

Physical Address range 0x132B,0000 ~ 0x132B,FFFF are reserved for TCSM. Physical address range 0x132B,0000 ~ 0x132B,3FFF are available and others are reserved, and corresponding address partition for the four banks are as following:

bank0: 0xF4000000~0xF4000FFF (virtual); 0x132B,0000~0x132B,0FFF (physical)
bank1: 0xF4001000~0xF4001FFF (virtual); 0x132B,1000~0x132B,1FFF (physical)
bank2: 0xF4002000~0xF4002FFF (virtual); 0x132B,2000~0x132B,2FFF (physical)
bank3: 0xF4003000~0xF4003FFF (virtual); 0x132B,3000~0x132B,3FFF (physical)

Therefore, arranging instructions and data into different banks can achieve best access performance. Similarly, using ping-pong buffers located in the separate banks for efficient data exchange between CPU core and other VPU's AHB masters is a better choice.

2.5 PMON

PMON is a simple performance monitor. In JZ4770, PMON can make real-time monitoring for following hardware events.

- I-cache miss times, D-cache miss times
- Total issued instructions, Discarded instructions
- Pipeline freeze cycles, CPU clock cycles
- TLB exceptions caused by instruction fetch, TLB exceptions caused by data load/store

Moreover, in JZ4770, PMON can be configured to work in expected mode.

- Normal mode (when PMON is enabled, always work until it is disabled)
- User mode (when PMON is enabled, only work in user mode and paused in kernel mode)
- Kernel mode (when PMON is enabled, only work in kernel mode and paused in user mode)
- PC mode (when PMON is enabled, only work when PC locates in preset address range)

A dedicated software interface is devised to manipulate PMON in kernel mode, that is, CP0 Config4 ~ Config7 registers are extended for PMON. Refer to chapter of CP0 in the document XBurst1_PM for detail. However, since new function and bit fields has been expanded for Config7 register, following definition of Config7 is the precise description for JZ4770.

Config16 **select 7**

Bit	31:16	15:12	11:9	8	7	6	5:3	2	1	0
Field	PC_HI	PEVENT	MODE	PME	Reserved	PK	Reserved	ALLOC	BTBV	BTBE
RST	? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ?	0 0 0 0	0 0 0	0	0	0	0 0 0	0	0	0

Bits	Name	Description	R/W
31:16	PC_HI	If PMON work in PC range mode, the valid range is PC_HI,0000 ~ PC_HI,FFFC.	RW
15:12	PEVENT	Event pair encoding. 0000: count of pipeline freeze cycles, count of cpu clock cycles 0001: times count of icache-miss, times count of dcache-miss 0010: count of discarded instructions, count of issued instructions 0011: count of TLB exceptions caused by fetching instruction, count of TLB exceptions caused by data load/store 0100 – 1111: reserved	RW
11:9	MODE	000: null mode, work unconditionally 001: work in user mode 010: work in kernel mode 011: work in specific PC range	RW
8	PME	PMON enable bit. 0: disable; 1: enable.	RW
7	Reserved	Writing has no effect, read as zero.	R

6	PK	Partial kernel mode. 0: forbid; 1: permit. Refer to later description.	RW
5:3	Reserved	Writing has no effect, read as zero.	R
2	ALLOC	Allocate hint of PREF instruction. 0: enabled (default); 1: disabled.	RW
1	BTBV	BTB invalid. Writing 1 to this bit to invalidates BTB.	W
0	BTBE	BTB enable. 0: enabled (default); 1: disabled.	RW

2.5.1 Fundamental

When PMON is enabled (set value 1 to config7.bit8), one preset event pair determined by config7.bit15~bit12 will be continuously monitored until PMON is disabled (set value 0 to config7.bit8). Finally, loading values of CP0.config4~CP0.config6 can get monitored result.

2.6 Partial Kernel Mode

Setting 1 to config7.bit6 can permit applications in user mode possess some kernel mode oriented resources including TCSM, CABAC I/F, CACHE instructions. This is a shortcut for those performance sensitive applications such as video codec. However, OS must make serious control for config7.bit6 and those dedicated resources to forbid this partial-kernel-mode permission for those malicious applications.

3 VPU Core

Video Processing Unit (VPU) core in this chip is dedicated for video decoding and encoding. VPU embeds an XBurst® CPU core (named AUX in VPU) and application specified hardware accelerators for common video compress/decompress algorithms, which includes Stream Parser, Motion Compensation, Motion Estimation, Quant/Inverse Quant, DCT/Inverse DCT and De-block engines. Further more 3 route general purpose DMA enhances data management and transfer efficiency during video encoding/decoding.

XBurst® core's powerful programming agility combining with specified algorithm accelerators' high hotspot processing ability ensures VPU's multi format supporting and high performance ability. This distinctive structure brings us a nice trade-off of DSP's high power consumption and low processing ability with Hardware IP's complicated large logic size and limited format supporting.

Key standards performance of VPU in the chip:

- RealVideo decoding up to 1080P 30fps
- MPEG-2 decoding up to 1080P 30fps
- MPEG-4 decoding up to 1080P 30fps
- VC-1 decoding up to 1080P 30fps
- H.264 decoding up to 1080P 30fps
- H.264 encoding up to 720P 30fps

3.1 Block Diagram

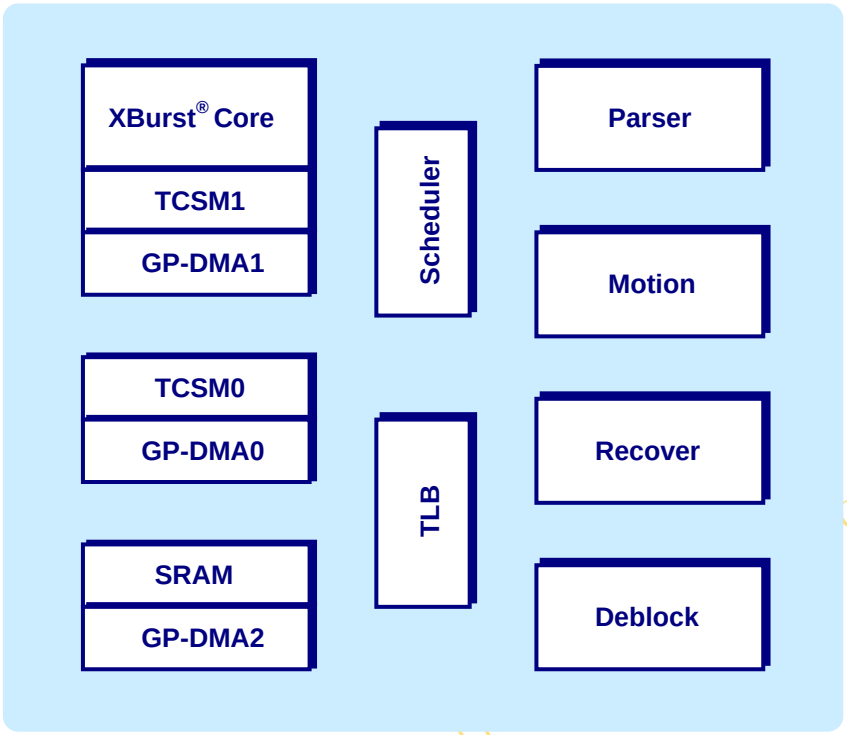


Figure 3-1 VPU Block Diagram

3.2 Features of VPU

Table 3-1 VPU Features

Item	Features
XBurst® core(AUX)	● XBurst-1 CPU – Industry standard RISC instruction set – 32 32-bit general purpose registers, no shadow GPR – Physical address accessing directly ● Media Extension Unit (MXU) – Ingenic SIMD instruction set II – fully pipelined
Tightly Coupled Sharing Memory (TCSM)	● TCSM0 – Size: 16K bytes – AHB slave interface supports external DMA access ● TCSM1 – Size: 48K bytes – AHB slave interface supports external DMA access **NOTE:** TCSM0 is coupled with J1 externally and serves as a memory interface for VPU, while TCSM1 is coupled with VPU XBurst® core internally.
Scratch RAM (SRAM)	● Size: 28K bytes ● AHB slave interface supports external DMA access
Gerenal Purpose DMA(GP_DMA)	● GP_DMA0/GP_DMA1/GP_DMA2 – Descriptor based DMA **NOTE:** GP_DMA0/GP_DMA1 is coupled with TCSM0/TCSM1 and GP_DMA2 is coupled with SRAM as well.
Parser (SDE)	Parser is a stream decode engine (named as SDE) in VPU ● Context adaptive binary arithmetic (CABAC) decoding support ● Context adaptive variable length (CAVLC) decoding support ● Programmable VLC table support for Gerenal Purpose VLC decoding accelerating
Motion (MCE)	Motion serves as a COMBO engine of compensation and estimation (named as MCE) in VPU ● Reference data cache embed ● Descriptor based task fetching ● Programmable processing size from 2x2 to 16x16 (in estimation the size is from 4x4 to 16x16) ● Programmable interplation filter from 2-tap to 8-tap ● Programmable sub-pixel accuracy from 1/2-pixel to 1/8-pixel (in estimation searching accuracy is supported from integer to 1/4-pixel)

	● Interlaced mode support ● Intensity compensation support ● Weighted prediction support ● Automatic rotation support for rotated referenced pictures ● Automatic expanding support for outside frame's reference ● Configurable searching strategy in estimation
Recover (VMAU)	Recover is a Matrix Arithmetic Unit in VPU (named as VMAU), it serves for pixel's recovery and reconstruction during vedio decoding and encoding. ● Descriptor based task fetching ● Configurable format intra prediction support ● Configurable format Inverse quant support ● Configurable format IDCT support ● Residual add for pixel's recovery ● Estimation subtract and pixel reconstruction in encoding flow
Deblock (DBLK)	● Descriptor based task fetching ● Dual-channel embeded (named as DBLK1 and DBLK2) ● RealVideo in loop filter support ● H.264 in loop filter support, MBAFF not support
Scheduler (SCH)	Scheduler is a special unit which is used to manage internel functional engines' handshake. ● 4 programable channels
Translation look-aside buffer (TLB)	● 8-entry based full associated ● Configurable page size

3.3 Internal physical address base definition

Table 3-2 VPU Internal physical address base definition

Module	Physical address base
AUX	0x132A_0000
TCSM0	0x132B_0000
GP_DMA0	0x1321_0000
TCSM1	0x132C_0000
GP_DMA1	0x1322_0000
SRAM	0x132F_0000
GP_DMA2	0x1323_0000
MCE	0x1325_0000
VMAU	0x1328_0000
DBLK1	0x1327_0000
DBLK2	0x132D_0000
SDE	0x1329_0000
SCH/TLB	0x1320_0000

3.4 AUX

3.4.1 Register Definition

3.4.1.1 Control and Status

AUX_CTRL **0x0**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SLEEP	Reserved																						BTB_INV	Reserved				MIRQ_EN	NMI_DIS	SW_NMI	SW_RST
RST	?	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	?	0	0	0	0	0	0	0	1

Bits	Name	Description	R/W
31	SLEEP	AUX sleep status. 1: sleep; 0: no sleep.	R
30:9	Reserved	Writing has no effect, read as zero.	R
8	BTB_INV	Writing 1 can invalid BTB. Writing 0 has no effect, read as zero.	W
7:4	Reserved	Writing has no effect, read as zero.	R
3	MIRQ_EN	1: enable message IRQ. 0: disable.	RW
2	NMI_DIS	1: NMI only wakes up AUX from sleep status 0: NMI wakes up AUX and switch PC to 0xF4000000	RW
1	SW_NMI	Nonmaskable IRQ (NMI). Writing 1 to the field triggers a NMI pulse to AUX. Writing 0 has no effect, read as zero.	W
0	SW_RST	Software reset. 1: let AUX keep at reset status; 0: do not reset.	RW

NOTES:

1 When NMI or IRQ or RESET exception occurs, AUX resumes from PC 0xF4000000.

2 When AUX wakes up by an NMI meanwhile NMI_DIS is 1, AUX just resumes from the next PC of the WAIT instruction.

3.4.1.2 SPINLOCK

AUX_SPINLK **0x4**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																														LOCK	
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	R/W
30:2	Reserved	Writing has no effect, read as zero.	R
1:0	LOCK	Lock status.	RW

AUX_SPIN1 **0x8**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																														SPIN1	
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	R/W
30:2	Reserved	Writing has no effect, read as zero.	R
1:0	SPIN1	Reading SPIN1 triggers following special hardware operations. First, value of AUX_SPINLK will be checked, if the value equals zero, the value of SPIN1 will overwrite AUX_SPINLK immediately, otherwise, AUX_SPINLK keeps unchanged. Then reading AUX_SPINLK instead of SPIN1 supplies the final read result. Writing SPIN1 is a normal write operation.	RW

AUX_SPIN2 **0xC**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																														SPIN2	
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	R/W
30:2	Reserved	Writing has no effect, read as zero.	R
1:0	SPIN2	The operations for SPIN1 also fit SPIN2 except the role of SPIN1 should be replaced by SPIN2.	RW

AUX_MIRQP **0x10**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																															MIRQP
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	R/W
31:1	Reserved	Writing has no effect, read as zero.	R
0	MIRQP	Pending status of MIRQ (message IRQ to CORE) which can only be set to 1 by HW and be reset to 0 by SW. This pending IRQ is routing to main CPU core.	RW

AUX_MSG **0x14**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MESG																															
RST	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	?	0	0

Bits	Name	Description	R/W
31:0	MESG	If AUX_CTRL.MIRQ_EN is value 1, writing the register raises an IRQ routing to the main CPU core meanwhile the AUX_MIRQP is set to 1 by HW automatically. The IRQ then keeps active until the register AUX_MIRQP is cleared to 0 by SW.	RW

CORE_MIRQP **0x18**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																															MIRQP
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	R/W
31:1	Reserved	Writing has no effect, read as zero.	R
0	MIRQP	Pending status of MIRQ (message IRQ to AUX) which can only be set 1 by HW and be clear to 0 by SW. This pending IRQ is routing to AUX.	RW

CORE_MSG **0x1C**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MESG																															
RST	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?

Bits	Name	Description	R/W
31:0	MESG	If AUX_CTRL.MIRQ_EN is value 1, writing the register raises an IRQ routing to the AUX. The IRQ then keeps active until the register CORE_MIRQP is cleared to 0 by SW.	RW

3.5 TCSM/SRAM

TCSM0/TCSM1/SRAM serves as the VPU control flow and data flow's communication between XBurst® CPU core with specified algorithm hardware accelerators and different hardware accelerators as well.

3.5.1 TCSM/SRAM space usage

Table 3-3 TCSM space usage

	XBurst® J1	XBurst® AUX	HW accelerator
TCSM0	0xF400_0000 ~ 0xF400_3FFF	0x132B_0000 ~ 0x132B_3FFF	0x132B_0000 ~ 0x132B_3FFF
TCSM1	0x132C_0000 ~ 0x132C_BFFF	0xF400_0000 ~ 0xF400_BFFF	0x132C_0000 ~ 0x132C_BFFF
SRAM	0x132F_0000 ~ 0x132F_6FFF	0x132F_0000 ~ 0x132F_6FFF	0x132F_0000 ~ 0x132F_6FFF

NOTES:

1 TCSM1/SRAM's space list for XBurst® J1 is physical address. In actual using it must be translated to its relative virtual address for XBurst® J1's accessing.

2 XBurst® J1 can not access SRAM with VPU internal masters simultaneously.

3.6 GP_DMA

3.6.1 Overview

GP_DMA is a 2-D data transfer DMA controller, which is tightly coupled with TCSM0/TCSM1/SRAM. Due to this tightly coupling, the data path for transferring should be limited as the following:

Table 3-4 GP_DMA data transfer path

GP_DMA	Validity of data transfer path
GP_DMA0	From other slavers except SRAM to TCSM0 is valid From TCSM0 to other slavers except SRAM is valid From TCSM0 to TCSM0 is forbiden
GP_DMA1	From other slavers except SRAM to TCSM1 is valid From TCSM1 to other slavers except SRAM is valid From TCSM1 to TCSM1 is forbiden
GP_DMA2	From other slavers to SRAM is valid From SRAM to other slavers is valid From SRAM to SRAM is forbiden

GP_DMA is working under descriptor-based configuration. Its descriptor node is defined as:

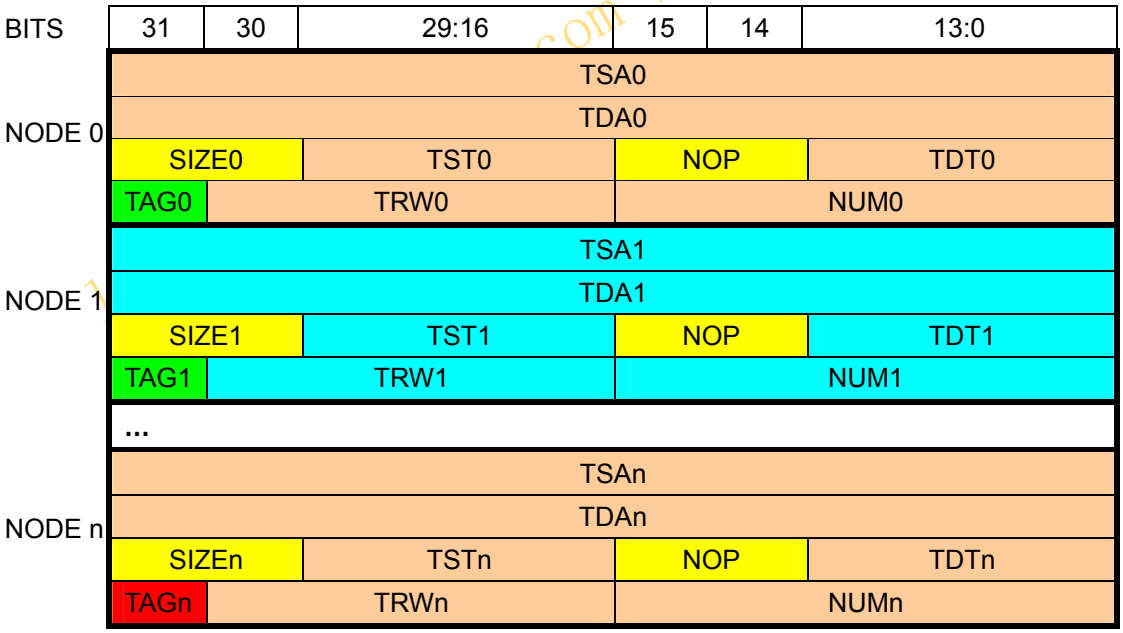


Figure 3-2 GP_DMA descriptor node structure

Table 3-5 GP_DMA descriptor node description

Item	Meaning
TSA	transfer source ADDRESS.
TDA	transfer destination ADDRESS.
TST	transfer source STRIDE.
TDT	transfer destination STRIDE.
TRW	transfer row WIDTH.
NUM	transfer byte NUMBER.
SIZE	transfer size type. 0: word 1: byte 2: half-word
TAG	Transfer link end tag. (GP_DMA parses each node to do data transfer and then go on parsing next adjacent node until it accomplishes a node with TAG equaling 1)

3.6.2 Register Definition

3.6.2.1 Descriptor Head Address (DHA)

DHA 0x0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DHA																														Reserved	
RST	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0

Bits	Name	Description	RW
31:2	DHA	Descriptor Head Address.	RW
1:0	Reserved	Writing has no effect, read as zero.	R

3.6.2.2 DMA Status/Command (DCS)

DCS 0x4

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BTN																NDN								Reserved					END	RST	SUP
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
31:16	BTN	Transfer number byte.	R
15:8	NDN	Transfer node number.	R
15~3	Reserved	Writing has no effect, read as zero.	R
2	END	0: GP_DMA in transferring 1: transmit end, GP_DMA is idle	R
1	RST	GP_DMA SW reset. GP_DMA would be reset when it was written as 1.	RW
0	SUP	GP_DMA startup.	RW

3.7 Video Acceleration Block

Please refer to relative programming manual documents.

4 GPU Core

4.1 Overview

Today's consumer devices feature rich, graphical user interfaces and run interactive applications like games and mobile web tools. GPU defines a family of high-performance cores that deliver hardware acceleration for 2D and 3D graphics displays on these devices. Addressable screen sizes range from the smallest cell phones to full HD 1080p displays.

GPU provides high performance, high quality graphics, low power consumption, and the smallest silicon footprint in every class. Dynamic power consumption is minimized by extensive use of multi-level hierarchical clock gating. The design also includes a 32-bit AHB interface, a 64-bit AXI interface, and support for virtual memory.

GPU accelerates numerous 2D and 3D graphics applications, including graphical user interfaces (GUI) and menu displays, Flash animation, and gaming, and it is a perfect fit for popular consumer devices like cell phones and smartphones, digital picture frames (DPF), digital signage, portable and in-dash GPS navigation systems, mobile internet devices (MID) and netbooks, handheld gaming consoles, set-top boxes, and HDTV.

An optimized software stack, complete software development tools, and a growing application ecosystem are supported by a robust graphics pipeline designed for industry-standard APIs, and with full support for Android, Linux, and Windows embedded development platforms. GPU supports the following graphics APIs:

- OpenGL ES 2.0
- OpenGL ES 1.1
- OpenVG 1.1
- DirectFB
- GDI/DirectDraw

4.2 Design Features

GPU includes a 32-bit AHB interface for register accesses and a 64-bit AXI interface for external memory accesses. It also includes virtual memory support. The following table describes the full feature set of GPU.

4.2.1 GPU Architecture Features

FEATURE	GPU Support
Primary API	OpenGL ES 1.1 and 2.0.
Additional APIs	OpenVG 1.1. DirectFB. GDI. DirectDraw.
Other graphics support	EGL 1.4.
Drivers	OpenGL ES 1.1 and 2.0. OpenVG 1.1. DirectFB. EGL. GDI/DirectDraw.
Operating systems	Windows CE. Linux. Embedded Android.
Z (depth)	Early Z support included.
Stencil	Early stencil support included.
Shader languages	GLSL ES 1.0.
Shader model compatibility	Shader model 3.0.
Shader types and execution units	One (1) programmable Scalable Ultra-threaded Unified Shader. (SIMD4:transcendental,ctl-flow,tx-load) One instruction issue per shader per clock; IEEE 32-bit floating-point pipeline supports long shader instructions.
FSAA anti-aliasing mechanisms	High quality MSAA 4x; MSAA 16x for OpenVG.
Code and data memory location restrictions	Unrestricted; arbitrary memory reads and writes.
Physical address	31 bits.
MMU description	32-bit virtual address; 4 kB pages, error reporting outside of address space.
TLB	4 cache lines per requestor.
Resource locks with CPU	Semaphore lock.
Max memory latency without a performance hit	128 GPU cycles.

4.2.2 GPU Command Processor Features

FEATURE	GPU Support
Command list structure	Linked memory buffer.
Branches	1-cycle; no penalty for dynamic branching.
GPU register access	AHB access to selected GPU registers.
GPU-CPU synchronization	Synchronization occurs via event queues.
Command buffering included in GPU IP	512 bytes; 64 words x 64 bits each.
Index buffer and vertex cache	512-byte index buffer; 1 kB vertex cache.
Set render state	One 32-bit register per cycle, 1-cycle throughput.
Set render target	One 32-bit register per cycle, 1-cycle throughput.
Set texture	One 32-bit register per cycle, 1-cycle throughput.
Set texture sampler	One 32-bit register per cycle, 1-cycle throughput.
Draw primitive	5-cycle throughput minimum; actual throughput depends on the number of vertices.
Draw indexed primitive	7-cycle throughput minimum; actual throughput depends on the number of vertices.
Counters	Variety of hardware counters for performance profiling.

4.2.3 Power Management Features

FEATURE	GPU Support
Low power CMOS technology compatible	Yes.
Automatic clock gating of flip flops and rams	Yes.
Global clock gating of unused macro blocks	Yes.
Software controlled effective clock frequency without changing the PLL	Yes.

4.2.4 GPU 2D Hardware Features

The features of the dedicated 2D unit are shown in the following table. These features include:

- Bit BLT and stretch BLT
- Rectangle fill and clear
- Line drawing
- High-performance stretch and shrink
- Monochrome expansion for text rendering
- ROP2, ROP3, ROP4
- Alpha blending including Java 2 Porter-Duff compositing blending rules
- 32k x 32k coordinate system
- 90, 180, and 270 degrees rotation
- Transparency by monochrome mask, chroma key, or pattern mask

FEATURE	GPU Support
Programmable Ops	ROP2, ROP3, ROP4 full alpha blending and transparency.
Fixed function	Line draw, Rectangle fill, Clear, Bit blit, Stretch blit, Filter blit.
Blit support	Copy (Bit), Filter, Monochrome Mask, Stretch/Shrink.
Source formats	RGBA4444/ 5551/ 8888, RGBX4444/ 5551/ 8888, RGB565, A8, UYVY(4:2:2), YUY2(4:2:2), YV12(4:2:0), 8-bit color index, NV12(4:2:0), NV16(4:2:2).
Destination formats	RGBA4444/ 5551/ 8888, RGBX4444/ 5551/ 8888, RGB565.
Alpha blending modes	Java2 Porter-Duff, Chroma Key, Pattern Mask.
Image scaling	Programmable high quality 9-tap, 32-phase filter.
Rotation	90 / 180 / 270 degrees on every 2D primitive.
Text rendering	Monochrome expansion; support for anti-aliased A8 fonts.
Alpha blend, scale, and rotation operations	Blending, scaling, and rotation are supported in one pass for stretch BLT.
Video	Video scaling and format conversion only.
Power for 2D vs. 3D doing 2D operations	Up to 90% less power required for dedicated 2D functions.
Rendering size	32k x 32k raster 2D coordinate system.

4.2.5 GPU 3D Hardware Features

The features of the GPU 3D unit are shown in the following table. These features include:

- OpenGL ES 2.0 compliance, including extensions; OpenGL ES 1.1; OpenVG 1.1
- IEEE 32-bit floating-point pipeline
- Ultra-threaded, unified vertex and fragment shaders
- Low bandwidth at both high and low data rates
- Low CPU loading
- Up to 12 programmable elements per vertex
- Dependent texture operation with high-performance
- Alpha blending
- Depth and stencil compare
- Support for 8 fragment shader simultaneous textures
- Support for 4 vertex shader simultaneous textures
- Point sampling, bi-linear sampling, tri-linear filtering, and cubic textures
- Resolve and fast clear
- 8k x 8k texture size and 8k x 8k rendering target

Unified vertex-fragment shader:

FEATURE	GPU Support
Shader type and execution units	Unified shader, SIMD4, SFP32 Trans.
Swizzle capabilities	Full 32-bit word level swizzle in a 128-bit vector.
GPR's per shader	Up to 512 general purpose registers, 128 bits each.
Uniform registers	Vertex Shader: 160 registers, 128 bits each. Fragment Shader: 64 registers, 128 bits each.
FP denorm and rounding options	Denorms are set to zero. Supports rounding to zero.
Maximum number of data input attributes	Maximum of 12 vertex shader input elements; maximum of 8 fragment shader input elements.
Maximum number of instructions	256 for vertex shaders; 256 for fragment shaders.
Maximum number of vertex streams	1.
Maximum number of threads in flight	256.
Subroutines	4 levels.
Conditional branch support	GT, LT, EQ, GE, LE, NE.
Shader instruction rate	1-cycle throughput for all shader instructions.
Floating-point instruction precision	Transcendental: 22 bits SIMD4 (vector): 23.5 bits.
Fragment shader video	Supports video texture.

Vertex Processing:

FEATURE	GPU Support
Vx D3D, OGL ES formats supported	BYTE, UBYTE, SHORT, USHORT, INT, UINT, DEC, UDEC, FLOAT, FLOAT16, D3DCOLOR, FIXED16DOT16.
Vertex data size limits	256 bytes.
Pre shader cache	1 Kb.
Post shader cache	8 vertices.

Primitive Processing:

FEATURE	GPU Support
Primitives supported	Triangle strip, fan, and list; line strip and list; point list.
Vertex/primitive geometry input index sizes	8-bit ,16-bit and 20-bit indices.
Setup parameters available to fragment shader	8 vec4 parameters; all available to fragment shader.

Texture Processing:

<table>
<tr><th>FEATURE</th><th>GPU Support</th></tr>
<tr><td>Fixed-point input texture formats</td><td>A8, L8, I8, A8L8, ARGB4, XRGB4, ARGB8, XRGB8, ABGR8, XBGR8, R5G6B5, A1RGB5, X1RGB5, YV12, YUY2, UYVY, D16, D24X8, A8_OES, DXT1, DXT2, DXT3, DXT4, DXT5, ETC1; all fixed-point formats are filtered.
<table>
<tr><th>Bits</th><th>Format</th><th>R</th><th>G</th><th>B</th><th>Alpha</th></tr>
<tr><td>16</td><td>ARGB4444</td><td>4</td><td>4</td><td>4</td><td>4</td></tr>
<tr><td>16</td><td>XRGB4444</td><td>4</td><td>4</td><td>4</td><td>4 don’t care</td></tr>
<tr><td>16</td><td>ARGB1555</td><td>4</td><td>4</td><td>4</td><td>1</td></tr>
<tr><td>16</td><td>XRGB1555</td><td>4</td><td>4</td><td>4</td><td>1 don’t care</td></tr>
<tr><td>16</td><td>RGB565</td><td>5</td><td>6</td><td>5</td><td>0</td></tr>
<tr><td>32</td><td>ARGB8888</td><td>8</td><td>8</td><td>8</td><td>8</td></tr>
<tr><td>32</td><td>XRGB8888</td><td>8</td><td>8</td><td>8</td><td>8 don’t care</td></tr>
<tr><td>32</td><td>ABGR8888</td><td>8</td><td>8</td><td>8</td><td>8</td></tr>
<tr><td>32</td><td>XBGR8888</td><td>8</td><td>8</td><td>8</td><td>8 don’t care</td></tr>
</table>
<table>
<tr><th>Planes</th><th>Format</th><th>Mode</th><th>Y</th><th>U</th><th>V</th><th>UV</th><th>YUYV</th><th>UYVY</th></tr>
<tr><td>3</td><td>YV12</td><td>4:2:0</td><td>1</td><td>1</td><td>1</td><td></td><td></td><td></td></tr>
<tr><td>2</td><td>NV12</td><td>4:2:0</td><td>1</td><td></td><td></td><td>1</td><td></td><td></td></tr>
<tr><td>1</td><td>YUY2</td><td>4:2:2:</td><td></td><td></td><td></td><td></td><td>1</td><td></td></tr>
<tr><td>1</td><td>UYVY</td><td>4:2:2</td><td></td><td></td><td></td><td></td><td></td><td>1</td></tr>
</table></td></tr>
<tr><td>Texture compression</td><td>4 bits and 8 bits per texel.</td></tr>
<tr><td>Compressed texture formats</td><td>DXT1, DXT2, DXT3, DXT4, DXT5, ETC1.
All compressed formats are filtered.</td></tr>
<tr><td>Texture size maximum</td><td>8k x 8k.</td></tr>
<tr><td>Addressing modes</td><td>wrap, mirror, clamp.</td></tr>
<tr><td>Mipmap support</td><td>14 mipmap levels; programmable LOD biasing & replacement.</td></tr>
<tr><td>Shadow texture</td><td>Depth texture PCF filtering.</td></tr>
<tr><td>Texture cache organization</td><td>Tiled, 4x4 texels.</td></tr>
<tr><td>Texture cache size</td><td>32 cache lines, with 64 bytes per cache line;
total of 2 kB texture cache.</td></tr>
<tr><td>Texture coordinate fraction bits</td><td>5 bits.</td></tr>
<tr><td>Texture sampler units</td><td>12 samplers, indexable.</td></tr>
<tr><td>Textures per fragment maximum</td><td>8 texture samplers.</td></tr>
<tr><td>Dependent texture operation</td><td>High performance; unlimited dependent texture reads.</td></tr>
<tr><td>Dependent tx per fragment max, relative sampling</td><td>No limit.</td></tr>
</table>

Texture repeat max	256.
Texture types	2D, cube map, 1D, projected, depth, bump map, displacement map.
Texture filters	Point sample, bi-linear, tri-linear.
Texture component mapping: D3D, OGL ES options	Supports both D3D and OES options.
Texture size types	Power-of-2, Non-square texture support.

Rasterization:

FEATURE	GPU Support
Interpolant attributes	8.
Render target size	8k x 8k.
Clipping window	Clipping rectangle supported.
Early Z	Yes.

Fragment Processing:

FEATURE	GPU Support
FSAA anti-aliasing mechanisms	High quality MSAA 4x; MSAA 16x for OpenVG.
Fragment color, alpha, Z, stencil precision	RGBA4444, RGBA5551, RGB565, RGBA8888, D16, D24, D24S8.
Fragment storage	16-bit color and Z, 32-bit color and Z for each fragment. Lossless compression, no storage reduction.
Alpha support	Individual fragment alpha masking.
Fragment cache	16 cache lines for color. 16 cache lines for Z. 64 bytes per cache line.

Dest/Alpha Blending:

FEATURE	GPU Support
Destination color formats	RGBA4444, RGBA5551, RGB565, RGBA8888.
Blend modes	Porter-Duff blending modes.
Dithering	Render target dithering support.

Z/Stencil Buffer:

FEATURE	GPU Support
Z/stencil formats	16-bit Z; 24-bit Z plus 8-bit stencil, with lossless compression support.
Z/stencil cache	16 cache lines; 64 bytes per line.

Stencil support	Both stencil and two-sided stencil.

Render Target:

FEATURE	GPU Support
Formats	16-bit and 32-bit, with lossless compression support.
RT buffer cache	16 cache lines; 64 bytes per line; RT caches are fully set associative.

5 DDR Controller

5.1 Overview

DDRC (DDR Controller) is a general IP which provide an interface to DDR2, DDR, mobile DDR memory. The DDRC IP is designed for SOC usage and is configurable, scalable to meet the requirement of various SOC.

Features:

- Support DDR2, DDR, mobile DDR (LPDDR) memory
- Support x16 and x32 external DDR data width
- Support clock frequency ratio – (BUS clock) : (DDR clock) = 2:1
- Support clock frequency ratio – (BUS clock) : (DDR clock) = 1:1
- Support clock-stop mode
- Support auto-refresh and self-refresh
- Support power-down mode and deep-power-down mode
- Programmable DDR timing parameters
- Programmable DDR row and column address width

5.1.1 Supported DDR SDRAM Types

In the following table, the DDR memory types in green are supported by DDRC.
Row address width 15-bit or more & Column width 11 or more are not supported.

64Mb			
Configuration	16Mb x 4	8Mb x 8	4Mb x 16
Number of Banks	4	4	4
Row address width	12	12	12
Column address width	10	9	8
128Mb			
Configuration	32Mb x 4	16Mb x 8	8Mb x 16
Number of Banks	4	4	4
Row address width	12	12	12
Column address width	11	10	9
256Mb			
Configuration	64Mb x 4	32Mb x 8	16Mb x 16
Number of Banks	4	4	4
Row address width	13	13	13
Column address width	11	10	9

512Mb			
Configuration	128Mb x 4	64Mb x 8	32Mb x 16
Number of Banks	4	4	4
Row address width	13	13	13
Column address width	12	11	10
1Gb			
Configuration	256Mb x 4	128Mb x 8	64Mb x 16
Number of Banks	4	4	4
Row address width	14	14	14
Column address width	12	11	10

5.1.2 Supported DDR2 SDRAM Types

In the following table, the DDR2 memory types in green are supported by DDRC.
All x4 (memory data width is 4-bit) devices are not supported.
Row address width 15-bit or more & Column width 11 or more are not supported.

256Mb			
Configuration	64Mb x 4	32Mb x 8	16Mb x 16
Number of Banks	4	4	4
Row address width	13	13	13
Column address width	11	10	9
512Mb			
Configuration	128Mb x 4	64Mb x 8	32Mb x 16
Number of Banks	4	4	4
Row address width	14	14	13
Column address width	11	10	10
1Gb			
Configuration	256Mb x 4	128Mb x 8	64Mb x 16
Number of Banks	8	8	8
Row address width	14	14	13
Column address width	11	10	10
2Gb			
Configuration	512Mb x 4	256Mb x 8	128Mb x 16
Number of Banks	8	8	8
Row address width	15	15	14
Column address width	11	10	10

5.1.3 Supported LPDDR SDRAM Types

In the following table, the LPDDR memory types in green are supported by DDRC.
Row address width 15-bit or more & Column width 11 or more are not supported.

128Mb			
Configuration		8Mb x 16	4Mb x 32
Number of Banks		4	-
Row address width		12	-
Column address width		9	-
256Mb			
Configuration		16Mb x 16	8Mb x 32
Number of Banks		4	4
Row address width		13	12
Column address width		9	9
512Mb			
Configuration		32Mb x 16	16Mb x 32
Number of Banks		4	4
Row address width		13	13
Column address width		10	9
1Gb			
Configuration		128Mb x 16	64Mb x 32
Number of Banks		8	8
Row address width		14	13
Column address width		10	10

5.1.4 Block Diagram

Following figure shows the functional block diagram of DDRC.

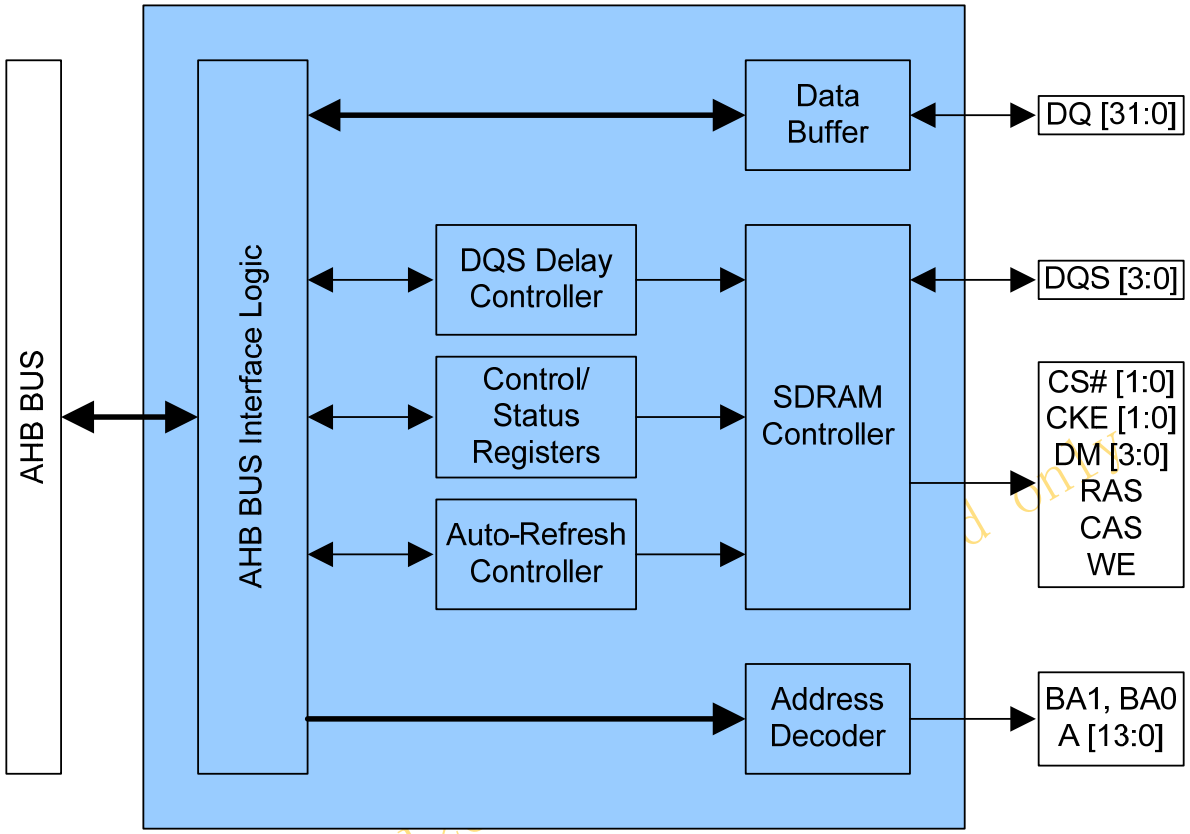


Figure 5-1 DDRC block diagram

5.2 Register Description

Table 5-1 DDRC Register lists the registers of DDR Controller. All of these registers are 32bit, and each bit of the register represents or controls one interrupt source that list in Table 5-1 DDRC Register.

All DDRC register 32bit access address is physical address.
The physical address base for the address-mapped registers of DDRC is 0x13020000.

Table 5-1 DDRC Register

Name	Address offset	Width	Access	Description
DSTATUS	0x00	32	RW	Status Register
DCFG	0x04	32	RW	DDR Configure Register
DCTRL	0x08	32	RW	DDR Control Register
DLMR	0x0C	32	RW	DDR Load-Mode-Register
DTIMING1	0x10	32	RW	DDR Timing Configure Register 1
DTIMING2	0x14	32	RW	DDR Timing Configure Register 2
DREFCNT	0x18	32	RW	Auto-Refresh Counter
DDQS	0x1C	32	RW	DDR DQS Delay Control Register
DDQSADJ	0x20	32	RW	DDR DQS Delay Adjust Register
DMMAP0	0x24	32	RW	DDR Memory CS0 Map Configure Register
DMMAP1	0x28	32	RW	DDR Memory CS1 Map Configure Register
DDELAYCTRL1	0x2C	32	RW	DDR Memory Delay Control Register1
DDELAYCTRL2	0x30	32	RW	DDR Memory Delay Control Register2
DSTRB	0x34	32	RW	Multi-media stride register
PMEMCTRL0	0x54	32	RW	IO pad control register
PMEMCTRL1	0x50	32	RW	IO pad control register

PMEMCTRL2	0x58	32	RW	IO pad control register
PMEMCTRL3	0x5C	32	RW	IO pad control register

5.2.1 DSTATUS

DSTATUS 0x13020000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																								ENDIAN	MISS	DPDN	PDN	AREF	SREF	Reserved	CKE0
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits 31~8: Reserved. Writing has no effect, read as zero.

ENDIAN: Read-only, indicate the data endian status.

Bit [7]	Description	Remark
0	Little data Endian.	(reset value)
1	Big data Endian.	

MISS: Indicate the bus memory-operation address out of DDRC memory mapping area. (this bit can be written)

Bit [6]	Description	Remark
0	No operation miss DDRC memory mapping.	(reset value)
1	At last one operation miss DDRC memory mapping.	

DPDN: Indicate the deep-power-down status of DDR memory.

Bit [5]	Description	Remark
0	DDR memory is NOT in deep-power-down state.	(reset value)
1	DDR memory is in deep-power-down state.	

PDN: Indicate the power-down status of DDR memory.

Bit [4]	Description	Remark
0	DDR memory is NOT in power-down state.	(reset value)
1	DDR memory is in power-down state.	

AREF: Indicate the auto-refresh status of DDR memory.

Bit [3]	Description	Remark
0	DDR memory is NOT in auto-refresh state.	(reset value)
1	DDR memory is in auto-refresh state.	

SREF: Indicate the self-refresh status of DDR memory.

Bit [2]	Description	Remark
0	DDR memory is NOT in self-refresh state.	(reset value)
1	DDR memory is in self-refresh state.	

CKE1: not support in this version.

Bit [1]	Description	Remark
0	CKE1 Pin is low.	(reset value)
1	CKE1 Pin is high.	

CKE0: Indicate the CKE0 Pin status of DDR memory.

Bit [0]	Description	Remark
0	CKE0 Pin is low.	(reset value)
1	CKE0 Pin is high.	

5.2.2 DCFG

Configure the external memory, once set; this register can NOT be changed on-the-fly.

DCFG 0x13020004

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved			ROW1			COL1		BA1	IMBA	DQSMD	BTRUN	Reserved				MISPE	TYPE			ROW0		COL0		CS1EN	CS0EN	CL				BA0	DW
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits 31~29, 20~16: Reserved. Writing has no effect, read as zero.

MISPE: Miss CS protect. Set 1 to enable.

If software read (or write) a memory space which is not select by any CS, this function will return random data to a read operation (or mask write operation) to avoid system bus be locked. A CS missing flag will set in DSTATUS.

DQSMD: Dqs pin mode. (only for inner test)

1: DQS pin with pull down resist

0: DQS pin without pull down resist

BTRUN: burst terminate enable. (only for mddr /ddr1)

1: enable

0: disable

IMBA:

0: CS0, CS1 connected 2 memory chips which has same ROW, COL, BA configuration.
In this mode, ROW,COL, BA configure both two chips. ROW1,COL1,BA1 are don't care

1: CS0, CS1 connected 2 memory chips which has different ROW, COL, BA configuration.
ROW, COL, BA refer to CS0; ROW1, COL1, BA1 refer to CS1

MEM_TYPE: Select external memory device type.

This field is not supported by current DDRC design.

Bit [14:12]	Description	Remark
000	Normal SDR (Single-Data-Rate) SDRAM(Not support).	(reset value)
001	Mobile SDR(Not support).	
010	Normal DDR1 (Double-Data-Rate) SDRAM.	
011	Mobile DDR.	
100	Normal DDR2.	
101	Mobile DDR2 (Not support).	
110	Normal DDR3 (Not support).	
111	Mobile DDR3 (Not support).	

ROW0/1: Row Address width. Specify the row address width of external DDR.

Bit [11:10]	Description	Remark
00	12-bit row address is used.	(reset value)
01	13-bit row address is used.	
10	14-bit row address is used.	
11	Reserved.	

COL0/1: Column Address width. Specify the Column address width of external DDR.

Bit [9:8]	Description	Remark
00	8-bit Column address is used.	(reset value)
01	9-bit Column address is used.	
10	10-bit Column address is used.	
11	11-bit Column address is used.	

CS1EN: DDR Chip-Select-1 Enable.

If there're ddr memory connected to ddr pin cs1, set CS1EN=1.

Bit [7]	Description	Remark
0	DDR Pin CS1 un-used.	(reset value)

1	There're DDR memory connected to CS1.	

CS0EN: DDR Chip-Select-0 Enable.

If there're ddr memory connected to ddr pin cs0, set CS0EN=1.

Bit [6]	Description	Remark
0	DDR Pin CS0 un-used.	(reset value)
1	There're DDR memory connected to CS0.	

CL: CAS Latency.

Bit [5:2]	Description	Remark
0,000	CL = 1 tCK.	(reset value)
0,001	CL = 1.5 tCK.	
1,001	CL = 2 tCK.	
0,010	CL = 2.5 tCK.	
1,010	CL = 3 tCK.	
0,011	CL = 3.5tCK.	
1,011	CL = 4 tCK.	
0,100	CL = 4.5 tCK.	
1,100	CL = 5 tCK.	Up to DDR2-533
Others	Reserved.	

Max frequency 533Mbps. If you use an high speed chips(for example DDR2-1066), you can still set CL value as DDR2-533.

BA0/1: Bank Address width of DDR memory.

Bit [1]	Description	Remark
0	4 bank device, Pin ba[1:0] valid, ba[2] un-used.	(reset value)
1	8 bank device, Pin ba[2:0] valid.	

DW: External DDR Memory Data Width.

Specify the external DDR memory data width.

Bit [0]	Description	Remark
0	External memory data width is 16-bit.	(reset value)
1	External memory data width is 32-bit.	

5.2.3 DCTRL

On the posedge of START, one command selected by CMD field will be performed.

DCTRL 0x13020008

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																ACTPD	PDT			ACTSTP	Reserved				DPD	SR	UNALIGN	ALH	Reserved	CKE	RESET	
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit 31~16, 10~7,2: Reserved. Writing has no effect, read as zero.

ACTSTP: Acitve Clock-Stop.

0: Clock can be stopped only after all banks be precharged

1: Clock can be stopped with some bank's row actived

ACTPD: Active Power-Down.

Some SDRAM devices support Active-Power-Down.

By default, ACTPD=0, hardware will precharge all active banks before entering Power-Down mode, so called Precharge-Power-Down.

By setting ACTPD=1, hardware drives SDRAM into Power-Down mode without precharge all active banks, some banks are still active in Power-Down mode, so called Active-Power-Down.

Bit [15]	Description	Remark
0	Precharge all banks before entering power-down.	(reset value)
1	Do not precharge all banks before entering power-down.	

PDT: Power-Down Timer.

When there's no access to DDR memory for a period of time, hardware drives DDR into power-down mode to save power consumption. Hardware can exit Power-Down mode automatically when new access arrives.

If PDT=0, power-down function disabled.

If use power-down, recommend to enable it after DDR initialization finished.

Bit [14:13]	Description	Remark
000	power-down disabled, hardware never drive SDRAM into power-down mode.	(reset value)
001	Enter power-down after 8 tCK idle.	
010	Enter power-down after 16 tCK idle.	
011	Enter power-down after 32 tCK idle.	
100	Enter power-down after 64 tCK idle.	
101	Enter power-down after 128 tCK idle.	

110 - 111	Reserved.	

SR: Software drive external DDR device entering Self-Refresh mode.

Software set SR=1 drive external DDR device entering self-refresh mode;

Software set SR=0 drive external DDR device exiting self-refresh mode;

In this mode, the CK to external DDR device would be stopped during self-refresh period;

But the clock supply to ddr_controller logic would not stop.

Software can read & write ddr_controller registers in this mode.

Software can NOT read or write memory data in this mode.

NOTE: Since ddr_controller registers are accessed via AXI bus interface, software must guarantee that there's no memory access during self-refresh mode. Otherwise, software can NOT exit this mode, **system would hangup!!**

Bit [5]	Description	Remark
0	Drive external DDR device entering self-refresh mode.	(reset value)
1	Drive external DDR device exiting self-refresh mode.	

DPD: Software drive external Mobile DDR device entering Deep-Power-Down mode.

Software set DPD = 1 drive external Mobile DDR device entering Deep-Power-Down mode instead of Power-Down mode, when there's no access to DDR memory for a period of time.

So you must first enable Power-Down mode (refer to PDT).

Software need to reset DDR controller and re-do a complete initial process to exit Deep-Power-Down mode.

When external device go to Deep-Power-Down mode, it will lose all data store in memory and registers.

The memory chip will disable inner power support to save power.

UNALIGN: Enable unaligned transfer on AXI BUS.

Bit [4]	Description	Remark
0	Disable unaligned transfer on AXI BUS.	(reset value)
1	Enable unaligned transfer on AXI BUS.	

ALH: Advanced Latency Hiding.

This is a test purpose register.

Some latency timings can be hidden in special cases.

Bit [3]	Description	Remark
0	Disable ALH.	(reset value)
1	Enable ALH.	

CKE: Control the status of CKE pin.

Write CKE=1 can set CKE pin to HIGH state.

Write CKE=0 would be ignored.

The default value of CKE Pin is low;

CKE0,1 Pins status is represented by DDR_STATUS register.

Caution: This register is used only for DDR initializing sequence; software can NOT update this register when DDR memory is in normal working mode.

Bit [1]	Description	Remark
0	Not set CKE Pin High.	(reset value)
1	Set CKE Pin HIGH.	

RESET: Module reset for ddr_controller.

Software reset ddr_controller by setting RESET bit high. Then, software end reset by setting RESET bit low.

Bit [0]	Description	Remark
0	End resetting ddr_controller.	(reset value)
1	Resetting ddr_controller.	

5.2.4 DLMR

DLMR register is used for initializing the DDR SDRAM memory device.

On the posedge of START, one command selected by CMD field will be performed.

DLMR **0x1302000C**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved								DDR_ADDR								Reserved					BA			Reserved		CMD		Reserved			START
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit 31~30, 15~11, 7~6, 3~1: Reserved. Writing has no effect, read as zero.

DDR_ADDR: When performing a DDR command, DDR_ADDR[13:0] corresponding to external DDR address Pin A[13:0]; DDR_ADDR[15:14] are reserved.

Bit [29:16]	Description	Remark
0000_0000	corresponding to external DDR address Pin A[13:0].	(reset value)

BA: Bank Address.

When performing a DDR command, BA[2:0] corresponding to external DDR address Pin BA[2:0].

Bit [10:8]	Description	Remark
000	corresponding to external DDR address Pin BA[2:0].	(reset value)

CMD: Select command to process when setting START from low to high.
On the posedge of START, one of the following commands will be performed.

Bit [5:4]	Description	Remark
00	Precharge one bank / All banks. (dependent field : BA, DDR_ADDR)	(reset value)
01	Auto-Refresh.	
10	Load Mode Register. (dependent field : BA, DDR_ADDR)	
11	Reserved.	

START: Start perform a command to external DDR memory.
The command is performed on the posedge of START; Hardware will clear START bit to zero when command issued out to external DDR memory.
Write 0 to START will be ignored and take no effect;
START=1 means hardware is busy executing current command and can NOT accept new command;
Software must check START=0 before writing 1 to START.

Bit [0]	Description	Remark
0	No command is performed.	(reset value)
1	On the posedge of START, perform a command defined by CMD field.	

5.2.5 DTIMING1,2 (DDR Timing Config Register 1, 2)

The timing parameters are identical to the JEDEC DDR Specification.

DTIMMING1 **0x13020010**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	tRAS				Reserved		tRTP		Reserved		tRP		Reserved	tRCD			tRC				Reserved		tRRD		Reserved	tWR			Reserved		tWTR	
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits 27~26, 23, 19, 11~10, 7, 3-2: Reserved. Writing has no effect, read as zero.
tRAS: ACTIVE to PRECHARGE command period.
tRAS defines the ACTIVE to PRECHARGE command period to the same bank.

Bit [31:28]	Description	Remark
0000	1 tCK.	(reset value)
0001	3 tCK.	
0010	5 tCK.	
0011	7 tCK.	
… …	**… 2 * tRAS + 1 …**	
1101	27 tCK.	
1110	29 tCK.	
1111	31 tCK.	

tRTP: READ to PRECHARGE command period.

Bit [25:24]	Description	Remark
00	1 tCK.	(reset value)
01	2 tCK.	
10	3 tCK.	
11	4 tCK.	

tRP: PRECHARGE command period.

tRP defines the PRECHARGE to next command period to the same bank.

Bit [22:20]	Description	Remark
000	1 tCK.	(reset value)
001	2 tCK.	
010	3 tCK.	
011	4 tCK.	
100	5 tCK.	
101	6 tCK.	
110	7 tCK.	
111	8 tCK.	

tRCD: ACTIVE to READ or WRITE command period.

tRCD defines the ACTIVE to READ/WRITE command period to the same bank.

Bit [18:16]	Description	Remark
000	1 tCK.	(reset value)
001	2 tCK.	
010	3 tCK.	
011	4 tCK.	
100	5 tCK.	
101	6 tCK.	
110	7 tCK.	

111	8 tCK.	

tRC: ACTIVE to ACTIVE command period.

tRC defines the ACTIVE to ACTIVE command period to the same bank.

Since tRCD + read/write-time + tRP > tRC always match, in most cases, tRC can be disabled.

Bit [15:12]	Description	Remark
0000	1 tCK.	(reset value)
0001	3 tCK.	
0010	5 tCK.	
0011	7 tCK.	
… …	**… 2 * tRC + 1 …**	
1101	27 tCK.	
1110	29 tCK.	
1111	31 tCK.	

tRRD: ACTIVE bank A to ACTIVE bank B command period.

tRRD defines the ACTIVE to ACTIVE command period to **different** banks.

Bit [9:8]	Description	Remark
00	Disable tRRD counter.	(reset value)
01	2 tCK.	
10	3 tCK.	
11	4 tCK.	

tWR: WRITE Recovery Time defined by register MR of DDR2 memory.

Bit [6:4]	Description	Remark
000	1 tCK.	(reset value)
001	2 tCK.	
010	3 tCK.	
011	4 tCK.	
100	5 tCK.	
101	6 tCK.	
110 - 111	Reserved.	

tWTR: WRITE to READ command delay.

Bit [1:0]	Description	Remark
00	1 tCK.	(reset value)
01	2 tCK.	

10	3 tCK.	
11	4 tCK.	

DTIMING2 **0x13020014**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved			tRFC						Reserved		tRWCOV		tCKE			Reserved				tMINSR				Reserved		tXP			Reserved		tMRD	
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits 31~30, 23~19, 15~12, 7, 3~2: Reserved. Writing has no effect, read as zero.

tRWCOV: in common, set this value equal to (DDELAYCTRL1.Tsel[1:0]) -1.If Tsel = 0, set tRWCOV to 0 too.

tCKE: minimum CKE pulse width.

tCKE define the minimum CKE pulse width, include high level and low level.

Bit [18:16]	Description	Remark
000	1 tCK.	(reset value)
001	2 tCK.	
010	3 tCK.	
011	4 tCK.	
100	5 tCK.	
101	6 tCK.	
110	7 tCK.	
111	8 tCK.	

tRFC: AUTO-REFRESH command period.

tRFC defines the minimum delay after an AUTO-REFRESH command. During tRFC period, no command can be issued to DDR memory.

Delay Time = **2 * tRFC + 1.**

Bit [29:24]	Description	Remark
000000	1 tCK.	(reset value)
000001	3 tCK.	
000010	5 tCK.	
000011	7 tCK.	
… …	**… 2 * tRFC + 1 …**	
111101	125 tCK.	
111110	127 tCK.	
111111	129 tCK.	

* tCK – one DDR memory clock cycle, typical tCK value is 7.5 ns (133MHz clock).

tMINSR: Minimum Self-Refresh / Deep-Power-Down time.

After DDR memory turns into Self-Refresh or Deep-Power-Down mode, it will NOT exit until tMINSR condition meets.

Delay Time = tMINSR * 8 + 1.

Bit [11:8]	Description	Remark
0000	1*8 + 1 tCK.	(reset value)
0001	2*8 + 1 tCK.	
0010	3*8 + 1 tCK.	
0011	4*8 + 1 tCK.	
… …	**… tMINSR * 8 + 1 …**	
1101	14*8 + 1 tCK.	
1110	15*8 + 1 tCK.	
1111	16*8 + 1 tCK.	

tXP: EXIT-POWER-DOWN to next valid command period.

tXP defines the EXIT-POWER-DOWN to next valid command period to all banks.

Bit [6:4]	Description	Remark
000	1 tCK.	(reset value)
001	1 tCK.	
010	2 tCK.	
011	3 tCK.	
100	4 tCK.	
101	5 tCK.	
110	6 tCK.	
111	7 tCK.	

tMRD: Load-Mode-Register to next valid command period.

tMRD defines the Load-Mode-Register to next valid command period.

Bit [1:0]	Description	Remark
00	1 tCK.	(reset value)
01	2 tCK.	
10	3 tCK.	
11	4 tCK.	

5.2.6 DREFCNT (DDR Auto-Refresh Counter)

DREFCNT 0x13020018

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved								CON								CNT								Reserved				CLK_DIV		REF_EN	
RST	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits 31~24, 7-4: Reserved. Writing has no effect, read as zero.

CON: A constant value used to compare with the CNT value.

After reset, CON=0xFF and CNT=0x00;

It is not recommended to set CON=0x00.

CNT: 8-bit counter; When the value of CNT match the value of CON, flag bit EQU is set high and an auto-refresh command will be issued to DDR memory. READ only.

CLK_DIV : Clock Divider.

Divide the dclk to generate a lower frequency of clock to drive the auto-refresh counter. This helps to save power consumption.

Set CLK_DIV=0 can disable the clock to auto-refresh counter.

When the DDR memory is in self-refresh mode or in deep-power-down mode, disable the clock of auto-refresh counter can save power consumption. Future more, the module clock to DDRC can also be stopped.

dclk is CKO clock, When ddr work in 500Mbps, dclk is 250Mhz.

Bit [3:1]	Description	Remark
000	dclk / 16.	(reset value)
001	dclk / 32.	
010	dclk / 64.	
011	dclk / 128.	
100	dclk / 256.	
101	dclk / 512.	
110	dclk / 1024.	
111	--	

REF_EN: Enable Refresh Counter.

Software set REF_EN=1 right after initialize ddr memory.

Bit [29]	Description	Remark
0	Enable auto-refresh counter.	(reset value)
1	Disable auto-refresh counter.	

5.2.7 DDQS (DDR DQS Delay Control Register)

DDRC contains an on-chip DLL to control the DQS Delay for read data and write data.

DDQS **0x1302001C**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved		ERROR	READY	Reserved		SRDET	DET	AUTO	Reserved	CLKD						Reserved		WDQS						Reserved		RDQS					
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits 31~30, 27-26, 22, 15-14, 7-6: Reserved. Writing has no effect, read as zero.

ERROR: ahb_clk Delay Detect ERROR, read-only.

When hardware detect one ahb_clk cycle delay failed, ERROR is set high;

ERROR is cleared zero when a new detection starts;

ERROR is valid only when READY=1.

Bit [29]	Description	Remark
0	delay detect success.	(reset value)
1	delay detect failed.	

READY: ahb_clk Delay Detect READY, read-only.

When hardware detect complete, ERROR is set high.

READY is cleared zero when a new detection starts.

Bit [28]	Description	Remark
0	delay detect NOT complete.	(reset value)
1	delay detect complete.	

SRDET: DDRC auto re-detect and set (if auto = = 1) delay line after clock change.

It will consume extra times in clock change process.

Bit [25]	Description	Remark
0	not enable.	(reset value)
1	Enable.	

AUTO: Hardware auto-detect & set delay line.

Bit [23]	Description	Remark
0	Hardware do NOT auto-set delay line.	(reset value)
1	Hardware auto-set delay line after detect success.	

DET: Start delay detecting.

Write 1 to START bit starts a new delay detect progress.
When delay detect complete, START is cleared zero by hardware.
START can be used as the BUSY flag. When START=1, it is busy.

Bit [24]	Description	Remark
0	No operation.	(reset value)
1	Delay detect in progress, busy.	

CLKD: Indicate the number of delay elements needed to delay ¼ tCK.
CLKD is a reference value for setting WDQS and RDQS.
CLKD is set when DLL detection finished.
The range of CLKD: [0, +63].

WDQS: Set the number of delay elements used on the write DQS delay-line.
When WDQS increase one, the delay value of write DQS increase approximately 0.1 ns .
The range of WDQS, RDQS: [0, +63].
NOTE: The delay value of each delay element depends on the technology and the structure of the delay cell; "0.1 ns" is just an example at .18 technology.

RDQS: Set the number of delay elements used on the read DQS delay-line.
When RDQS increase one, the delay value of read DQS increase approximately 0.1 ns.

5.2.8 DDQSADJ (DDR DQS Delay Adjust Register)

DDQSADJ **0x13020020**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DQSCON																Reserved		WSIGN	WDQS							Reserved		RSIGN	RDQS				
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits 15~14, 7-6: Reserved. Writing has no effect, read as zero.
DQSCON: DQS detect looping counter threshold. When inner counter equal to DQSCON, then trigger a DQS detect operation to auto adjust DQS delay line.This inner counter use AUTO_REFRESH's divided clock (refer to DREFCNT). When DQSCON set to 0, this function be disabled.
WSIGN, RSIGN: The adjust value's sign. 0: plus; 1: minus.
WDQS, RDQS: The adjust value for WRITE and READ DQS delay.
WDQS, RDQS can be either positive or negative number.
For negative number, it should be in "complemental code" format;
The range of WDQS, RDQS : [-16, +15].

For READ: DQS_Delay = DDQS.RDQS +/- DDQSADJ.RDQS.
For WRITE: DQS_Delay = DDQS.WDQS +/-DDQSADJ.WDQS.

5.2.9 DMMAP0,1 (DDR Memory Map Config Register)

The physical base address and size of external DDR Memory can be configured by DMMAP register. The size of external DDR Memory must be: **2^(24+n), n=0, 1, 2, 3, …**.

When the following equation is met:

(AXI_BUS_Address[31:24] & MASK[7:0]) == BASE

The DDR Memory is selected.

DMMAP0,1 **0x13020024, 0x13020028**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																BASE								MASK							
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0

Bits 31~16: Reserved. Writing has no effect, read as zero.

BASE: base address.

MASK: address mask.

Examples:

1 DDR address space in system memory : 0x2000_0000 ~ 0x2FFF_FFFF (256MB)
BASE=0x20 MASK=0xF0.

2 DDR address space in system memory : 0x5000_0000 ~ 0x57FF_FFFF (128MB)
BASE=0x50 MASK=0xF8.

NOTE: If DDRC is disabled, please set DMMAP=0x0000_FF00 (reset value).

5.2.10 DDELAYCTRL

This register can be re-configured at any time, but this change takes effect after an Auto-Refresh command occurs.

DDELAYCTRL1 **0x1302002C**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved												TSEL		MSEL		HL	QUAR	Reserved							MAUTO	MSIGN	MASK_DELAY_SEL_ADJ						
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDELAYCTRL2 **0x13020030**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																										MASK_DELAY_SEL						
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Reserved: Writing has no effect, read as zero.

MSEL[1:0], HL, QUAR use to adjust the position of DQS mask for DDR PHY.

MSEL: Mask delay selection.

Bit [17:16]	Description	Remark
00	No delay.	(reset value)
01	delay 1 tCK.	
10	delay 2 tCK.	
11	delay 3 tCK.	

HL: Half clock delay selection.

Adjust MSEL delay 1/2 tCK.

0: delay no change

1: delay reduced 1/2 tCK

QUAR: Quarter clock delay selection.

Adjust MSEL delay 1/4 tCK.

0: delay no change

1: delay add 1/4 tCK

Msel[1]	Msel[0]	HL	QUAR	DELAY
0	0	1	0	- 0.5 tCK
0	0	1	1	- 0.25 tCK
0	0	0	0	0 tCK
0	0	0	1	0.25 tCK
0	1	1	0	0.5 tCK
0	1	1	1	0.75 tCK
0	1	0	0	1 tCK
0	1	0	1	1.25 tCK
1	0	1	0	1.5 tCK
1	0	1	1	1.75 tCK
1	0	0	0	2 tCK
1	0	0	1	2.25 tCK
1	1	1	0	2.5 tCK
1	1	1	1	2.75 tCK
1	1	0	0	3 tCK
1	1	0	1	3.25 tCK

TSEL: Read delay selection.

For transferring read data from PHY to DDR controller.

Bit [19:18]	Description	Remark
00	No delay.	(reset value)
01	delay 1 Tck.	
10	delay 2 tCK.	
11	delay 3 tCK, only use when CL < 5	

MASK_DELAY_SEL_ADJ , MASK_DELAY_SEL, MSIGN:

Internal use. To adjust QUAR.

MAUTO:

Enable inner mask delay function. In normal keep this bit to 1.

5.2.11 DSTRB

DSTRB **0x13020034**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved				STRB0												Reserved				STRB1											
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Internal use.

Just for Video Decoder.

5.2.12 DDR PAD CONTROLL REGISTER 0

PMEMCTRL0 **0x13020054**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ODTDQS3		ODTDQS2		ODTDQS1		ODTDQS0		ODTDQ3		ODTDQ2		ODTDQ1		ODTDQ0		SSELDQS3		SSELDQS2		SSELDQS1		SSELDQS0		SSELDQ3		SSELDQ2		SSELDQ1		SSELDQ0	
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0

Bits	Name	Description	RW
31:30	ODTDQS3	ODT configure for DQS3.	RW
29:28	ODTDQS2	ODT configure for DQS2.	RW
27:26	ODTDQS1	ODT configure for DQS1.	RW
25:24	ODTDQS0	ODT configure for DQS0.	RW
23:22	ODTDQ3	ODT configure for DQ3.	RW
21:20	ODTDQ2	ODT configure for DQ1.	RW

19:18	ODTDQ1	ODT configure for DQ1.	RW
17:16	ODTDQ0	ODT configure for DQ0.	RW
15:14	SSELDQS3	Output mode & strength select for DQS[3].	RW
13:12	SSELDQS2	Output mode & strength select for DQS[2].	RW
11:10	SSELDQS1	Output mode & strength select for DQS[1].	RW
9:8	SSELDQS0	Output mode & strength select for DQS[0].	RW
7:6	SSELDQ3	Output mode & strength select for DQ[31:24].	RW
5:4	SSELDQ2	Output mode & strength select for DQ[23:16].	RW
3:2	SSELDQ1	Output mode & strength select for DQ[15:8].	RW
1:0	SSELDQ0	Output mode & strength select for DQ[7:0].	RW

5.2.13 DDR PAD CONTROLL REGISTER 1

PMEMCTRL1 **0x13020050**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved												ODTCK		ODTCKE		ODTADDR		ODTDM3		ODTDM2		ODTDM1		ODTDM0		ODTCMD		ODTCS1		ODTCS0	
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
19:18	ODTCK	ODT configure for CK.	RW
17:16	ODTCKE	ODT configure for CKE.	RW
15:14	ODTADDR	ODT configure for ADDR.	RW
13:12	ODTDM3	ODT configure for DM3.	RW
11:10	ODTDM2	ODT configure for DM2.	RW
9:8	ODTDM1	ODT configure for DM1.	RW
7:6	ODTDM0	ODT configure for DM0.	RW
5:4	ODTCMD	ODT configure for CMD.	RW
3:2	ODTCS1	ODT configure for CS1.	R
1:0	ODTCS0	ODT configure for CS0.	RW

ODT[1:0]	ODT Rtt
00	disable ODT
01	75 ohm
10	150 ohm
11	Reserved

5.2.14 DDR PAD CONTROLL REGISTER 2

This register is used to select strength of SSTL18, SSTL2, MDDR and LVTTL combo single-end/differential transmitter.

PMEMCTRL2 **0x13020058**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved												SSELCK		SSELCKE		SSELADDR		SSELDM3		SSELDM2		SSELDM1		SSELDM0		SSELCMD		SSELCS1		SSELCS0	
RST	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0

Bits	Name	Description	RW
31:20	Reserved	Writing has no effect, read as zero.	R
19:18	SSELCK	Output mode & strength select for CKO.	RW
17:16	SSELCKE	Output mode & strength select for CKE.	RW
15:14	SSELADDR	Output mode & strength select for ADDR[16:0].	RW
13:12	SSELDM3	Output mode & strength select for DM3.	RW
11:10	SSELDM2	Output mode & strength select for DM2.	RW
9:8	SSELDM1	Output mode & strength select for DM1.	RW
7:6	SSELDM0	Output mode & strength select for DM0.	RW
5:4	SSELCMD	Output mode & strength select for CMD(RAS, CAS,WE).	RW
3:2	SSELCS1	Output mode & strength select for CS1.	RW
1:0	SSELCS0	Output mode & strength select for CS0.	RW

SSEL configure:

MODE	POWER	SSEL[1:0]			
		Reduced strength		Full strength	
DDR1	2.5v	10(CLASS II)		00(CLASS I)	
DDR2	1.8v	10		00	
MDDR	1.8v	11 (2mA)	10 (4mA)	01(8mA)	00 (10mA)

5.2.15 DDR PAD CONTROLL REGISTER 3

This register is used to select output enable signal (low active) of SSTL18, SSTL2, MDDR and LVTTL combo single-end transceiver.

PMEMCTRL3 **0x1302005C**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PDDQS3	PDDQS2	PDDQS1	PDDQS0	PDDQ3	PDDQ2	PDDQ1	PDDQ0	Reserved						SSTL2	LVCMOS	SINGLEDQS	OENCK	OENBA2	OENBA1	OENBA0	OENA13	OENA12	OENA11_0	OENDM3	OENDM2	OENDM1	OENDM0	OENCMD	OENCS1	OENCS0	OENCKE
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
31	PDDQS3	input enable signal for CKO. 0: enable; 1: disable.	RW
30	PDDQS2	input enable signal for CKO. 0: enable; 1: disable.	RW
29	PDDQS1	input enable signal for CKO. 0: enable; 1: disable.	RW
28	PDDQS0	input enable signal for CKO. 0: enable; 1: disable.	RW
27	PDDQ3	input enable signal for CKO. 0: enable; 1: disable.	RW
26	PDDQ2	input enable signal for CKO. 0: enable; 1: disable.	RW
25	PDDQ1	input enable signal for CKO. 0: enable; 1: disable.	RW
24	PDDQ0	input enable signal for CKO. 0: enable; 1: disable.	RW
23:18	Reserved	Writing has no effect, read as zero.	R
17	SSTL2	SSTL2 select pin. 0: SSTL2(DDR1); 1:SSTL18(DDR2).	RW
16	LVCMOS	LVCMOS select pin. 0: not LVCMOS; 1: LVCMOS(MDDR).	RW
15	SINGLEDQS	single end DQS 0: differential DQS.	RW
14	OENCK	output enable signal for CKO. 0: enable; 1: disable.	RW
13	OENBA2	output enable signal for BA[2] (ADDR[16]). 0: enable; 1: disable.	RW
12	OENBA1	output enable signal for BA[1] (ADDR[15]). 0: enable; 1: disable.	RW
11	OENBA0	output enable signal for BA[0] (ADDR[14]). 0: enable; 1: disable.	RW
10	OENA13	output enable signal for ADDR[13]. 0: enable; 1: disable.	RW
9	OENA12	output enable signal for ADDR[12]. 0: enable; 1: disable.	RW
8	OENA11_0	output enable signal for ADDR[11:0]. 0: enable; 1: disable.	RW
7	OENDM3	output enable signal for DM3. 0: enable; 1: disable.	RW
6	OENDM2	output enable signal for DM2. 0: enable; 1: disable.	RW
5	OENDM1	output enable signal for DM1. 0: enable; 1: disable.	RW
4	OENDM0	output enable signal for DM0. 0: enable; 1: disable.	RW
3	OENCMD	output enable signal for CMD(RAS,CAS,WE). 0: enable; 1: disable.	RW
2	OENCS1	output enable signal for CS1. 0: enable; 1: disable.	RW
1	OENCS0	output put enable signal for CS0. 0: enable; 1: disable.	RW
0	OENCKE	output put enable signal for CKE. 0: enable; 1: disable.	RW

5.2.16 DDRMPORT

DDRMPORT **0x13020060**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CH5G	CH4G	CH3G	CH2G	CH1G	Reserved	CH4LKEN	CH3LKEN	Reserved		IPUUG	Reserved				CH1RDFST	CH5PRIEN	CH4PRIEN	CH3PRIEN	CH2PRIEN	CH1PRIEN	Reserved	CH5PRI		CH4PRI		CH3PRI		CH2PRI		CH1PRI	
RST	0	0	1	0	0	0	1	1	0	0	1	0	0	0	0	1	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
31	CH5G	Channel 5 ‘s read will be blocked by other channels write at same address. 0: blocked; 1: not blocked.	RW
30	CH4G	Channel 4 ‘s read will be blocked by other channels write at same address. 0: blocked; 1: not blocked.	RW
29	CH3G	Channel 3 ‘s read will be blocked by other channels write at same address. 0: blocked; 1: not blocked.	RW
28	CH2G	Channel 2 ‘s read will be blocked by other channels write at same address. 0: blocked; 1: not blocked.	RW
27	CH1G	Channel 1‘s read will be blocked by other channels write at same address. 0: blocked; 1: not blocked.	RW
26	Reserved	Writing has no effect, read as zero.	R
25	CH4LKEN	Channel 4 support lock function. 0: not support; 1: support.	RW
24	CH3LKEN	Channel 3 support lock function. 0: not support; 1: support.	RW
23:22	Reserved	Writing has no effect, read as zero.	R
21	IPUUG	1: IPU read will never block by other channels write at same address. 0: IPU read will block by other channels write at same address.	RW
20:17	Reserved	Writing has no effect, read as zero.	R
16	CH1RDFST	Channel 1 support read first than write function. 0: not support; 1: support.	RW
15	CH5PRIEN	Channel 5 support bus priority function. 0: use CH5PRI instead; 1: use bus priority.	RW
14	CH4PRIEN	Channel 4 support bus priority function. 0: use CH4PRI instead; 1: use bus priority.	RW
13	CH3PRIEN	Channel 3 support bus priority function. 0: use CH3PRI instead; 1: use bus priority.	RW
12	CH2PRIEN	Channel 2 support bus priority function. 0: use CH2PRI instead; 1: use bus priority.	RW
11	CH1PRIEN	Channel 1 support bus priority function. 0: use CH1PRI instead; 1: use bus priority.	RW
10	Reserved	Writing has no effect, read as zero.	R
9:8	CH5PRI	set channel 5’s priority if CH5PRIEN = 0 (3-hightest ~ 0-lowest).	RW

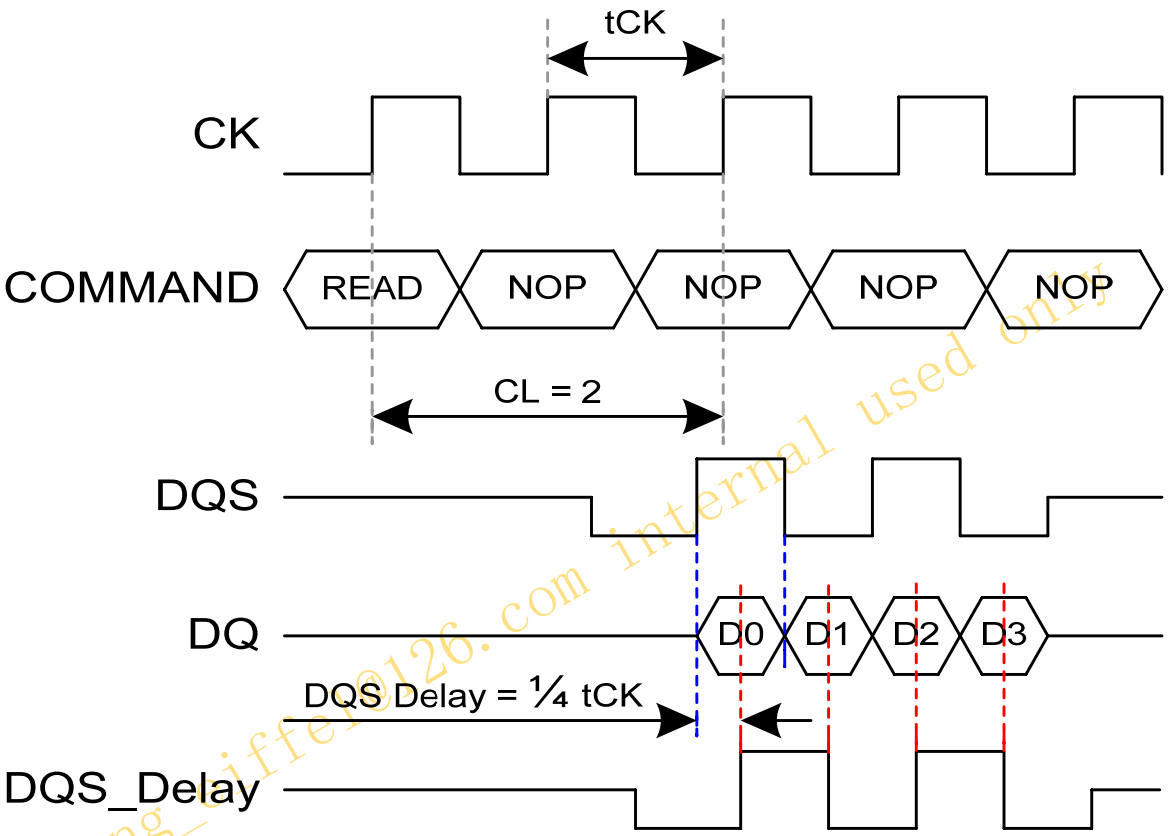
7:6	CH4PRI	set channel 4's priority if CH4PRIEN = 0 (3-hightest ~ 0-lowest).	RW
5:4	CH3PRI	set channel 3's priority if CH3PRIEN = 0 (3-hightest ~ 0-lowest).	RW
3:2	CH2PRI	set channel 2's priority if CH2PRIEN = 0 (3-hightest ~ 0-lowest).	RW
1:0	CH1PRI	set channel 1's priority if CH1PRIEN = 0 (3-hightest ~ 0-lowest).	RW

5.3 Functional Description

5.3.1 DDR DQS Delay Detect-and-Set Processing

Sub-module “DQS Delay Controller” of DDRC generates DQS_Delay signal to capture data;

The following figure illustrates the DQS_Delay in READ case. The DQS_Delay in WRITE case is similar to READ case.



DQS_Delay is used to capture DQ by its posedge and negedge. DQS_Delay signal is generated by the “DQS Delay Controller” sub-module of DDRC.

There’re delay elements in “DQS Delay Controller”; each delay element adds approximately 0.1 ns delay value to between its input and output. The number of delay elements used can be controlled by DDQS.RDQS and DDQS.WDQS.

Note that the delay value of each delay element changes according to the temperature and voltage. It is recommended to adjust the value of DDQS.RDQS and DDQS.WDQS periodically according to the

temperature change.

The “DQS Delay Controller” provides a mechanism to automatically adjust the DDQS value periodically. Alternatively, this function can be disabled to enable fully manual control.

5.3.2 Detect dclk delay

Setting DDQS.START=1 and DDQS.AUTO=0, hardware do the detect processing one time.
Setting DDQS.START=1 and DDQS.AUTO=1, hardware do the detect-and-set processing one time.
Setting DDQS.START=1 and DDQS.AUTO=2, hardware periodically do the detect-and-set processing after each auto-refresh command.

The detection result stores in DDQS.CLKD. DDQS.CLKD indicates the delay value of half dclk clock cycle (frequency Fdclk=F(DDR CK)). Thus, the delay value:

DQS_Delay = ¼ tCK = ½ DDQS.CLKD <1>

Delay DQS by ¼ tCK means to the ideal case. Actually, there’re always a gap between the ideal value and the real value. The gap comes from many factors such as IC manufacture processing, PCB layout, noise, etc. So, a revised parameter is introduced to equation <1>:

For READ, DQS_Delay = ½ DDQS.CLKD + DDQSADJ.RDQS <2>
For WRITE, DQS_Delay = ½ DDQS.CLKD + DDQSADJ.WDQS <3>
DDQSADJ can be either positive or negative number to add or sub value from DQS_Delay.
After reset, DDQSADJ.RDQS/WDQS = 0.

5.3.3 Set DDQS.RDQS and DDQS.WDQS

When hardware complete a detect processing:

If DDQS.AUTO=0, hardware do NOT set DDQS.RDQS and DDQS.WDQS;
If DDQS.AUTO=1 or 2, hardware set DDQS.RDQS and DDQS.WDQS according to equation <2> and <3>.

5.3.4 Manual Detect-and-Set Processing

DQS delay value can be set manually.

Step 1: Software set DDQS.RDQS and DDQS.WDQS.
Step 2: Software do write- and-read test on DDR Memory, then compare the read data with the write data.
Step 3: Repeat step 1 and 2.

When the tests complete, software can choose the fittest value for RDQS and WDQS.

5.3.5 Handling the DQS delay detection “ERROR”

The number of delay elements for detection dclk: 256;

The number of delay elements for RDQS: 128;
The number of delay elements for WDQS: 128;

DDRC can't do the detect processing successfully when:

Tmlck > 25.6 ns (256 x 0.1 ns = 25.6 ns)
Or: Fdclk < 40 MHz

According to JEDEC DDR Specification:

For normal DDR, CK > 83 MHz;
For mobile DDR, CK > 0 MHz; there have no requirement for the lower range of CK;
In case detection failed, hardware set RDQS and WDQS with max number if DDQS.AUTO ≠ 0.

5.3.6 DDRC and DDR2 Memory Initialization Sequence

5.3.6.1 Example 1

One 512Mb x16 DDR2 device connected on CS0;
No memory device connected on CS1;
DCK = 133 MHz, CL = 3.

1 After system reset, wait system clock stable before initialize ddrc.
2 Configure the Clock-Control module for ddrc clocks.
3 DDR Memory device need at least 200us initialization time after power-on before it can accept any command.

//----------
// INIT DDRC
//----------

4 Configure DCFG = 0x.
5 Configure DTIMING1 = 0x.
6 Configure DTIMING1 = 0x.
7 Configure DMMAP0 = 0x.
8 Configure DMMAP1 = 0x0000FF00.

//----------
// INIT DDR memory device
//----------

9 Set CKE Pin HIGH : Configure DCTRL = 0x00000002.
10 PRECHAREG-ALL : Configure DCTRL = 0x.
11 Load-Mode-Register EMR2 : Configure DCTRL = 0x.
12 Load-Mode-Register EMR3 : Configure DCTRL = 0x.
13 Load-Mode-Register EMR1 : Configure DCTRL = 0x.
14 Load-Mode-Register MR with DLL reset : Configure DCTRL = 0x.
15 PRECHAREG-ALL : Configure DCTRL = 0x.
16 AUTO-REFRESH : Configure DCTRL = 0x.
17 AUTO-REFRESH : Configure DCTRL = 0x.
18 Load-Mode-Register MR with DLL reset end : Configure DCTRL = 0x.
19 Load-Mode-Register EMR1 with OCD default : Configure DCTRL = 0x.
20 Load-Mode-Register EMR1 with OCD exit : Configure DCTRL = 0x.
21 Wait at least 200 tCK before next step.

//----------
// Enable Refresh Counter
//----------

22 Enable Refresh Counter : Configure DREFCNT = 0x.
23 AUTO-REFRESH : Configure DCTRL = 0x.

//----------
// DQS Delay Detect
//----------

24 Configure DDQSADJ = 0x.
25 Configure DDQS = 0x.
26 Read register DDQS.
27 configure TSEL form min to max, under each value of TSEL, do 28.
28 Configure {MSEL, HL, QUAR} register, form min delay to max delay (relate to 1.2.7). You need write/read some data by CPU or DMA or other device, to check if the sdram work properly. During this process, record the pass configure, you may found there has several configure pass the test, then chose the one that TSEL min &
{MSEL, HL, QUAR} min passed <= {MSEL,HL,QUAR} <= {MSEL, HL, QUAR}max passed.

```
//-----------------------------------------------------------------------------------------------------------
//   END INITIALIZING SEQUENCE
//-----------------------------------------------------------------------------------------------------------
```

5.4 Change Clock Frequency

To save power consumption, the system clock frequency may be changed frequently according to the application. There're 3 ways to change the clock frequency.

5.4.1 Clock-Stop Mode(only in Mobile-ddr)

CPM will auto drive DDRC to clock-stop mode, when use mobile-ddr.
How to change clock, relate CPM spec.

5.4.2 Manually SELF-REFRESH Mode

DDR can stay in SELF-REFRESH & DEEP-POWER-DOWN mode for a long period of time. System clock frequency can be changed during this time. Even more, the clocks to DDRC module can also be stopped to save power-consumption.
Reference Sequence:

1 Manually issue SELF-REFRESH command to DDR.
2 Change relates register in CPM.
3 Change system clock frequency.
4 Drive DDR exit SELF-REFRESH mode.

5.4.3 CPM driven SELF-REFRESH Mode

CPM will auto drive DDRC to self-refresh mode, when use ddr2, ddr1.
How to change clock, relate CPM spec.

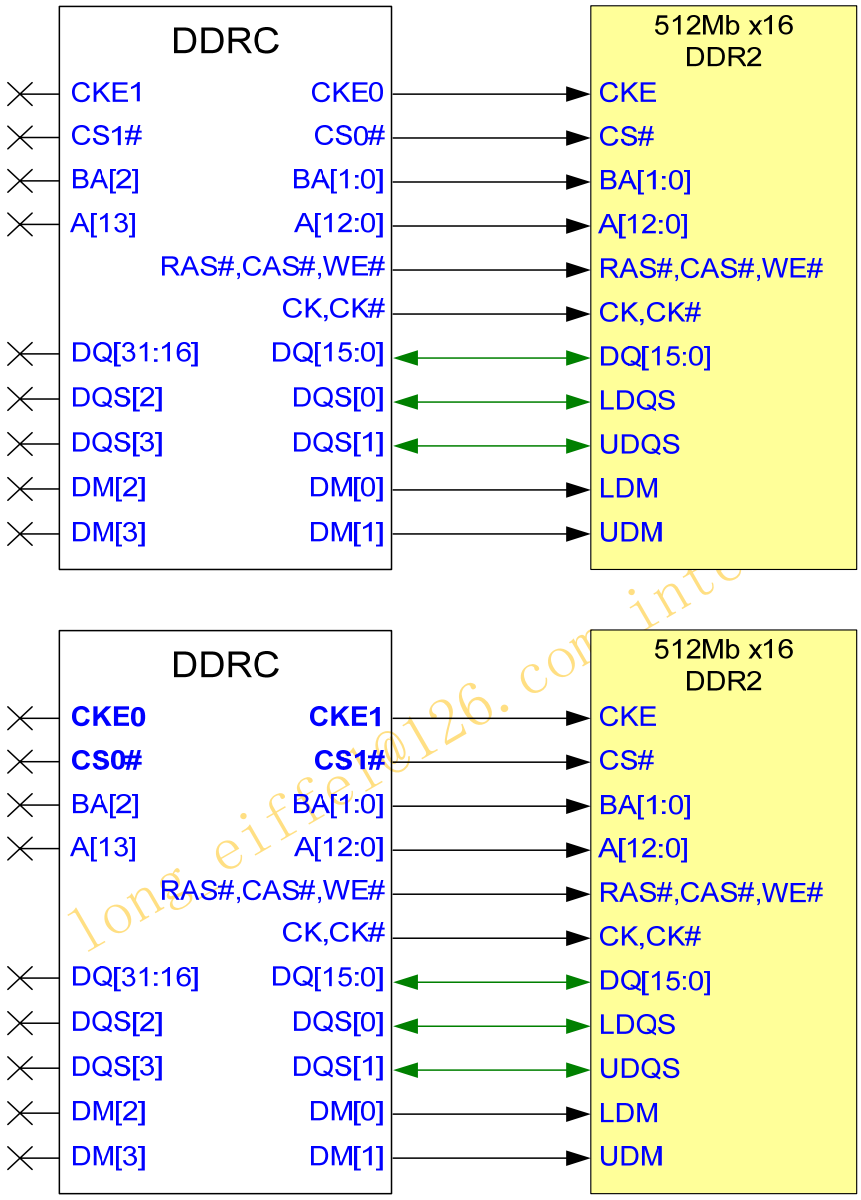
5.5 Data Endian

Fix to little Endian.

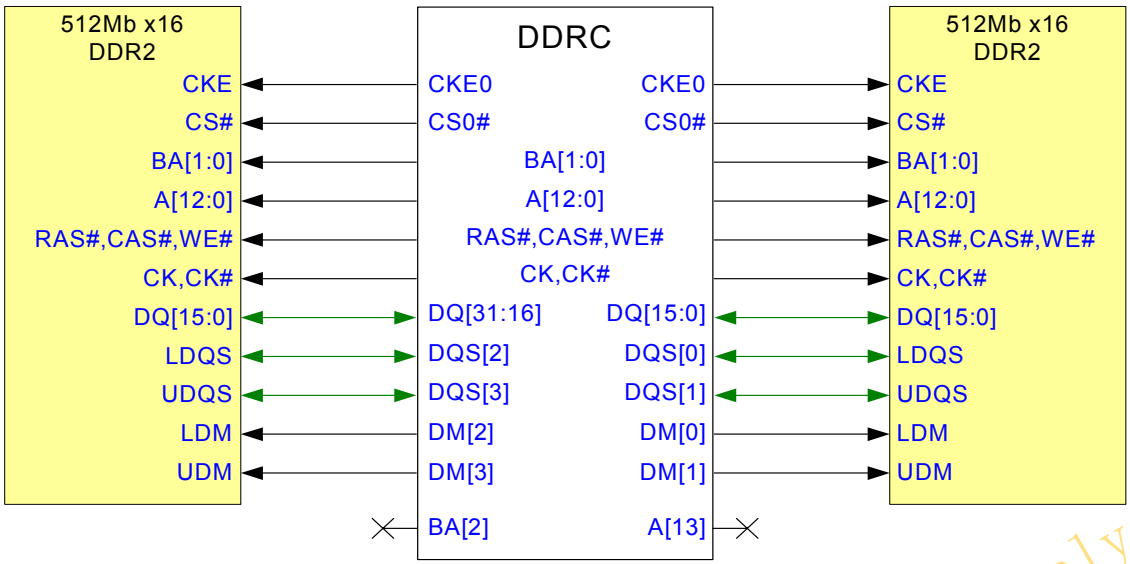
5.6 DDR Connection Diagrams

The following diagrams give examples on the connection to external DDR2 devices.
Note not all the possible connections are listed.

5.6.1 Connection to one 512Mb x16 DDR2 device



5.6.2 Connection to two 512Mb x16 DDR2 devices



6 External NAND Memory Controller

6.1 Overview

The External NAND Memory Controller (NEMC) divides the off-chip memory space and outputs control signals complying with specifications of various types of static memory and bus interfaces. It enables the connection of static memory such as NAND flash memory, etc. to this processor.

- Static memory interface
  - Support 6 external chip selection CS6~1#. Each bank can be configured separately
  - The size and base address of static memory banks are programmable
  - Direct interface to 8-bit or 16-bit (no byte control) bus width external memory interface devices or external static memory to each bank. Read/Write strobe setup time and hold time periods can be programmed and inserted in an access cycle to enable connection to low-speed memory
  - Wait insertion by WAIT pin
  - Automatic wait cycle insertion to prevent data bus collisions in case of consecutive memory accesses to different banks, or a read access followed by a write access to the same bank

- NAND flash interface
  - Support on CS6~CS1, sharing with static memory bank6~bank1
  - Support most types of NAND flashes, including 8-bit and 16-bit bus width, 512B/2K/4K/8KB page size. For 512B page size, 3 and 4 address cycles are supported. For 2K/4K/8KB page size, 4 and 5 address cycles are supported
  - Support read/erase/program NAND flash memory
  - Support boot from NAND flash

6.2 Pin Description

Following table list the NEMC pins.

Table 6-1 NEMC Pin Description

Pin Name	I/O	Signal	Description
Data Bus	I/O	SD15 – SD0	Data I/O.
Address bus	O	SA5–SA0	Address output.
Static chip select 6 ~ 1	O	CS6~1#	Chip select signal that indicates the static bank being accessed.
Read enable	O	RD# /	For Static memory read enable signal.
Write enable	O	WE# /	Static memory write enable signal.
Wait	I	Wait# /	External wait state request signal for memory-like devices.
NAND flash read enable	O	FRE#	NAND flash read enable signal.
NAND flash write enable	O	FWE#	NAND flash write enable signal.
NAND flash ready/busy	I	FRB#	Indicates NAND flash is ready or busy. (When Nand flash boot, GPC30 is used as FRB# of CS1#)

6.3 Physical Address Space Map

Both virtual spaces and physical spaces are 32-bit wide in this architecture. Virtual addresses are translated by MMU into physical address which is further divided into several partitions for static memory, SDRAM, and internal I/O.

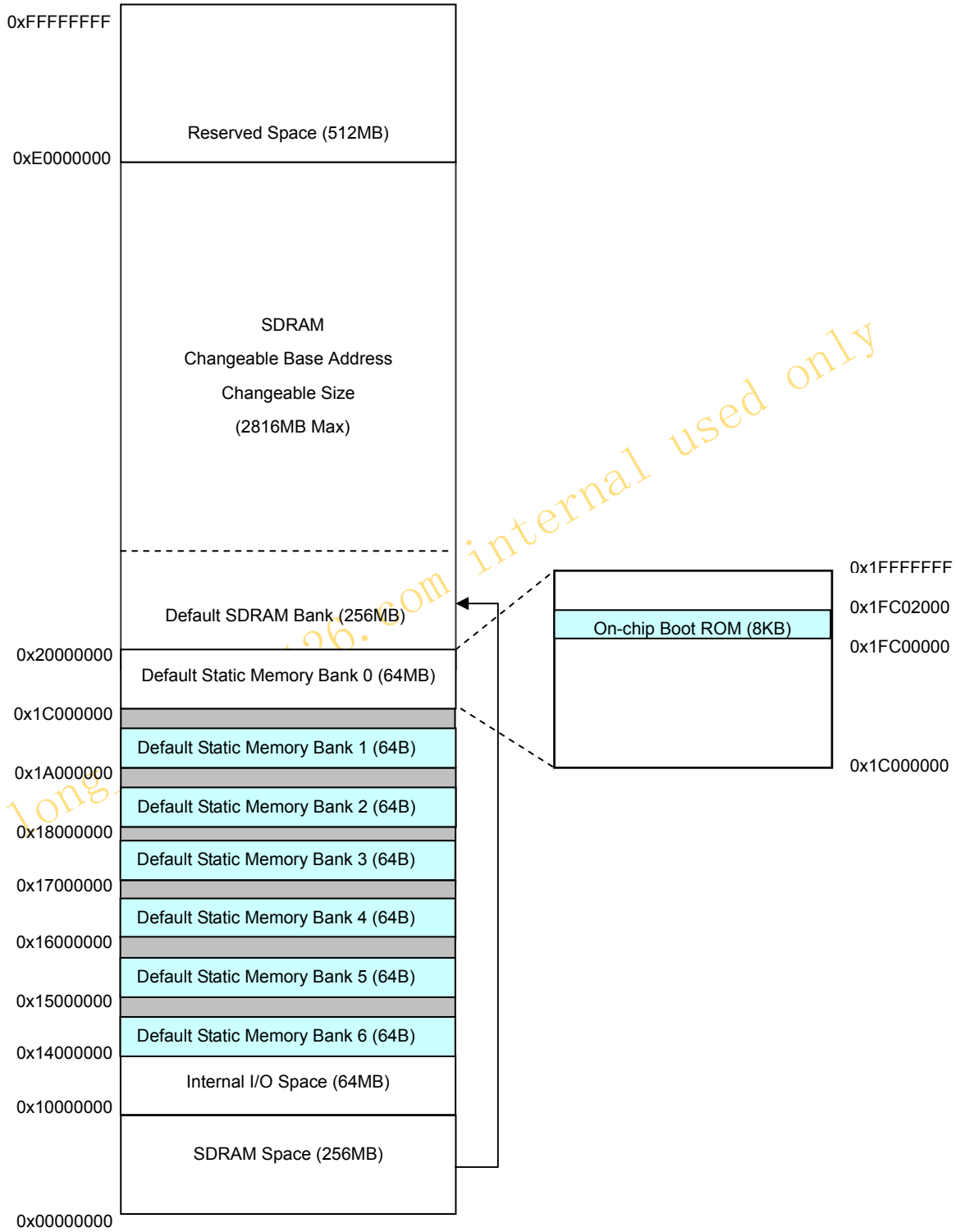


Figure 6-1 Physical Address Space Map

Table 6-2 Physical Address Space Map

Start Address	End Address	Connectable Memory	Capacity
0x00000000	0x0FFFFFFF	SDRAM Memory	256
0x10000000	0x10FFFFFF	I/O Devices on APB Bus	16
0x11000000	0x12FFFFFF	Reserved	32
0x13000000	0x13FFFFFF	I/O Devices on AHB Bus	16
0x14000000	0x1400003F	Static Memory, CS6#	64B
0x14000040	0x14FFFFFF	Reserved	
0x15000000	0x1500003F	Static Memory, CS5#	64B
0x15000040	0x15FFFFFF	Reserved	
0x16000000	0x1600003F	Static Memory, CS4#	64B
0x16000040	0x16FFFFFF	Reserved	
0x17000000	0x1700003F	Static Memory, CS3#	64B
0x17000040	0x17FFFFFF	Reserved	
0x18000000	0x1800003F	Static Memory, CS2#	64B
0x18000040	0x19FFFFFF	Reserved	
0x1A000000	0x1A00003F	Static Memory, CS1#	64B
0x1A000040	0x1BFFFFFF	Reserved	
0x1C000000	0x1FBFFFFF	Reserved	60
0x1FC00000	0x1FC01FFF	On-chip Boot ROM (8kB)	0.008
0x1FC02000	0x1FFFFFFF	Reserved	3.992
0x20000000	0xDFFFFFFF	SDRAM Memory	3072
0xE0000000	0xFFFFFFFF	Reserved	512

The base address and size of each memory banks are configurable. Software can re-configure these memory banks according to the actual connected memories. Following table lists the default configuration after reset.

Table 6-3 Default Configuration of NEMC Chip Select Signals

Chip-Select Signal	Connected Memory	Capacity	Memory Width *1	Start Address	End Address
CS1#	Static memory bank 1	64 B	8, 16, 32	0x1A000000	0x1A00003F
CS2#	Static memory bank 2	64 B	8, 16, 32	0x18000000	0x1800003F
CS3#	Static memory bank 3	64 B	8, 16, 32	0x17000000	0x1700003F
CS4#	Static memory bank 4	64 B	8, 16, 32	0x16000000	0x1600003F
CS5#	Static memory bank 5	64 B	8, 16, 32	0x15000000	0x1500003F
CS6#	Static memory bank 6	64 B	8, 16, 32	0x14000000	0x1400003F

NOTES:

1 Data width of static memory banks can be configured to 8, 16 bits by software.

2 The 8KB address space from H'1FC00000 to H'1FC01FFF in bank 0 is mapped to on-chip boot ROM. The other memory spaces in bank 0 are not used.

6.4 Static Memory Interface

NEMC provides a glueless interface to normal static memory which don't need byte control like SRAM, memory interface IO devices, etc.. It can directly control up to 6 devices using six chip select lines. Additional devices may be supported through external decoding of the address bus.

Each chip select can directly access memory or IO devices that are 8-bits or 16-bits wide. Each device connected to a chip select line has 2 associated registers that control its operation and the access timing to the external device. The Static Memory Control Register SMCRn specifies various configurations for the device. The Static Memory Address Configuration Register SACRn specifies the base address and size for each device, enabling any device to be located anywhere in the physical address range.

The static memory interface includes the following signals:

- Six chip selects, CS6~1#
- Six address signals, SA5-SA0
- One read enable, RD#
- One write enable, WE#
- One wait pin, WAIT#

The SMT field in SMCRn registers specifies the type of memory and BW field specifies the bus width. BOOT_SEL[1:0] pin defines whether system boot from Nor or Nand flash and the page size when boot from Nand flash.

6.4.1 Register Description

Table 6-4 Static Memory Interface Registers

Name	Description	RW	Reset Value	Address	Access Width
SMCR1	Static memory control register 1	RW	0x0FFF7700	0x13410014	32
SMCR2	Static memory control register 2	RW	0x0FFF7700	0x13410018	32
SMCR3	Static memory control register 3	RW	0x0FFF7700	0x1341001C	32
SMCR4	Static memory control register 4	RW	0x0FFF7700	0x13410020	32
SMCR5	Static memory control register 5	RW	0x0FFF7700	0x13410024	32
SMCR6	Static memory control register 6	RW	0x0FFF7700	0x13410028	32
SACR1	Static memory bank 1 address configuration register	RW	0x00001AFE	0x13410034	32
SACR2	Static memory bank 2 address configuration register	RW	0x000018FE	0x13410038	32
SACR3	Static memory bank 3 address configuration register	RW	0x000017FF	0x1341003C	32
SACR4	Static memory bank 4 address configuration register	RW	0x000016FF	0x13410040	32
SACR5	Static memory bank 5 address configuration register	RW	0x000015FF	0x13410044	32
SACR6	Static memory bank 6 address	RW	0x000014FF	0x13410048	32

	configuration register				

6.4.1.1 Static Memory Control Register (SMCR1~6)

SMCR1~6 are 32-bit read/write registers that contain control bits for static memory. On reset, SMCR1~6 are initialized to 0x0FFF7700.

SMCR1 **0x13410014**
SMCR2 **0x13410018**
SMCR3 **0x1341001C**
SMCR4 **0x13410020**
SMCR5 **0x13410024**
SMCR6 **0x13410028**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved			STRV					TAW				TBP				TAH				TAS				BW		Reserved				BL		SMT
RST	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	0	1	1	1	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
31:30	Reserved	Writing has no effect, read as zero.	R
29:24	STRV	**Static Memory Recovery Time.** Its value is the number of idle cycles (0~63 cycles) inserted between bus cycles when switching from one bank to another bank or between a read access to a write access in the same bank. Its initial value is 0xF (15 cycles).	RW
23:20	TAW	**Access Wait Time.** For normal memory, these bits specify the number of wait cycles to be inserted in read strobe time. For burst ROM, these bits specify the number of wait cycles to be inserted in first data read strobe time. **TAW3~0 Wait cycle Wait# Pin** 0000 0 cycle Ignored 0001 1 cycle Enabled 0010 2 cycles Enabled 0011 3 cycles Enabled 0100 4 cycles Enabled 0101 5 cycles Enabled 0110 6 cycles Enabled 0111 7 cycles Enabled 1000 8 cycles Enabled 1001 9 cycles Enabled 1010 10 cycles Enabled	RW

		1011 12 cycles Enabled 1100 15 cycles Enabled 1101 20 cycles Enabled 1110 25 cycles Enabled 1111 31 cycles Enabled (Initial Value)	
19:16	TBP	**Burst Pitch Time.** For burst ROM, these bits specify the number of wait cycles to be inserted in subsequent access. For normal memory, these bits specify the number of wait cycles to be inserted in write strobe time. **TBP3~0 Wait cycle Wait# Pin** 0000 0 cycle Ignored 0001 1 cycle Enabled 0010 2 cycles Enabled 0011 3 cycles Enabled 0100 4 cycles Enabled 0101 5 cycles Enabled 0110 6 cycles Enabled 0111 7 cycles Enabled 1000 8 cycles Enabled 1001 9 cycles Enabled 1010 10 cycles Enabled 1011 12 cycles Enabled 1100 15 cycles Enabled 1101 20 cycles Enabled 1110 25 cycles Enabled 1111 31 cycles Enabled (Initial Value)	RW
15	Reserved	Writing has no effect, read as zero.	R
15:12	TAH	**Address Hold Time.** These bits specify the number of wait cycles to be inserted from negation of read/write strobe to address. **TAH2~0 Wait cycle** 0000 0 cycle 0001 1 cycle 0010 2 cycles 0011 3 cycles 0100 4 cycles 0101 5 cycles 0110 6 cycles 0111 7 cycles (Initial Value) 1000 8 cycles 1001 9 cycles 1010 10 cycles 1011 11 cycles	RW

		1100 12 cycles 1101 13 cycles 1110 14 cycles 1111 15 cycles	
11	Reserved	Writing has no effect, read as zero.	R
11:8	TAS	**Address Setup Time.** These bits specify the number of wait cycles (0~15 cycles) to be inserted from address to assertion of read/write strobe. **TAS2~0 Wait cycle** 000 0 cycle 001 1 cycle 010 2 cycles 011 3 cycles 100 4 cycles 101 5 cycles 110 6 cycles 111 7 cycles (Initial Value) 1000 8 cycles 1001 9 cycles 1010 10 cycles 1011 11 cycles 1100 12 cycles 1101 13 cycles 1110 14 cycles 1111 15 cycles	RW
7:6	BW	**Bus Width.** These bits specify the bus width. this filed is writeable and are initialized to 0 by a reset. **BW1~0 Bus Width** 00 8 bits (Initial Value) 01 16 bits 10 Reserved 11 Reserved	RW
5:3	Reserved	Writing has no effect, read as zero. **NOTE:** Don't write Bit3 to 1.	R
2:1	BL	**Burst Length (BL1, BL0).** When Burst ROM is connected; these bits specify the number of burst in an access. These bits are only valid when SMT is set to 1. **BL1~0 Burst Length** 00 4 consecutive accesses. Can be used with 8- or 16-bit bus width (Initial Value) 01 8 consecutive accesses. Can be used with 8- or 16-bit bus width	

		10 16 consecutive accesses. Can only be used with 8- or 16-bit bus width 11 32 consecutive accesses. Can only be used with 8-bit bus width	
0	SMT	**Static Memory Type (SMT).** This bit specifies the type of static memory. **SMT** **Description** 0 Normal Memory (Initial Value) 1 Burst ROM	RW

6.4.1.2 Static Bank Address Configuration Register (SACR1~6)

SACR1~6 defines the physical address for static memory bank 1 to 6, respectively. Each register contains a base address and a mask. When the following equation is met:

(*physical_address* [31:24] & **MASK$_n$**) == **BASE$_n$**

The bank n is active. The *physical_address* is address output on internal system bus. Static bank regions must be programmed so that each bank occupies a unique area of the physical address space. Programming overlapping bank regions will result in unpredictable error. These registers are initialized by a reset.

SACR1 **0x13410034**
SACR2 **0x13410038**
SACR3 **0x1341003C**
SACR4 **0x13410040**
SACR5 **0x13410044**
SACR6 **0x13410048**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																BASE								MASK							
RST1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	0	1	1	1	1	1	1	1	0
RST2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	1	1	1	1	1	1	1	0
RST3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	1	1	1	1	1	1	1	1	1
RST4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0	1	1	1	1	1	1	1	1
RST5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	1	1	1	1	1	1	1	1
RST6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	1	1	1	1	1	1	1	1

Bits	Name	Description	RW
31:16	Reserved	Writing has no effect, read as zero.	R
15:8	BASE	**Address Base:** Defines the base address of Static Bank n (n = 1 to 6).	RW

		The initial values are: SACR1.BASE 0x1A SACR2.BASE 0x18 SACR3.BASE 0x17 SACR4.BASE 0x16 SACR5.BASE 0x15 SACR6.BASE 0x14	
23:20	MASK	**Address Mask**: Defines the mask of Static Bank n (n = 1 to 6). The initial values are: SACR1.MASK 0xFE SACR2.MASK 0xFE SACR3.MASK 0xFF SACR4.MASK 0xFF SACR5.MASK 0xFF SACR6.MASK 0xFF	RW

6.4.2 Example of Connection

Following figures shows examples of connection to 16- and 8-bit data width normal memory.

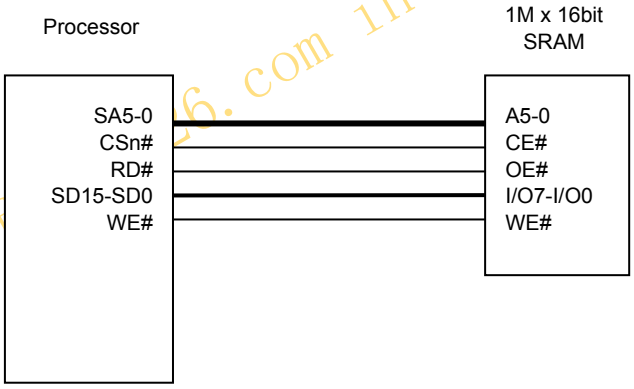


Figure 6-2 Example of 16-Bit Data Width SRAM Connection

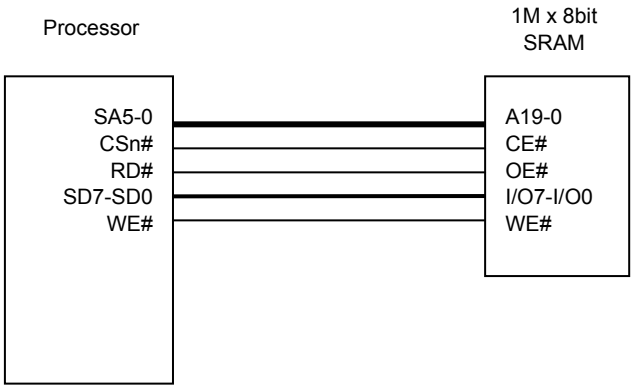


Figure 6-3 Example of 8-Bit Data Width SRAM Connection

6.4.3 Basic Interface

When SMT field in SMCRn (n = 1 to 6) is 0, normal memory (non-burst ROM, Flash, normal SRAM or memory-like device) is connected to bank n. When bank n (n = 1 to 6) is accessed, CSn# is asserted as soon as address is output. In addition, the RD# signal, which can be used as OE#, and write control signals WE# is asserted.

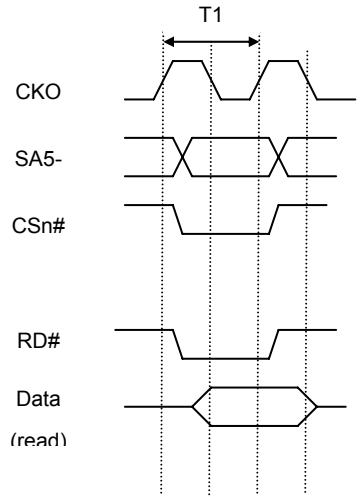
The TAS field in SMCRn is the latency from CSn# to read/write strobe. The TAW3 field is the delay time of RD# in read access. TBP3~0 field is the delay time of WE# and WEn# in write access. In addition, any number of waits can be inserted by means of the external pin (WAIT#). The TAH field is the latency from RD# and WEn# negation to CSn# negation, also the hold time to address and write data.

All kinds of normal memories (non-burst ROM, normal SRAM and Flash) have the same read and write timing. There are some requirements for writes to flash memory. Flash memory space must be un-cacheable and un-buffered. Writes must be exactly the width of the populated Flash devices on the data bus (no byte writes or word writes to a 16-bit bus, and so on). Software is responsible for partitioning commands and data, and writing them out to Flash in the appropriate sequence.

Glossary

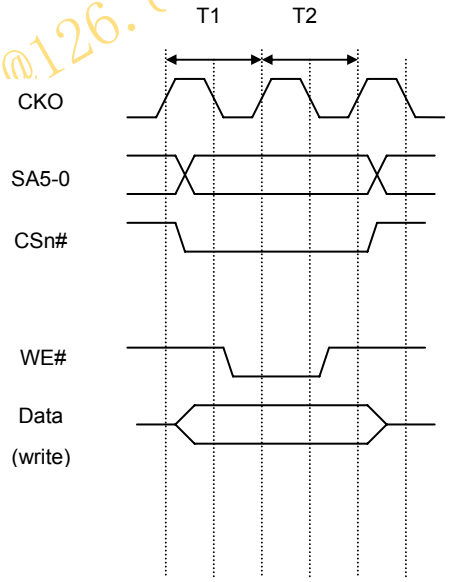
Th – hold cycle
Tw – wait cycle
Ts – setup cycle
T1 – read inherent cycle or first write inherent cycle
T2 – last write inherent cycle
Tb – burst read inherent cycle

Following figures show the timing of normal memory. A no-wait read access is completed in one cycle and a no-wait write access is completed in two cycles. Therefore, there is no negation period in case of access at minimum pitch.



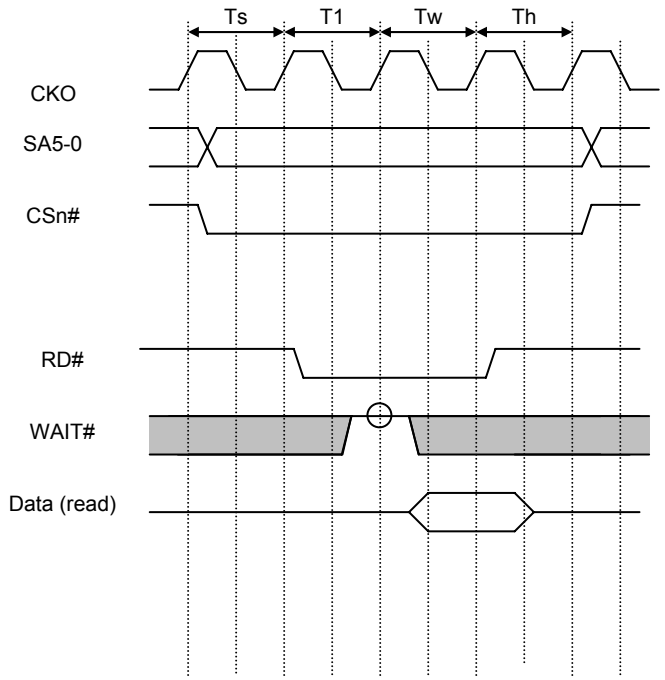
*In this example, SMCRn:MT = 0, TAS = 0, TAW = 0, TAH = 0

Figure 6-4 Basic Timing of Normal Memory Read



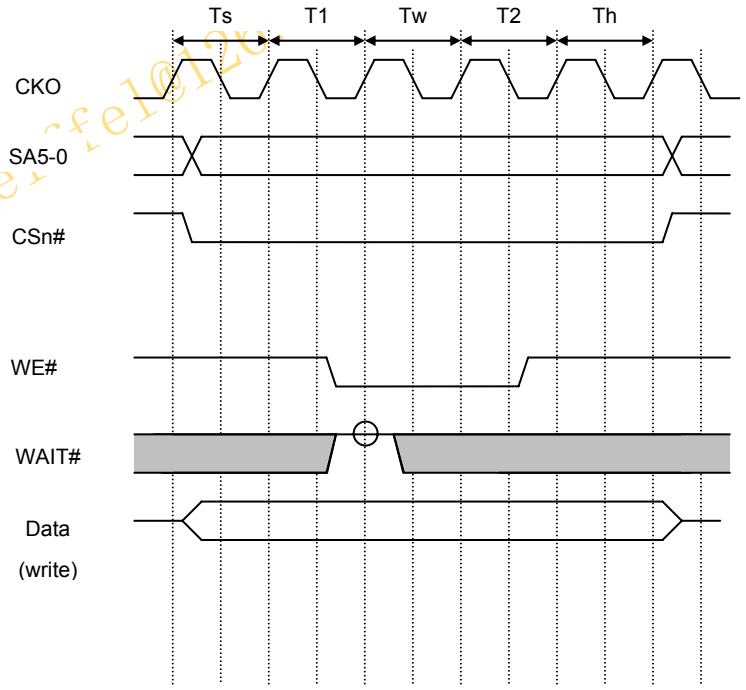
*In this example, SMCRn: SMT = 0, TAS = 0, TBP = 0, TAH = 0

Figure 6-5 Basic Timing of Normal Memory Write



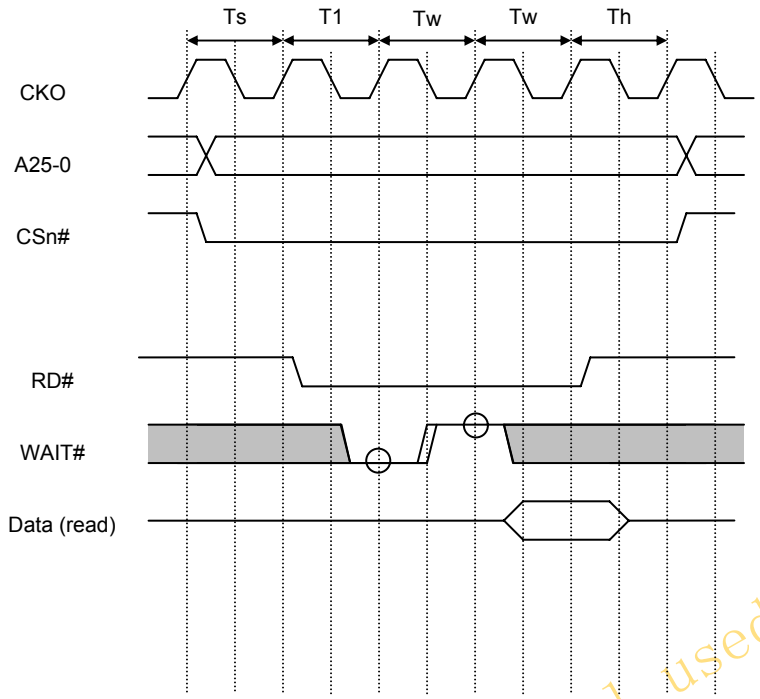
*In this example, SMCRn: SMT = 0, TAS = 1, TAW = 1, TAH = 1

Figure 6-6 Normal Memory Read Timing With Wait (Software Wait Only)



*In this example, SMCRn: SMT = 0, TAS = 1, TBP = 1, TAH = 1

Figure 6-7 Normal Memory Write Timing With Wait (Software Wait Only)



*In this example, SMCRn: SMT = 0, TAS = 1, TAW = 1, TAH=1

Figure 6-8 Normal Memory Read Timing With Wait (Wait Cycle Insertion by WAIT# pin)

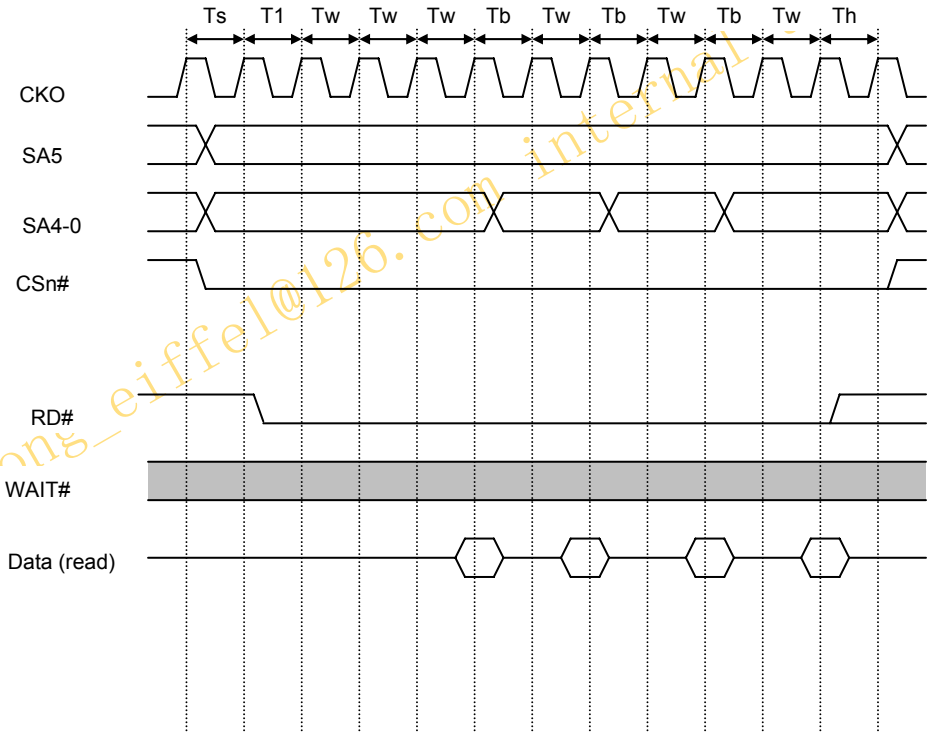
6.4.4 Burst ROM Interface

Setting SMT to 1 in SMCRn allows burst ROM to be connected to bank n (n = 1 to 6). The burst ROM interface provides high-speed access to ROM that has a nibble access function. Basically, access is performed in the same way as for normal memory, but when the first cycle ends, only the address is changed before the next access is executed. When 8-bit burst ROM is connected, the number of consecutive accesses can be set as 4, 8, 16, or 32 with bits BL1~0. When 16-bit ROM is connected, 4, 8, or 16 can be set in the same way.

For burst ROM read, TAW sets the delay time from read strobe to the first data, TBP sets the delay time from consecutive address to data. Burst ROM writes have the same timing as normal memory except TAW instead of TBP is used to set the delay time of write strobe.

WAIT# pin sampling is always performed when one or more wait states are set.

Following figures show the timing of burst ROM.



*In this example, SMT = 1, BL = 0, TAS = 1, TAW = 3,TBP = 1,TAH = 1

Figure 6-9 Burst ROM Read Timing (Software Wait Only)

6.5 NAND Flash Interface

NAND flash can be connected to static memory bank 6~ band 1. Both 8-bit and 16-bit NAND flashes are supported. A mechanism for booting from NAND flash is also supported.

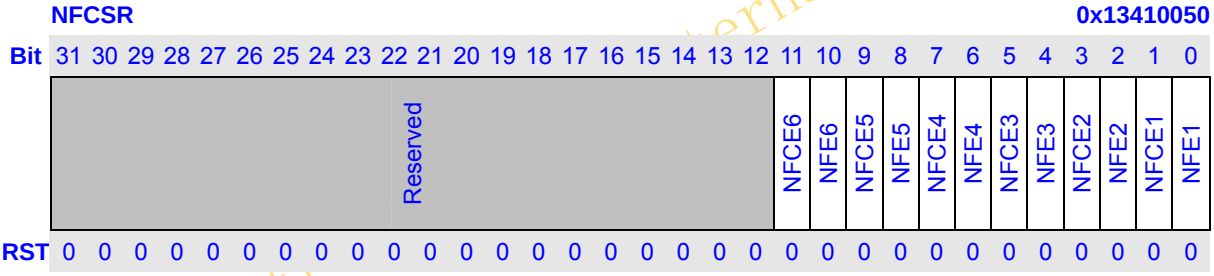
6.5.1 Register Description

Table 6-5 NAND Flash Interface Registers

Name	Description	RW	Reset Value	Address	Access Width
NFCSR	NAND flash control/status register	RW	0x00000000	0x13410050	32
PNCR	NAND PN control register	RW	0x00000000	0x13410100	32
PNDR	NAND PN data register	RW	0x00005AA5	0x13410104	32
BITCNT	NAND bit counter	R	0x00000000	0x13410108	32

6.5.1.1 NAND Flash Control/Status Register (NFCSR)

NFCSR is a 32-bit read/write register that is used to configure NAND flash. It is initialized by any reset.



Bits	Name	Description	RW
31:16	Reserved	Writing has no effect, read as zero.	R
1/3/5/7/9/11	FCEn (n=1,2,3,4,5,6)	**NAND Flash FCE# Assertion Control :** Controls the assertion of NAND Flash FCEn#. When set, FCEn# is always asserted until this bit is cleared. When the NAND flash require FCEn# to be asserted during read busy time, this bit should be set. **FCE** **Description** 0 FCEn# is asserted as normal static chip enable(Initial value) 1 FCEn# is always asserted	RW
0/2/4/6/8/10	NFEn (n=1,2,3,4,5,6)	**NAND Flash Enable:** Specifies if NAND flash is connected to static bank n. When system is configured to boot from NAND flash, this bit is initialized to 1. **NFE** **Description** 0 Static bank n is not used as NAND flash 1 Static bank n is used as NAND flash	RW

6.5.1.2 NAND PN Control Register (PNCR)

PNCR is a 32-bit read/write register that is used to control NAND flash data randomization. It is initialized by any reset.

PNCR **0x13410100**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																										BIT_RST	BIT_SEL	BIT_EN	Reserved	PNRST	PNEN
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
31:6	Reserved	Writing has no effect, read as zero.	R
5	BIT_RST	**NAND BIT Counter Reset:** Reset Bit counter. When this bit is written to 1, the bit counter is reset to 0. This bit is write-only.	W
4	BIT_SEL	**NAND BIT Counter Select:** Bit counter's counting select. **BIT_SEL Description** 0 Calculate number of "0" in NAND read data 1 Calculate number of "1" in NAND read Data	RW
3	BIT_EN	**NAND BIT Counter Enable:** Enable/disable bit counter counting. **BIT_EN Description** 0 Bit counting is disabled 1 Bit counting is enabled	RW
2	Reserved	Writing has no effect, read as zero.	R
1	PNRST	**NAND Flash PN Reset:** Reset seed of randomizer. When this bit is written to 1, the seed of randomizer is reset. This bit is write-only.	W
0	PNEN	**NAND Flash PN Enable:** Specifies if NAND flash read/write data randomization is enabled. This bit is initialized to 0. **PNEN Description** 0 Data randomization is disabled 1 Data randomization is enabled	RW

6.5.1.3 NAND PN Data Register (PNDR)

PNDR is a 23-bit read/write register that is used for seed of randomizer during NAND read/write.

PNDR **0x13410104**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved									PNDR																						
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0	1	0	1	0	1	0	0	1	0	1

6.5.1.4 NAND Bit Counter (BITCNT)

BITCNT is a 32-bit read/write register that is used to counting the number of “1” or “0” (based on BIT_SEL) in Nand read data and keep counting during Nand read till BIT Counter Reset. It is initialized by any reset.

BITCNT **0x13410108**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BITCNT																															
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

6.5.2 NAND Flash Boot Loader

To support boot from NAND flash, 8KB on-chip Boot ROM is implemented. Following figure illustrates the structure of NAND Flash Boot Loader.

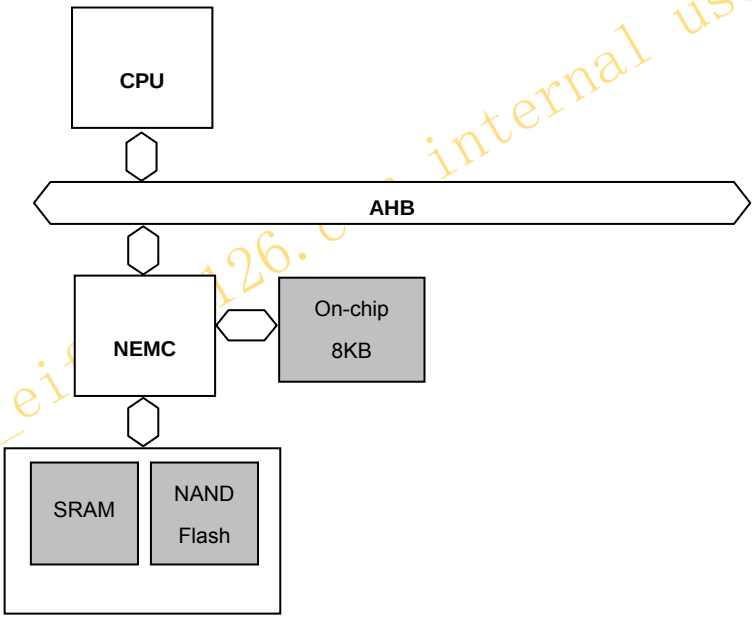


Figure 6-10 Structure of NAND Flash Boot Loader

When system is configured to boot from NAND flash, after reset, the program in Boot ROM is executed and the program will copy the first 8K bytes of NAND flash to internal memory for further initialization.

Generally, the boot code will copy more NAND flash content to DRAM. Then the main program will be executed on DRAM.

When system is configured to boot from NAND flash, software may know the nand flash page size through BOOT_SEL[2:0] pin.

6.5.3 NAND Flash Operation

Set NFEn bit of NAND Flash Control/Status Register (NFCSR) will enable access to NAND flash. The partition of static bank n (n=1~6) is changed as following figure. Writes to any of address space will be translated to NAND flash address cycle. Writes to any of command space will be translated to NAND flash command cycle. **CAUTION:** don't read to address and command space, and these two partitions should be uncacheable. Reads and writes to any of data space will be translated to NAND flash data read/write cycle. DMA access to data space is supported to increase the speed of data read/write. The DMA access cannot exceed the page boundary (512 bytes or 2K bytes) of NAND.

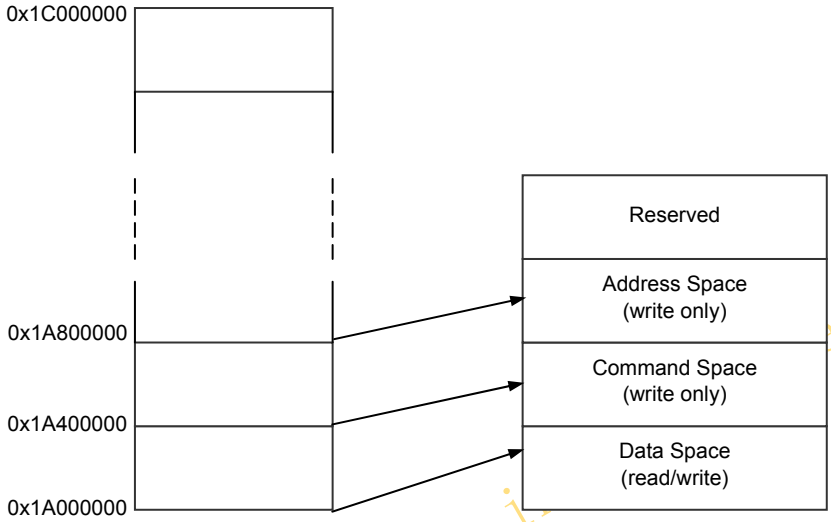


Figure 6-11 Static Bank 1 Partition When NAND Flash is Used (an example)

The timing of NAND flash access is configured by SMCRn and is same as normal static memory timing, except that CSn# is controlled by NFCE bit NFCSR. CSn# is always asserted when NFCE is 1. When NFCE is 0, CSn# is asserted as normal static memory access.

The control signals for direction connection of NAND flash are CSn#, FRE#, FWE#, FRB#(GPIO), A1 and A0. Following figure shows the connection between processor and NAND Flash.

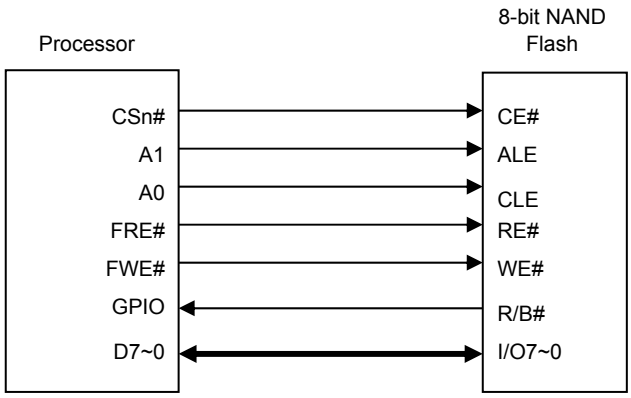


Figure 6-12 Example of 8-bit NAND Flash Connection

7 BCH Controller

7.1 Overview

The BCH Controller implements data ECC encoding and decoding.

7.2 Register Description

Table 7-1 BCH Registers

Name	Description	RW	Reset Value	Address	Access Width
BHCR	BCH Control register	R	0x00000000	0x134D0000	32
BHCSR	BCH Control Set register	W	Undefined	0x134D0004	32
BHCCR	BCH Control Clear register	W	Undefined	0x134D0008	32
BHCNT	BCH ENC/DEC Count register	RW	0x00000000	0x134D000C	32/16
BHDR	BCH data register	W	Undefined	0x134D0010	8
BHPAR0	BCH Parity 0 register	RW	0x00000000	0x134D0014	32/16/8
BHPAR1	BCH Parity 1 register	RW	0x00000000	0x134D0018	32/16/8
BHPAR2	BCH Parity 2 register	RW	0x00000000	0x134D001C	32/16/8
BHPAR3	BCH Parity 3 register	RW	0x00000000	0x134D0020	32/16/8
BHPAR4	BCH Parity 4 register	RW	0x00000000	0x134D0024	32/16/8
BHPAR5	BCH Parity 5 register	RW	0x00000000	0x134D0028	32/16/8
BHPAR6	BCH Parity 6 register	RW	0x00000000	0x134D002C	32/16/8
BHPAR7	BCH Parity 7 register	RW	0x00000000	0x134D0030	32/16/8
BHPAR8	BCH Parity 8 register	RW	0x00000000	0x134D0034	32/16/8
BHPAR9	BCH Parity 9 register	RW	0x00000000	0x134D0038	32/16/8
BHERR0	BCH Error Report 0 register	R	0x00000000	0x134D003C	32/16
BHERR1	BCH Error Report 1 register	R	0x00000000	0x134D0040	32/16
BHERR2	BCH Error Report 2 register	R	0x00000000	0x134D0044	32/16
BHERR3	BCH Error Report 3 register	R	0x00000000	0x134D0048	32/16
BHERR4	BCH Error Report 4 register	R	0x00000000	0x134D004C	32/16
BHERR5	BCH Error Report 5 register	R	0x00000000	0x134D0050	32/16
BHERR6	BCH Error Report 6 register	R	0x00000000	0x134D0054	32/16
BHERR7	BCH Error Report 7 register	R	0x00000000	0x134D0058	32/16
BHERR8	BCH Error Report 8 register	R	0x00000000	0x134D005C	32/16
BHERR9	BCH Error Report 9 register	R	0x00000000	0x134D0060	32/16
BHERR10	BCH Error Report 10 register	R	0x00000000	0x134D0064	32/16
BHERR11	BCH Error Report 11 register	R	0x00000000	0x134D0068	32/16
BHINT	BCH Interrupt Status register	R	0x00000000	0x134D006C	32
BHINTE	BCH Interrupt Enable register	RW	0x00000000	0x134D0070	32
BHINTES	BCH Interrupt Set register	W	Undefined	0x134D0074	32
BHINTEC	BCH Interrupt Clear register	W	Undefined	0x134D0078	32

7.2.1 BCH Control Register (BHCR)

BHCR is a 32-bit read/write register that is used to configure BCH controller. It is initialized by any reset.

BHCR 0x134D0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																								BDMA	Reserved	BSEL			ENCE	BRST	BCHE
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
31:8	Reserved	Writing has no effect, read as zero.	R
7	BDMA	**BCH DMA Enable**: It is used to enable or disable dma transfer during correction. **BDMA** **Description** 0 DMA transfer is disabled (Initial value) 1 DMA transfer is enabled	RW
6	Reserved	Writing has no effect, read as zero.	R
5:3	BSEL	**BCH Encoding/Decoding Bit Select**: It is used to select the correction algorithm among 4-bit, 8-bit, 12-bit, 16-bit, 20-bit and 24-bit BCH. **BSEL** **Description** 000 4-bit correction (initial value) 001 8-bit correction 010 12-bit correction 011 16-bit correction 100 20-bit correction 101 24-bit correction	
2	ENCE	**BCH Encoding/Decoding Select**: It is used to define whether in encoding or in decoding phase when BCH is used. **ENCE** **Description** 0 Decoding (Initial value) 1 Encoding	RW
1	BRST	**BCH Reset:** It is used to reset BCH controller. This bit is cleared automatically by hardware and always read as 0. **BRST** **Description** 0 BCH controller is not reset (Initial value) 1 BCH controller is reset	W
0	BCHE	**BCH Enable:** BCH correction is enable/disable. **BCHE** **Description** 0 BCH is disabled (initial value) 1 BCH is enabled	RW

7.2.2 BCH Control Set Register (BHCSR)

BHCSR is a 32-bit write-only register that is used to set BCH controller to 1.

When write 1 to BHCSR, the corresponding bit in BHCR register is set to 1. Write 0 to BHCSR is ignored.

BHCSR **0x134D0004**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																								BDMAS	Reserved	BSELS			ENCES	BRSTS	BCHES
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
31:8	Reserved	Writing has no effect, read as zero.	R
7	BDMAS	**BCH DMA Enable Set**: It is used to set BHCR.BDMA to 1.	W
6	Reserved	Writing has no effect, read as zero.	R
5:3	BSELS	**BCH Encoding/Decoding Bit Select Set**: It is used to set BHCR.BSEL to 1.	W
2	ENCES	**BCH Encoding/Decoding Select Set**: It is used to set BHCR.ENCE to 1.	W
1	BRSTS	**BCH Reset Set:** It is used to set BHCR.BRST to 1.	W
0	BCHES	**BCH Enable Set:** It is used to set BHCR.BCHE to 1.	W

7.2.3 BCH Control Clear Register (BHCCR)

BHCCR is a 32-bit write-only register that is used to clear BCH controller to 0.

When write 1 to BHCCR, the corresponding bit in BHCR register is cleared to 0. Write 0 to BHCCR is ignored.

BHCCR **0x134D0008**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																								BDMAC	Reserved	BSELC			ENCEC	BRSTC	BCHEC
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
31:8	Reserved	Writing has no effect, read as zero.	R
7	BDMAC	**BCH DMA Enable Clear**: It is used to clear BHCR.BDMA to 0.	W
6	Reserved	Writing has no effect, read as zero.	R

5:3	BSELC	**BCH Encoding/Decoding Bit Select Clear**: It is used to clear BHCR.BSEL to 0.	W
2	ENCEC	**BCH Encoding/Decoding Select Clear**: It is used to clear BHCR.ENCE to 0.	W
1	Reserved	Writing has no effect, read as zero.	R
0	BCHEC	**BCH Enable Clear:** It is used to clear BHCR.BCHE to 0.	W

7.2.4 BCH ENC/DEC Count Register (BHCNT)

BCHCNT is a 32-bit read/write register that is used to indicate the total number of 4-bit data during encoding or decoding. It is initialized by any reset.

BHCNT **0x134D000C**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved					DEC Count											Reserved					ENC Count											
RST	0	0	0	0	0	1	0	0	0	0	1	1	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
31:27	Reserved	Writing has no effect, read as zero.	R
26:16	DEC Count	**DEC Count**: It is used to indicate total 4-bit data count in BCH decoding which includes data + parity. For example, total data + parity is 538 bytes, the field of DEC Count should be set to 'h434 which is the initial value.	RW
15:11	Reserved	Writing has no effect, read as zero.	R
10:0	ENC Count	**ENC Count**: It is used to indicate total byte count in BCH encoding which just includes 4-bit data and should be less and equal to 1996 4-bit (which is equal to 998 Bytes) when 16-bit BCH is selected.	RW

7.2.5 BCH Data Register (BHDR)

BHDR is an 8-bit write-only register that is used to transfer ecc data to BCH.

BHDR **0x134D0010**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																								BCHDR							
RST	0	0	0	0	0	0	0	0	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	X	x	x	X

7.2.6 BH Parity Register (BHPARn, n=0,1,2,3,4,5,6,7,8,9)

BHPARn (n=0,1,2,3,4,5,6,7,8,9) are all 32-bit read/write register that contains the encoding parity data

during BCH correction. It is initialized by any reset and BRST of BHCR.
When 24-bit BCH is selected, BHPAR0~BHPAR9 consist of the 312 bits of parity data and bit 0 of BHPAR0 is the 312th bit of parity data and bit 23 of BHPAR9 is the 1st bit of parity data.

When 20-bit BCH is selected, BHPAR0~BHPAR8 consist of the 260 bits of parity data, and bit 0 of BHPAR0 is the 260th bit of parity data and bit 3 of BHPAR8 is the 1st bit of parity data.

When 16-bit BCH is selected, BHPAR0~BHPAR6 consist of the 208 bits of parity data, and bit 0 of BHPAR0 is the 208th bit of parity data and bit 15 of BHPAR6 is the 1st bit of parity data.

When 12-bit BCH is selected, BHPAR0~BHPAR4 consist of the 156 bits of parity data, and bit 0 of BHPAR0 is the 156th bit of parity data and bit 27 of BHPAR4 is the 1st bit of parity data.

When 8-bit BCH is selected, BHPAR0~BHPAR3 consist of the 104 bits of parity data and bit 0 of BHPAR0 is the 104th bit of parity data and bit 7 of BHPAR3 is the 1st bit of parity data.

Similarly, when 4-bit BCH is selected, the two parity register, BHPAR0 and BHPAR1 together consist of the 52 bits of parity data and bit 0 of BHPAR0 is the 52th bit of parity data and bit 19 of BHPAR1 is the 1st bit of parity data.

BHPAR0																															**0x134D0014**
BHPAR1																															**0x134D0018**
BHPAR2																															**0x134D001C**
BHPAR3																															**0x134D0020**
BHPAR4																															**0x134D0024**
BHPAR5																															**0x134D0028**
BHPAR6																															**0x134D002C**
BHPAR7																															**0x134D0030**
BHPAR8																															**0x134D0034**
BHPAR9																															**0x134D0038**
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BHPAR0~BHPAR9																															
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

7.2.7 BCH Error Report Register (BCHERRn, n=0,1,2,3,4,5,6,7,8,9,10,11)

BCHERRn is 32-bit read/write register that contains the index for each error after BCH decoding. It is initialized by any reset and BRST of BCHCR.

BCHERR0 contains INDEX1 and INDEX0.
BCHERR1 contains INDEX3 and INDEX2.
BCHERR2 contains INDEX5 and INDEX4.
BCHERR3 contains INDEX7 and INDEX6.
BCHERR4 contains INDEX9 and INDEX8.
BCHERR5 contains INDEX11 and INDEX10.

BCHERR6 contains INDEX13 and INDEX12.
BCHERR7 contains INDEX15 and INDEX14.
BCHERR8 contains INDEX17 and INDEX16.
BCHERR9 contains INDEX19 and INDEX18.
BCHERR10 contains INDEX21 and INDEX20.
BCHERR11 contains INDEX23 and INDEX22.

BCHERR0	**0x134D003C**
BCHERR1	**0x134D0040**
BCHERR2	**0x134D0044**
BCHERR3	**0x134D0048**
BCHERR4	**0x134D004C**
BCHERR5	**0x134D0050**
BCHERR6	**0x134D0054**
BCHERR7	**0x134D0058**
BCHERR8	**0x134D005C**
BCHERR9	**0x134D0060**
BCHERR10	**0x134D0064**
BCHERR11	**0x134D0068**

Bit	31:29	28:16	15:13	12:0
	Reserved	INDEXn(n=1,3,5,7,9,11,13,15,17,19,21,23)	Reserved	INDEXn(n=0,2,4,6,8,10,12,14,16,18,20,22)
RST	0	0	0	0

Bits	Name	Description	RW
31:29	Reserved	Writing has no effect, read as zero.	R
28:16	INDEXn	**Error Bit Index:** It is used to indicate the location of the error bit. For example, INDEX=2, it means the second bit is an error bit.	R
15:13	Reserved	Writing has no effect, read as zero.	R
12:0	INDEXn	**Error Bit Index:** It is used to indicate the location of the error bit. For example, INDEX=2, it means the second bit is an error bit.	R

7.2.8 BCH Interrupt Status Register (BHINT)

BHINT is a 32-bit read-only register that contains the interrupt flag and error count information during BCH correction. It is initialized by any reset. Software write 1 to clear the corresponding bit except ERRC.

BHINT **0x134D006C**

Bit	31:27	26:5	4	3	2	1	0
	ERRC	Reserved	ALL_f	DECF	ENCF	UNCOR	ERR
RST	0	0	0	0	0	0	0

Bits	Name	Description	RW
31:27	ERRC	**Error Count:** It indicates the number of errors in the data block and these bits are also reset by BHCR.BRST bit. **ERRC Description** 0 No errors or uncorrection error occurs (Initial value) 1 One error in the data block 2 Two errors in the data block 3 Three errors 4 Four errors 5 Five errors 6 Six errors 7 Seven errors 8 Eight errors … 24 Twenty-four errors	R
27:5	Reserved	Writing has no effect, read as zero.	R
4	ALL_f	**ALL_f:** It indicates that all data received during decoding are 0xf. When receiving all 0xf data, BCH doesn't correct the data and no error occurs. **ALL_f Description** 0 Not all data (data + parity bytes) are 0xf (Initial value) 1 All data (data + parity bytes) are 0xf	R
3	DECF	**Decoding Finish:** It indicates that hardware finish BCH decoding. **DECF Description** 0 Decoding not Finish (Initial value) 1 Decoding Finish	R
2	ENCF	**Encoding Finish:** It indicates that hardware finish BCH encoding. **ENCF Description** 0 Encoding not Finish (Initial value) 1 Encoding Finish	R
1	UNCOR	**Uncorrection Error:** It indicates that hardware finish BCH encoding. **UNCOR Description** 0 No uncorrectable error (Initial value) 1 Uncorrectable error occur	R
0	ERR	**Error:** It indicates that hardware detects error bits in data in the data block during BCH decoding. **ERR Description** 0 No error (Initial value) 1 Error occur	R

7.2.9 BCH Interrupt Enable Set Register (BHINTES)

BHINTES is a 32-bit write-only register that is used to set BHINTE register. Writing 1 to BHINTES will

set the corresponding bit in BHINTE to 1. Writing 0 to BHINTES is ignored.

BHINTES **0x134D0074**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																											ALL_FES	DECFES	ENCFES	UNCORES	ERRES
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
31:5	Reserved	Writing has no effect, read as zero.	R
4	ALL_FES	**ALL_F Interrupt Enable Set:** It is used to set BHINTE.ALL_FE to 1.	W
3	DECFES	**Decoding Finish Interrupt Enable Set:** It is used to set BHINTE.DECFE to 1.	W
2	ENCFES	**Encoding Finish Interrupt Enable Set:** It is used to set BHINTE.ENCFE to 1.	W
1	UNCORES	**Uncorrection Error Interrupt Enable Set:** It is used to set BHINTE.ENCFE to 1.	W
0	ERRES	**Error Interrupt Enable Set:** It is used to set BHINTE.ERRE to 1.	W

7.2.10 BCH Interrupt Enable Clear Register (BHINTEC)

BHINTEC is a 32-bit write-only register that is used to clear BHINTE register. Writing 1 to BHINTEC will clear the corresponding bit in BHINTE to 0. Writing 0 to BHINTEC is ignored.

BHINTEC **0x134D0078**

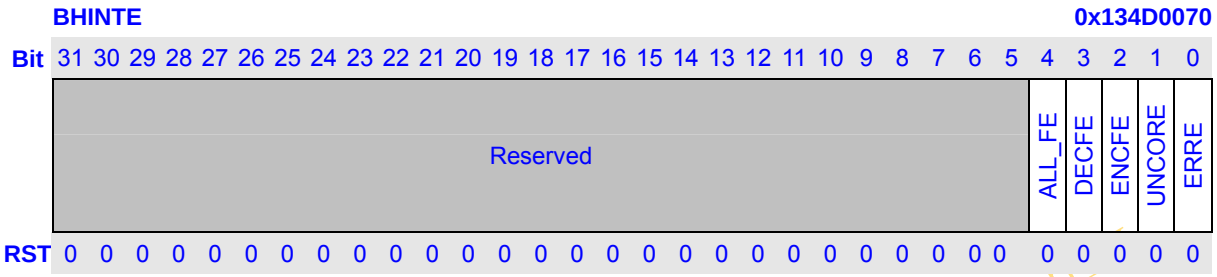
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																											ALL_FEC	DECFEC	ENCFEC	UNCORC	ERREC
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
31:5	Reserved	Writing has no effect, read as zero.	R
4	ALL_FEC	**ALL_F Interrupt Enable Clear:** It is used to clear BHINTE.ALL_FE to 0.	W
3	DECFEC	**Decoding Finish Interrupt Enable Clear:** It is used to clear BHINTE.DECFE to 0.	W
2	ENCFEC	**Encoding Finish Interrupt Enable Clear:** It is used to clear BHINTE.ENCFE to 0.	W
1	UNCOREC	**Uncorrection Error Interrupt Enable Clear:** It is used to clear	W

		BHINTE.ENCFE to 0.	
0	ERREC	**Error Interrupt Enable Clear:** It is used to set BHINTE.ERRE to 0.	W

7.2.11 BCH Interrupt Enable Register (BHINTE)

BHINTE is a 32-bit read/write register that is used to enable/disable interrupts during BCH correction. It is initialized by any reset.



Bits	Name	Description	RW
31:5	Reserved	Writing has no effect, read as zero.	R
4	ALL_FE	**ALL_F Interrupt Enable:** It is used enable or disable all_f data interrupt. **ALL_FE Description** 0 Disable ALL_F data interrupt (Initial value) 1 Enable ALL_F data interrupt	RW
3	DECFE	**Decoding Finish Interrupt Enable:** It is used to enable or disable decoding finish interrupt. **DECFE Description** 0 Disable Decoding Finish Interrupt (Initial value) 1 Enable Decoding Finish Interrupt	RW
2	ENCFE	**Encoding Finish Interrupt Enable:** It is used to enable or disable encoding finish interrupt. **ENCFE Description** 0 Disable Encoding Finish Interrupt (Initial value) 1 Enable Encoding Finish Interrupt	RW
1	UNCORE	**Uncorrection Error Interrupt Enable:** It is used to enable or disable uncorrection error interrupt. **UNCORE Description** 0 Disable Uncorrectable Error interrupt (Initial value) 1 Enable Uncorrectable Error Interrupt	RW
0	ERRE	**Error Interrupt Enable:** It is used to enable or disable error interrupt. **ERRE Description** 0 Disable Error interrupt (Initial value) 1 Enable Error interrupt	RW

7.3 BCH Operation

BCH controller uses BCH(n, k) codes. Here n is less and equal to 8191-bit and k is less and equal to 7879-bit in 24-bit correction, 7931-bit in 20-bit correction, 7983-bit in 16-bit correction, 8035-bit in 12-bit correction, 8087-bit in 8-bit correction and 8139-bit in 4-bit correction. During encoding, hardware will generate 312-bit parity data in 24-bit correction, 260-bit parity data in 20-bit correction, 208-bit parity data in 16-bit correction, 156-bit parity data in 12-bit correction, 104-bit parity data in 8-bit correction or 52-bit parity data in 4-bit correction. Parity data can be read out by cpu or dma. During decoding, if there are error bits in data block, after decoding BCHERRn registers will hold the error bit location that can be read by cpu or dma.

7.3.1 Encoding Sequence

BCH encoding can be operated by cpu or dma.

7.3.1.1 CPU

1 Set BCHCR.BCHE to 1 to enable BCH controller.
2 Select 24-bit, 20-bit, 16-bit, 12-bit, 8-bit or 4-bit correction by setting BCHCR.BSEL.
3 Set BCHCR.ENCE to 1 to enable encoding.
4 Set BCHCR.BRST to 1 to reset BCH controller.
5 Set BCHCNT.ENC_COUNT to data block size in bytes.
6 Byte-write all data block to BCHDR.
7 Check BCHINTS.ENCF bit or by enabling encoding finish interrupt.
8 When encoding finishes, read out the parity data in BCHPARn.

7.3.1.2 DMA

1 Set BCHCR.BCHE to 1 to enable BCH controller.
2 Select 24-bit, 20-bit, 16-bit, 12-bit, 8-bit or 4-bit correction by setting BCHCR.BSEL.
3 Set BCHCR.ENCE to 1 to enable encoding.
4 Set BCHCR.BRST to 1 to reset BCH controller.
5 Set BCHCNT.ENC_COUNT to data block size in bytes.
6 Set BCHCR.BDMA to 1 to select DMA transfer.
7 Start DMA transfer after configuring DMA channel.
8 DMA read data block from system memory and write to BCH controller automatically.
9 DMA will wait BCH encoding request when finishes writing data block.
10 BCH controller will issue encoding request to DMA when encoding ends.
11 DMA start to read out parity data.
12 After parity data is read out, BCH automatically reset itself and clear BCHINT.ENCF.

NOTE: When DMA is enabled, software should guarantee not to enable encoding finish interrupt.

7.3.2 Decoding Sequence

BCH decoding can be operated by cpu or dma.

7.3.2.1 CPU

1 Set BCHCR.BCHE to 1 to enable BCH controller.
2 Select 24-bit, 20-bit, 16-bit, 12-bit, 8-bit or 4-bit correction by setting BCHCR.BSEL.
3 Clear BCHCR.ENCE to 0 to enable decoding.
4 Set BCHCR.BRST to 1 to reset BCH controller.
5 Set BCHCNT.DEC_COUNT to data block size in bytes.
6 Byte-write all data block to BCHDR.
7 Check BCHINTS.DECF bit or by enabling decoding finish interrupt.
8 When decoding finishes, read out the status in BCHINT and error report in BCHERRn.

7.3.2.2 Decoding Sequence

1 Set BCHCR.BCHE to 1 to enable BCH controller.
2 Select 24-bit, 20-bit, 16-bit, 12-bit, 8-bit or 4-bit correction by setting BCHCR.BSEL.
3 Clear BCHCR.ENCE to 0 to enable decoding.
4 Set BCHCR.BRST to 1 to reset BCH controller.
5 Set BCHCNT.DEC_COUNT to data block size in bytes.
6 Set BCHCR.BDMA to 1 to select DMA transfer.
7 Start DMA transfer after configuring DMA channel.
8 DMA read data block from system memory and write to BCH controller automatically.
9 DMA will wait BCH decoding request when finishes writing data block.
10 BCH controller will issue decoding request to DMA when decoding ends.
11 DMA start to read out bch int status and error report data and write to memory.
12 If using descriptor DMA, if the data block needs error correction, the current data block syndrome generation and last data block error correction can be executed in pipeline automatically by DMA.
13 After status and error report data is read out, BCH automatically reset itself and clear BCHINT.DECF and Error status in BCHINT.

8 BDMA Controller

BDMA controller (BDMAC) is dedicated to transfer data between BCH, external memories and memory-mapped external devices.

8.1 Features

- Support up to 3 independent DMA channels
- Descriptor or No-Descriptor Transfer
- Transfer data units: byte, 2-byte (half word), 4-byte (word), 16-byte, 32-byte or 64-byte
- Transfer number of data unit: 1 ~ 224
- Independent source and target port width: 8-bit, 16-bit, 32-bit

8.2 Register Descriptions

Table 8-1 BDMAC Registers

Name	Description	RW	Reset Value	Address	Access Size (bit)
DSA0	DMA Source Address 0	RW	0x0	0x13450000	32
DTA0	DMA Target Address 0	RW	0x0	0x13450004	32
DTC0	DMA Transfer Count 0	RW	0x0	0x13450008	32
DRT0	DMA Request Source 0	RW	0x0	0x1345000C	32
DCS0	DMA Channel Control/Status 0	RW	0x0	0x13450010	32
DCM0	DMA Command 0	RW	0x0	0x13450014	32
DDA0	DMA Descriptor Address 0	RW	0x0	0x13450018	32
DSD0	DMA Stride Address 0	RW	0x0	0x1345001C	32
DSA1	DMA Source Address 1	RW	0x0	0x13450020	32
DTA1	DMA Target Address 1	RW	0x0	0x13450024	32
DTC1	DMA Transfer Count 1	RW	0x0	0x13450028	32
DRT1	DMA Request Source 1	RW	0x0	0x1345002C	32
DCS1	DMA Channel Control/Status 1	RW	0x0	0x13450030	32
DCM1	DMA Command 1	RW	0x0	0x13450034	32
DDA1	DMA Descriptor Address 1	RW	0x0	0x13450038	32
DSD1	DMA Stride Address 1	RW	0x0	0x1345003C	32
DSA2	DMA Source Address 2	RW	0x0	0x13450040	32
DTA2	DMA Target Address 2	RW	0x0	0x13450044	32
DTC2	DMA Transfer Count 2	RW	0x0	0x13450048	32
DRT2	DMA Request Source 2	RW	0x0	0x1345004C	32
DCS2	DMA Channel Control/Status 2	RW	0x0	0x13450050	32
DCM2	DMA Command 2	RW	0x0	0x13450054	32
DDA2	DMA Descriptor Address 2	RW	0x0	0x13450058	32
DSD2	DMA Stride Address 2	RW	0x0	0x1345005C	32
DNT0	DMA Nand Timer 0	RW	0xC	0x134500C0	32
DNT1	DMA Nand Timer 1	RW	0xC	0x134500C4	32
DNT2	DMA Nand Timer 2	RW	0xC	0x134500C8	32

DMAC1	DMA Control 1 Register	R/ W	0x0	0x13450300	32
DIRQP1	DMA Interrupt Pending 1	R	0x0	0x13450304	32
DDR1	DMA Doorbell 1 Register	RW	0x0	0x13450308	32
DDRS1	DMA Doorbell Set 1 Register	W	0x0	0x1345030C	32
DCKE1	DMA Clock Enable 1 Register	RW	0x0	0x13450310	32
DCKES1	DMA Clock Enable Set Register	W	0x0	0x13450314	32
DCKEC1	DMA Clock Enable Clear Register	W	0x0	0x13450318	32

8.2.1 DMA Source Address (DSAn, n = 0 ~ 2)

DSA0, DSA1, DSA2 **0x13450000, 0x13450020, 0x13450040**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SA																															
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
31:0	SA	Source physical address.	RW

8.2.2 DMA Target Address (DTAn, n = 0 ~ 2)

DTA0, DTA1, DTA2 **0x13450004, 0x13450024, 0x13450044**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TA																															
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
31:0	TA	Target physical address.	RW

8.2.3 DMA Transfer Count (DTCn, n = 0 ~ 2)

DTC0, DTC1, DTC2 **0x13450008, 0x13450028, 0x13450048**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved								TC																							
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
31:24	Reserved	Write has no effect, read as zero.	R
23:0	TC	When Stride address transfer is disabled: TC holds the number of data unit to transfer and it counts down to 0 at the end. When Stride address transfer is enabled: TC composes of two parts: The lower 16 bits: the number of data unit for sub-block transfer The higher 8 bits: the number of sub-block And both the two parts count down to 0 at the end.	RW

8.2.4 DMA Request Types (DRTn, n = 0 ~ 2)

DRT0, DRT1, DRT2 **0x1345000c, 0x1345002c, 0x1345004c**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																										RT					
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
31:6	Reserved	Write has no effect, read as zero.	R
5:0	RT	Transfer request type.	RW

Table 8-2 Transfer Request Types

RT5-0	Description
000000	Reserved.
000001	Reserved.
000010	BCH Encoding DMA request.
000011	BCH Decoding DMA request.
000100	Reserved.
000101	Reserved.
000110	NAND0 DMA request.
000111	NAND1 DMA request.
001000	Auto-request.
001001	Reserved.
001010	Reserved.
001011	Reserved.
001100	External request with DREQn. (external address ←→ external device with DACKn)
Other	Reserved.

NOTES:

1 Only auto request can be concurrently selected in all channels with different source and target address.

2 Only channel 1 and channel 2 can handle external request. Channel 2 handles external request 0, and channel 1 handles external request 1.

3 NAND0 and NAND1 are corresponded to two NAND chip selects, for example chip select 0 and chip select 1.

8.2.5 DMA Channel Control/Status (DCSn, n = 0 ~ 2)

DCS0, DCS1, DCS2 **0x13450010, 0x13450030, 0x13450050**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	NDES	DES8	LASTMD		Reserved	FRBS			CDOA								Reserved				BERR					Reserved	BUERR	AR	TT	HLT	BAC	CTE
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
31	NDES	Descriptor or No-Descriptor Transfer Select. 0: Descriptor Transfer; 1: No-descriptor Transfer.	RW
30	DES8	Descriptor 8 Word. 0: 4-word descriptor; 1: 8-word descriptor.	RW
29:28	LASTMD	BCH Decoding Last Mode.	RW

		00: Last Mode 0 which means there is one descriptor for a BCH decoding block 01: Last Mode 1 which means there is two descriptor for a BCH decoding block, the last descriptor points to parity or part of data and parity 10: Last Mode 2 which means there is three descriptor for a BCH decoding block, the last descriptor points to parity data 11: reserved **NOTES:** 1 Here a BCH decoding block includes decoding data and parity. 2 LASTMD is only for channel 0.	
27	Reserved	Write has no effect, read as zero.	R
26:24	FRBS	FRB Pin Select. 000: GPIO Group A 27 bit as frb1 001: GPIO Group A 28 bit as frb1 010: GPIO Group A 29 bit as frb1 011: GPIO Group B 4 bit as frb1 100: GPIO Group B 5 bit as frb1 101~111: reserved **NOTE:** FRBS is only for channel 1.	RW
23:16	CDOA	Copy of offset address of last completed descriptor from that in DMA command register. Software could know which descriptor is just completed combining with count terminate interrupt resulted by DCSn.CT. (Ignored in No-Descriptor Transfer)	RW
15:12	Reserved	Write has no effect, read as zero.	R
11:7	BERR	BCH error number. It indicates the biggest error number within a BCH decoding block during the whole descriptor chain. If it is 0, it means there is no error during the whole descriptor chain. (Only channel 0 has this field for BCH transfer)	RW
6	Reserved	Write has no effect, read as zero.	R
5	BUERR	BCH Uncorrectable Error. 0: No uncorrectable error occurs during the whole descriptor chain 1: Uncorrectable error occurs during the whole descriptor chain (Only channel 0 has this field for BCH transfer)	RW
4	AR	Address Error. 0: no address error; 1: address error.	RW
3	TT	Transfer Terminate. 0: No-Link Descriptor or No-Descriptor DMA transfer does not end 1: No-Link Descriptor or No-Descriptor DMA transfer end	RW
2	HLT	DMA halt.	RW

		0: DMA transfer is in progress; 1: DMA halt.	
1	BAC	BCH Auto Correction. 0: BCH auto-correction is disabled 1: BCH auto-correction is enabled (Only channel 0 has this bit for BCH auto correction)	RW
0	CTE	Channel transfer enable. 0: disable; 1: enable.	RW

8.2.6 DMA Channel Command (DCMn, n = 0 ~ 2)

DCM0, DCM1, DCM2 **0x13450014, 0x13450034, 0x13450054**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	EACKS	EACKM	ERDM		Reserved		BLAST	Reserved	SAI	DAI	Reserved						SP		DP		Reserved	TSZ			NRD	NWR	NAC	Reserved		STDE	TIE	LINK
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
31	EACKS	External DACK Output Level Select. 0: active high; 1: active low.	RW
30	EACKM	External DACK Output Mode Select. 0: output in read cycle; 1: output in write cycle.	RW
29:28	ERDM	External DREQ Detection Mode Select. 00: Low level detection 01: Falling edge detection 10: High level detection 11: Rising edge detection	RW
27:26	Reserved	Write has no effect, read as zero.	R
25	BLAST	BCH/NAND last. 0: non-last data block for BCH/NAND; 1: last data block for BCH/NAND. (Only channel 0 support BCH transfer; all channel support Nand transfer, when it is used for nand, it means the last data block transfer for one nand dma request detection)	RW
24	Reserved	Write has no effect, read as zero.	R
23	SAI	Source Address Increment. 0: no increment; 1: increment.	RW
22	DAI	Target Address Increment. 0: no increment; 1: increment.	RW
19:16	Reserved	Write has no effect, read as zero.	R
15:14	SP	Source port width. 00: 32-bit; 01: 8-bit; 10: 16-bit; 11: reserved.	RW

13:12	DP	Target port width. 00: 32-bit; 01: 8-bit; 10: 16-bit; 11: reserved. (**NOTE:** for bch transfer encoding, DP only can be 32-bit or 8-bit; for bch decoding, DP only can be 32-bit)	RW
11	Reserved	Write has no effect, read as zero.	R
10:8	TSZ	Transfer Data Size of a data unit. 000: 32-bit; 001: 8-bit; 010: 16-bit; 011: 16-byte; 100: 32-byte; 101: 64-byte; others: reserved.	RW
7	NRD	Direct read nand. 0: non-direct read nand; 1: enable direct read nand. (Only channel 1 has this field for Nand transfer)	RW
6	NWR	Direct write nand. 0: non-direct write nand; 1: enable direct write nand. (Only channel 1 has this field for Nand transfer)	RW
5	NAC	Nand AL/CL from Data. 0: AL/CL from data is disabled; 1: enable AL/CL from data. (If AL/CL from data is disabled, AL/CL is from address[23:22] written to nand; When AL/CL from data is enabled, bit 31 of the data indicates AL, bit 30 of the data indicates CL; When AL/CL from data is enabled, be sure to set SP to 32-bit, set DP according to SMCR.BW in memory controller) (Only channel 1 has this field for Nand transfer)	RW
4:3	Reserved	**NOTE:** Don't write 1 to these bits, reserved them to 0.	RW
2	STDE	Stride Disable/Enable. 0: address stride disable; 1: address stride enable.	RW
1	TIE	Transfer Interrupt Enable (TIE). 0: disable interrupt; 1: enable interrupt when TT is set to 1.	RW
0	LINK	Descriptor Link Enable. 0: disable; 1: enable. (Ignored in No-Descriptor Transfer)	RW

8.2.7 DMA Descriptor Address (DDAn, n = 0 ~ 2)

This register is ignored in No-Descriptor Transfer.

DDA0, DDA1, DDA2 **0x13450018, 0x13450038, 0x13450058**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DBA																				DOA								Reserved			
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
31:12	DBA	Descriptor Base Address.	RW
11:4	DOA	Descriptor Offset Address. When 4-descriptor is used, DOA is used for the offset address of DDA for next descriptor fetch.	RW
3:0	Reserved	Write has no effect, read as zero.	R

NOTE: When 8-descriptor is used, next descriptor fetch address is from 8th word of last descriptor, that is the 0th~27th bit of the 8th word of last descriptor is mapped to DDA[31:4] for next descriptor fetch.

8.2.8 DMA Stride Address (DSDn, n = 0 ~ 2)

This register is ignored in No-Descriptor Transfer.

When address stride transfer is enabled in Descriptor mode, after a sub-block defined in DTCRn is finished transferring, the source or target stride address will be added up to the corresponding source or target address and the transfer will keep going until the transfer ends which means TC in DTCRn reach 0.

DSD0, DSD1, DSD2 **0x1345001C, 0x1345003C, 0x1345005C**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TSD																SSD															
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
31:16	TSD	Target Stride Address.	RW
15:0	SSD	Source Stride Address.	RW

8.2.9 DMA Nand Timer (DNTn, n = 0 ~ 2)

This register is used for nand low pulse detect and for AL and CL from data.

DNT0, DNT1, DNT2 **0x134500C0, 0x134500C4, 0x134500C8**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved										DTCT						DNTE	Reserved											DNT			
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0

Bits	Name	Description	RW
31:23	Reserved	Write has no effect, read as zero.	R
22:16	DTCT	Tail Counter. When Nand AL/CL from data is enabled, the counter indicates the actual word written to nand in the last transfer. It is used for transfer count when nand AL/CL from data is enabled. During the last transfer (DTCR == 1), if DCMR.TSZ is set for 16-byte, 32-byte or 64-byte, when DTCT is not equal to 0, the value in DTCT indicates the actual word number written to nand in the last transfer.	RW
15	DNTE	Nand Detect Timer enable. 0: Nand detect timer disable 1: Nand detect timer enable	RW
14:6	Reserved	Write has no effect, read as zero.	R
5:0	DNT	Nand Detect Timer. When Nand detect timer is enabled, the timer starts running down, when the timer is down to zero, it generates a request to DMA for data transfer.	RW

8.2.10 DMA Control

DMAC1 controls channel 0~2.

DMAC1 **0x13450300**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																						PM		Reserved				HLT	AR	Reserved	DMAE
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
31:10	Reserved	Write has no effect, read as zero.	R
9:8	PM	Channel priority mode. 00: CH0, CH1 > CH2 01: CH1, CH2 > CH0 10: CH2 > CH0, CH1 11: CH0, CH1, CH2 For example, when PM == 2'b00, it means set1 includes ch0 and ch1 and set2 includes ch2, set 1 has the higher priority than set 2, within one set, channel priority is round robin, that is: ch0→ch1→ch2.	RW

7:4	Reserve	Write has no effect, read as zero.	R
3	HLT	Global halt status, halt occurs in any channel, the bit should set to 1. 0: no halt 1: halt occurred	RW
2	AR	Global address error status, address error occurs in any channel, the bit should be set to 1. 0: no address error 1: address error occurred	RW
1	Reserved	Write has no effect, read as zero.	R
0	DMAE	Global DMA transfer enable. 0: disable DMA channel transfer 1: enable DMA channel transfer	RW

8.2.11 DMA Interrupt Pending (DIRQP)

DMAC supports total 3 pending interrupt which are in DIRQP.

DIRQP **0x13450304**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																													CIRQ2	CIRQ1	CIRQ0
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
31:3	Reserved	Write has no effect, read as zero.	R
2:0	CIRQn	CIRQn (n=0~2) denotes pending status for corresponding channel. 0: no abnormal situation or normal DMA transfer is in progress 1: abnormal situation occurred or normal DMA transfer done	RW

8.2.12 DMA Doorbell (DDR)

DDR supports channel 0~2.

DDR **0x13450308**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																													DB2	DB1	DB0
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
31:3	Reserved	Write has no effect, read as zero.	R
2:0	DBn	DMA Doorbell for each channel, n=0~2, for example DB0 is for DMA channel 0. Software set it to 1 and hardware clears it to 0. 0: disable DMA controller to fetch the first descriptor or DMA controller clears it to 0 as soon as it starts to fetch the descriptor 1: Write 1 to DDS will set the corresponding DBn bit to 1 and enable DMA controller to fetch the first descriptor For example, write 0x00000001 to DDS, DB0 bit is set to 1 and enable DMA channel 0 to fetch the first descriptor. Write 0 to DDS, no meaning.	R

8.2.13 DMA Doorbell Set (DDRS)

DDRS supports channel 0~2.

DDRS **0x1345030c**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																													DBS2	DBS1	DBS0
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
31:3	Reserved	Write has no effect, read as zero.	R
2:0	DBSn	DMA Doorbell Set for each channel. 0: ignore 1: Set the corresponding DBn bit to 1	W

8.2.14 DMA Clock Enable (DCKE)

DCKE supports channel 0~2.

DCKE **0x13450310**

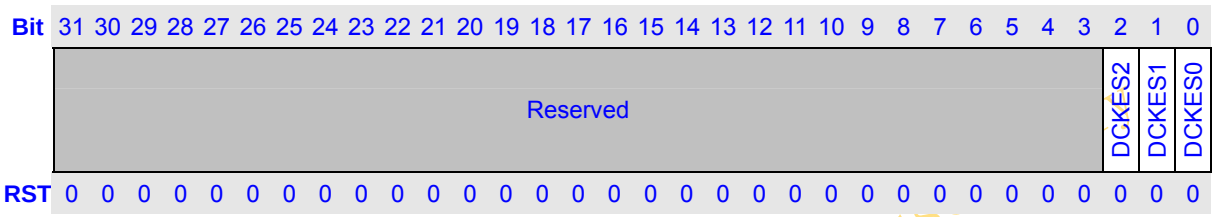
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																													DCKE2	DCKE1	DCKE0
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
31:3	Reserved	Write has no effect, read as zero.	R
2:0	DCKEn	DMA Clock Enable for each channel. 0: ignore 1: Set the corresponding DCKEn bit to 1	RW

8.2.15 DMA Clock Enable Set (DCKES)

DCKES supports channel 0~2.

DCKES **0x13450314**

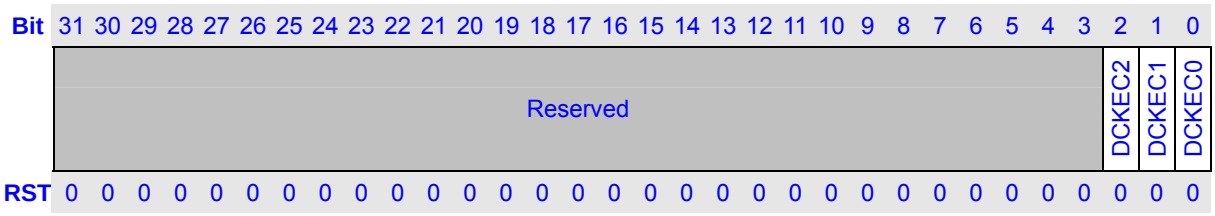


Bits	Name	Description	RW
31:3	Reserved	Write has no effect, read as zero.	R
2:0	DCKESn	DMA Clock Enable Set for each channel. 0: ignore 1: Set the corresponding DCKESn bit to 1 to enable corresponding channel clock	W

8.2.16 DMA Clock Clear Set (DCKEC)

DCKEC supports channel 0~2.

DCKEC **0x13450318**



Bits	Name	Description	RW
31:3	Reserved	Write has no effect, read as zero.	R
2:0	DCKECn	DMA Clock Enable Clear for each channel. 0: ignore 1: Set the corresponding DCKECn bit to 1 to disable corresponding channel clock	W

8.3 DMA manipulation

8.3.1 Descriptor Transfer

8.3.1.1 Normal Transfer

To do proper Descriptor DMA transfer, do as following steps:

1 First of all, open channel clock by setting DCKEn register for corresponding channel.
2 Check whether the status of DMA controller are available, that is, for global control (DMAC), ensure that DMAC.AR=0 and DMAC.HLT=0; while for expected channels, ensure that DCSn.AR=0, DCSn.HLT=0, DCSn.TT=0 and DTCn=0.
3 Select 4 word or 8 word descriptor by DCSn.DES8.
4 For Descriptor transfer, guarantee DCSn.NDES=0.
5 Initiate channel request register DRSRn.
6 Build descriptor in memory. Write the first descriptor address in DDAn and the address must be 16Bytes aligned in 4word descriptor and 32Bytes aligned in 8word descriptor. The descriptor address includes two parts: Base and Offset address. If the descriptor is linked, the 32-bit address of next descriptor is composed of 20-bit Base address in DDAn and 8-bit Offset address in DES3.DOA and the four LSB is 0x0. See Table 8-3 for the detailed 4-word descriptor structure.

   NOTE: if stride address transfer is enabled, the address must be 32Bytes aligned because DES4 needs to read out.

7 Set 1 to the corresponding bit in DDR to initiate descriptor fetch.
8 Set DMAC.DMAE=1 and expected DCSn.CTE=1 to launch DAM transfer.
9 Hardware clears the corresponding bit in DDR as soon as it starts to fetch the descriptor.
10 Waits for dma request from peripherals to start dma transfer.
11 After DMAC completes the current descriptor dma transfer, if DES0.Link=0, it sets DCSn.TT to 1. If the interrupt enabled, it will generates the corresponding interrupts.
12 If DES0.LINK=1, after DMAC completes the current descriptor dma transfer and return to fetch the next descriptor and continues dma transfer until completes the descriptor dma transfer which DES0.LINK=0.
13 When transfer end, clr DCSn.CTE to 0 to close the channel, and then clear DCSn.TT bits.

Table 8-3 Descriptor Structure

Word	Bit	Name	Function
1st (DES0)	31	EACKS	External DMA DACKn output polarity select
	30	EACKM	External DMA DACKn output Mode select
	29-28	ERDM	External DMA request detection Mode
	27	EOPM	External DMA End of process mode
	26	Reserved	
	25	BLAST	BCH Last (Only for BCH and Nand transfer)
	24	Reserved	
	23	SAI	Source Address Increment
	22	DAI	Target Address Increment
	21-20	Reserved	
	19-16	RDIL	Request Detection Interval Length
	15-14	SP	Source port width
	13-12	DP	Target port width
	11	Reserved	
	10-8	TSZ	Transfer Data Size
	7	NRD	Direct read nand
	6	NWR	Direct write nand
	5	NAC	Nand AL/CL from data
	4:3	Reserved	
	2	STDE	Stride transfer enable
	1	TIE	Transfer Interrupt Enable
	0	LINK	Descriptor Link Enable
2nd (DES1)	31-0	DSA	Source Address
3rd (DES2)	31-0	DTA	Target Address
4th (DES3)	31-24	DOA	Descriptor Offset address
	23-0	DTC	Transfer Counter
5th (DES4)	31-16	TSD	Target Stride Address
	15-0	SSD	Source Stride Address
6th(DES5)	31-16	Reserved	
	15:6	Reserved	
	5-0	DRT	DMA Request Type
7th(DES6)	31-23	DNTE1	Nand request 1 timer enable
	30:23	DNT1	Nand request 1 detect timer
	22-16	DTCT	Nand tail counter
	15	DNTE	Nand request 0 detect timer enable
	14-6	Reserved	
	5-0	DNT	Nand request 0 detect timer
8th(DES7)	31-4	DDA	Next descriptor address
	3-0	Reserved	

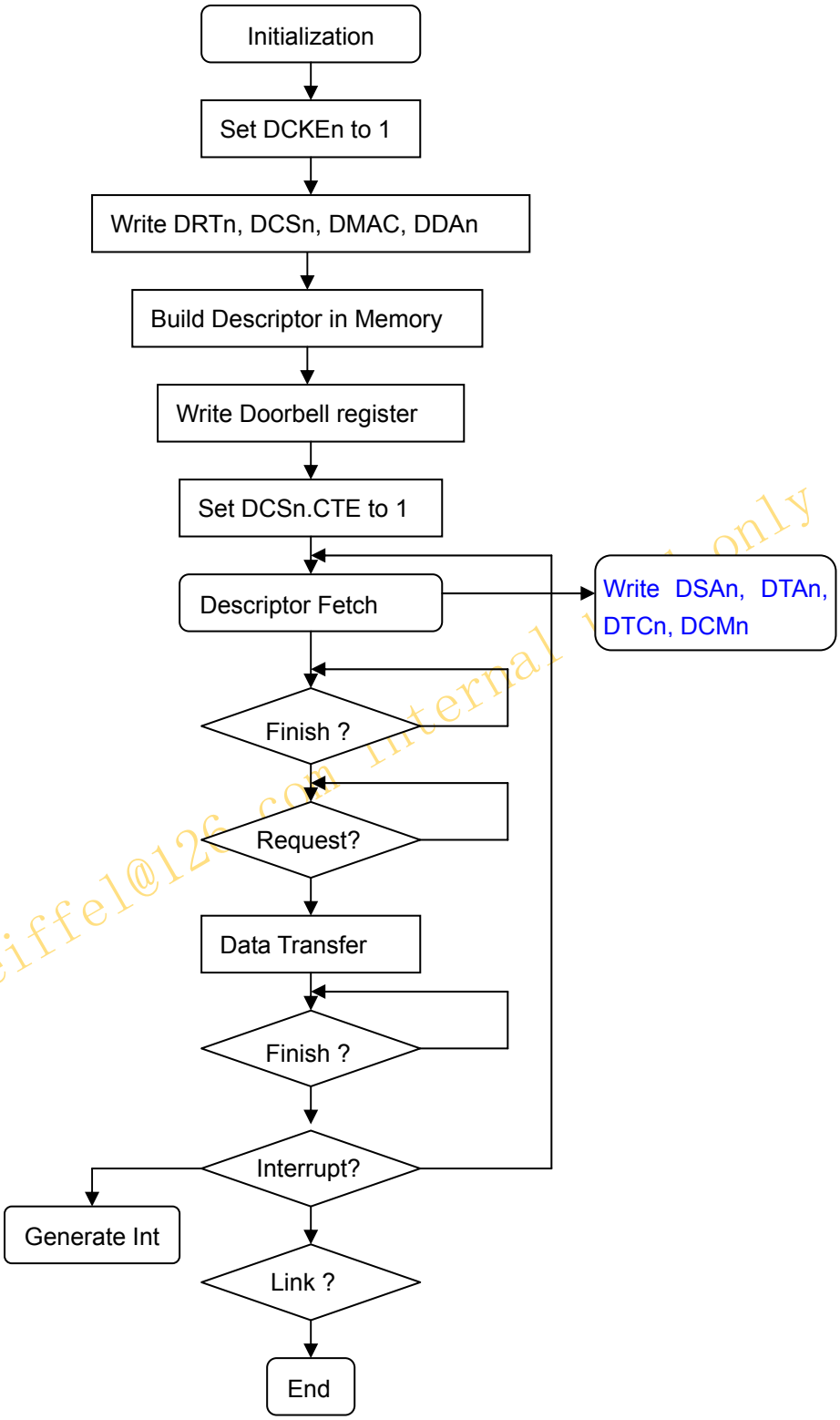


Figure 8-1 Descriptor Transfer Flow

8.3.1.2 Stride Address Transfer

During transfer, source or target address can be not continuous and the source and target stride offset address are showed in DSDn registers.

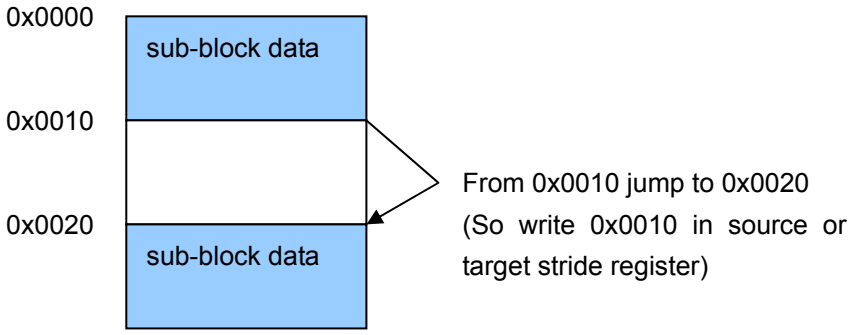


Figure 8-2 Example for Stride Address Transfer

8.3.1.3 BCH DMA Transfer

Channel 0 supports BCH DMA transfer.

During BCH encoding, DMA read data from memory pointed by DSAR0 and write to BCH data register BHDR, after BCH encoding finishes, DMA write BHINT and BCH parity data BHPAR0~9 respectively to memory pointed by DTAR0, and then DMA clear BHINT and set BCH reset to BCH automatically.

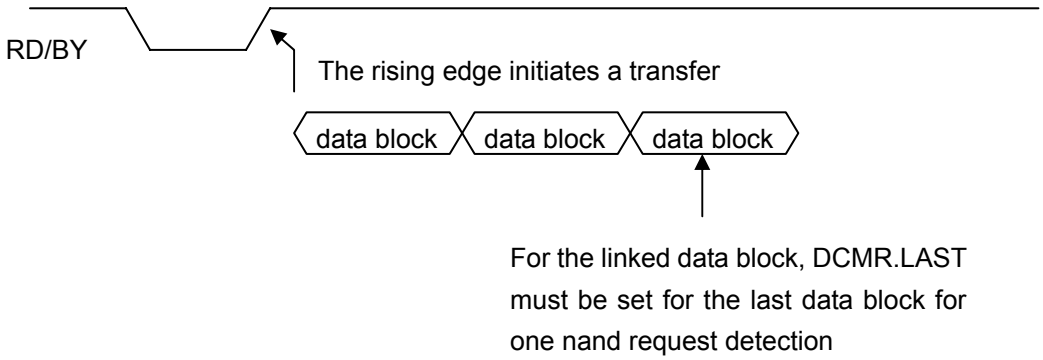
During BCH decoding, DMA read data from memory pointed by DSAR0 and write to BCH data register BHDR, after BCH decoding finishes, if there is error in the data block, DMA will write BHINT, BHERR0~11 to memory pointed by DTAR0 or if there is no error in the data block, DMA will only write BHINT to memory, and then DMA clear BHINT and set BCH reset to BCH. If multiple data block are linked to wait for BCH decoding, data transfer and decoding can be executed in pipeline, that is when the first data block is being decoding, and second data can be transfer to BCH for syndrome generation.

Here one data block means, for encoding, the entire data bytes need encoding, for decoding, the entire data bytes and parity bytes need decoding. DCM.BLAST must be used in descriptor BCH transfer. When one data block is in a continuous memory space, BLAST must be set to 1 for this data block; when one data block is linked in multiple data space, BLAST must be set to 1 for the last data space.

8.3.1.4 Nand Transfer

Two ways are for nand RB detect.

One way is to detect RD/BY rising edge as the following waveform.



The other way is using DNT register. When the rising edge is missed by DMA, DNT timer also can be used for nand RB detect. The timer is used to detect the high level duration of RD/BY signal, when the high level keep high longer than DNT counting periods, then nand transfer request generates.

DCMR1.DNT and DCMR1.DNTE are for nand request 0. DCMR1.DNT1 and DCMR1.DNTE1 are for nand request 1.

8.3.2 No-Descriptor Transfer

To do proper DMA transfer, do as following steps:

2 First of all, check whether the status of DMA controller are available, that is, for global control (DMAC), ensure that DMAC.AR=0 and DMAC.HLT=0; while for expected channels, ensure that DCSn.AR=0, DCSn.HLT=0 and DCSn.TT=0 and DTCn=0.

3 For each channel n, initialize DSAn, DTAn, DTCn, DRTn, DCSn, DCMn properly.

4 Set DMAC.DMAE=1 and expected DCSn.CTE=1 and DCSn.NDES=1 to launch DAM transfer.

For a channel with auto-request (DRTn.RT=0x8), the transfer begins automatically when the DCSn.CTE bit and DMAC.DMAE bit are set to 1. While for a channel with other request types, the transfer does not start until a transfer request is issued and detected.

For any channel n, The DTCn value is decremented by 1 for each successful transaction of a data unit. When the specified number of transfer data unit has been completed (DTCn = 0), the transfer ends normally. Meanwhile corresponding bit of DIRQP is set to 1. If DCMn.TIE bit is set to 1, an interrupt request is sent to the CPU. However, during the transfer, if a DMA address error occurs, the transfer is suspended, both DCSn.AR and DMAC.AR are set to 1 as well as corresponding bit of DIRQP. Then an interrupt request is sent to the CPU despite of DCMn.TIE.

Sometimes, for example, an UART parity error occurs for a channel that is transferring data between such UART and another terminal. In the case, both DCSn.HLT and DMAC.HLT are set to 1 and the transfer is suspended. Software should identify halt status by checking such two bits and re-configure

DMA to let DMA rerun properly later.

For non-descriptor BCH transfer, there is no pipeline execution for BCH decoding. DCM.BLAST doesn't need to be set in non-descriptor BCH transfer.

8.4 DMA Requests

DMA transfer requests are normally generated from either the data transfer source or target, but also they can be issued by on-chip peripherals that are neither the source nor the target. There are two DMA transfer request types: auto-request, and on-chip peripheral request. For any channel n, its transfer request type is determined through DRTn.

8.4.1 Auto Request

When there is no explicit transfer request signal available, for example, memory-to-memory transfer or memory to some on-chip peripherals like GPIO, the auto-request mode allows the DMA to automatically generate a transfer request signal internally. Therefore, when DMA initialization done, once the DMAC.DMAE and DCSn.CTE are set to 1, the transfer begins immediately in channel n which DRTn=0x8.

8.4.2 On-Chip Peripheral Request

In the mode, transfer request signals come from on-chip peripherals. All request types except 0x8 (value of DRT) belong to the mode. **NOTE:** the transfer byte number for one request detection according to DCMn.RDIL must be equal or less than the byte number according to receive or transmit trigger value of source or target devices.

8.5 Channel Priorities

There are two dma cores, each one supports 6 channels dma transfer. The two cores have the same priority.

In each core, there are two sets: set 1 has the higher priority than set 2, within each set priority is round robin.

Table 8-4 Relationship among DMA transfer connection, request mode & transfer mode

Transfer Connection	Request Mode	Transfer Mode	Data Size (bits)	Channel
External memory or memory-mapped external device and on-chip peripheral module	Auto on-chip	Single	8/16/32 16-byte/32-byte	0~5

8.6 Examples

8.6.1 Memory-to-memory auto request No-Descriptor Transfer

Suppose you want to do memory move between two different memory regions through channel 3, for example, moving 1KB data from address 0x20001000 to 0x20011000, do as following steps:

1 Check if (DMAC.AR==0 && DMAC.HLT==0 && DCS3.AR==0 && DCS3.HLT==0 && DCS3.CT==0 && DCS3.NDES=1 && DTC3==0).
2 If above condition is true, set value 0 to DCS3.CTE to disable the channel 3 temporarily.
3 Set source address 0x20001000 to DSA3 and target address 0x20011000 to DTA3.
4 Suppose the data unit is word, set transfer count number 256 (1024/4) to DTC3.
5 Set auto-request (0x8) to DRT3.
6 Up to now, only the most important channel control register DCM3 is left, set it carefully:
   – Set value 1 to SAI and DAI[*1].
   – Ignore RDIL because in the case there is no explicit request signal can be detected.
   – Set word size (0) to SP and DP[*2].
   – Set value 1 to TIE to let CPU do some post process after the transfer done.
7 Set value 1 to DCS3.CTE and DMAC.DMAE to launch the transfer in channels 3.
8 When the transfer terminates normally (DTC3==0 && DCS3.TT==1), DIRQP.CIRQ3 will automatically be set value 1 and an interrupt request will be sent to CPU.
9 When CPU grants the interrupt request, in the corresponding IRQ handler, software must clear the DCS3.CT to value 0, and the behavior will automatically clear DIRQP.CIRQ3.

NOTES:

1 Either source or target is a FIFO, must not enable corresponding address increment.
2 When either source or target need be accessed through EMC (external memory controller), the real port with of the device is encapsulated by EMC, so you can set any favorite port with for it despite of the real one.

9 DMA Controller

DMA controller (DMAC) is dedicated to transfer data between on-chip peripherals (MSC, AIC, UART, etc.), external memories, and memory-mapped external devices.

9.1 Features

- Support up to 12 independent DMA channels
- Two independent DMA core, each supports 6 channels
- Descriptor or No-Descriptor Transfer
- Transfer data units: byte, 2-byte (half word), 4-byte (word), 16-byte, 32-byte or 64-byte
- Transfer number of data unit: 1 ~ 224
- Independent source and target port width: 8-bit, 16-bit, 32-bit
- Two channel priority modes: fixed, round robin

9.2 Register Descriptions

Table 9-1 DMAC Registers

Name	Description	RW	Reset Value	Address	Access Size (bit)
DSA0	DMA Source Address 0	RW	0x0	0x13420000	32
DTA0	DMA Target Address 0	RW	0x0	0x13420004	32
DTC0	DMA Transfer Count 0	RW	0x0	0x13420008	32
DRT0	DMA Request Source 0	RW	0x0	0x1342000C	32
DCS0	DMA Channel Control/Status 0	RW	0x0	0x13420010	32
DCM0	DMA Command 0	RW	0x0	0x13420014	32
DDA0	DMA Descriptor Address 0	RW	0x0	0x13420018	32
DSD0	DMA Stride Address 0	RW	0x0	0x1342001C	32
DSA1	DMA Source Address 1	RW	0x0	0x13420020	32
DTA1	DMA Target Address 1	RW	0x0	0x13420024	32
DTC1	DMA Transfer Count 1	RW	0x0	0x13420028	32
DRT1	DMA Request Source 1	RW	0x0	0x1342002C	32
DCS1	DMA Channel Control/Status 1	RW	0x0	0x13420030	32
DCM1	DMA Command 1	RW	0x0	0x13420034	32
DDA1	DMA Descriptor Address 1	RW	0x0	0x13420038	32
DSD1	DMA Stride Address 1	RW	0x0	0x1342003C	32
DSA2	DMA Source Address 2	RW	0x0	0x13420040	32
DTA2	DMA Target Address 2	RW	0x0	0x13420044	32
DTC2	DMA Transfer Count 2	RW	0x0	0x13420048	32
DRT2	DMA Request Source 2	RW	0x0	0x1342004C	32
DCS2	DMA Channel Control/Status 2	RW	0x0	0x13420050	32
DCM2	DMA Command 2	RW	0x0	0x13420054	32
DDA2	DMA Descriptor Address 2	RW	0x0	0x13420058	32
DSD2	DMA Stride Address 2	RW	0x0	0x1342005C	32
DSA3	DMA Source Address 3	RW	0x0	0x13420060	32
DTA3	DMA Target Address 3	RW	0x0	0x13420064	32
DTC3	DMA Transfer Count 3	RW	0x0	0x13420068	32
DRT3	DMA Request Source 3	RW	0x0	0x1342006C	32
DCS3	DMA Channel Control/Status 3	RW	0x0	0x13420070	32
DCM3	DMA Command 3	RW	0x0	0x13420074	32
DDA3	DMA Descriptor Address 3	RW	0x0	0x13420078	32
DSD3	DMA Stride Address 3	RW	0x0	0x1342007C	32
DSA4	DMA Source Address 4	RW	0x0	0x13420080	32
DTA4	DMA Target Address 4	RW	0x0	0x13420084	32
DTC4	DMA Transfer Count 4	RW	0x0	0x13420088	32

DRT4	DMA Request Source 4	RW	0x0	0x1342008C	32
DCS4	DMA Channel Control/Status 4	RW	0x0	0x13420090	32
DCM4	DMA Command 4	RW	0x0	0x13420094	32
DDA4	DMA Descriptor Address 4	RW	0x0	0x13420098	32
DSD4	DMA Stride Address 4	RW	0x0	0x1342009C	32
DSA5	DMA Source Address 5	RW	0x0	0x134200A0	32
DTA5	DMA Target Address 5	RW	0x0	0x134200A4	32
DTC5	DMA Transfer Count 5	RW	0x0	0x134200A8	32
DRT5	DMA Request Source 5	RW	0x0	0x134200AC	32
DCS5	DMA Channel Control/Status 5	RW	0x0	0x134200B0	32
DCM5	DMA Command 5	RW	0x0	0x134200B4	32
DDA5	DMA Descriptor Address 5	RW	0x0	0x134200B8	32
DSD5	DMA Stride Address 5	RW	0x0	0x134200BC	32
DSA6	DMA Source Address 6	RW	0x0	0x13420100	32
DDA6	DMA Target Address 6	RW	0x0	0x13420104	32
DTC6	DMA Transfer Count 6	RW	0x0	0x13420108	32
DRT6	DMA Request Source 6	RW	0x0	0x1342010C	32
DCS6	DMA Channel Control/Status 6	R/W	0x0	0x13420110	32
DCM6	DMA Command 6	RW	0x0	0x13420114	32
DDA6	DMA Descriptor Address 6	RW	0x0	0x13420118	32
DSD6	DMA Stride Address 6	RW	0x0	0x1342011C	32
DSA7	DMA Source Address 7	RW	0x0	0x13420120	32
DDA7	DMA Target Address 7	RW	0x0	0x13420124	32
DTC7	DMA Transfer Count 7	RW	0x0	0x13420128	32
DRT7	DMA Request Source 7	RW	0x0	0x1342012C	32
DCS7	DMA Channel Control/Status 7	R/W	0x0	0x13420130	32
DCM7	DMA Command 7	RW	0x0	0x13420134	32
DDA7	DMA Descriptor Address 7	RW	0x0	0x13420138	32
DSD7	DMA Stride Address 7	RW	0x0	0x1342013C	32
DSA8	DMA Source Address 8	RW	0x0	0x13420140	32
DDA8	DMA Target Address 8	RW	0x0	0x13420144	32
DTC8	DMA Transfer Count 8	RW	0x0	0x13420148	32
DRT8	DMA Request Source 8	RW	0x0	0x1342014C	32
DCS8	DMA Channel Control/Status 8	R/W	0x0	0x13420150	32
DCM8	DMA Command 8	RW	0x0	0x13420154	32
DDA8	DMA Descriptor Address 8	RW	0x0	0x13420158	32
DSD8	DMA Stride Address 8	RW	0x0	0x1342015C	32
DSA9	DMA Source Address 9	RW	0x0	0x13420160	32
DDA9	DMA Target Address 9	RW	0x0	0x13420164	32
DTC9	DMA Transfer Count 9	RW	0x0	0x13420168	32
DRT9	DMA Request Source 9	RW	0x0	0x1342016C	32

DCS9	DMA Channel Control/Status 9	R/W	0x0	0x13420170	32
DCM9	DMA Command 9	RW	0x0	0x13420174	32
DDA9	DMA Descriptor Address 9	RW	0x0	0x13420178	32
DSD9	DMA Stride Address 9	RW	0x0	0x1342017C	32
DSA10	DMA Source Address 10	RW	0x0	0x13420180	32
DDA10	DMA Target Address 10	RW	0x0	0x13420184	32
DTC10	DMA Transfer Count 10	RW	0x0	0x13420188	32
DRT10	DMA Request Source 10	RW	0x0	0x1342018C	32
DCS10	DMA Channel Control/Status 10	R/W	0x0	0x13420190	32
DCM10	DMA Command 10	RW	0x0	0x13420194	32
DDA10	DMA Descriptor Address 10	RW	0x0	0x13420198	32
DSD10	DMA Stride Address 10	RW	0x0	0x1342019C	32
DSA11	DMA Source Address 11	RW	0x0	0x134201A0	32
DDA11	DMA Target Address 11	RW	0x0	0x134201A4	32
DTC11	DMA Transfer Count 11	RW	0x0	0x134201A8	32
DRT11	DMA Request Source 11	RW	0x0	0x134201AC	32
DCS11	DMA Channel Control/Status 11	R/W	0x0	0x134201B0	32
DCM11	DMA Command 11	RW	0x0	0x134201B4	32
DDA11	DMA Descriptor Address 11	RW	0x0	0x134201B8	32
DSD11	DMA Stride Address 11	RW	0x0	0x134201BC	32
DMAC1	DMA Control 1 Register	R/W	0x0	0x13420300	32
DIRQP1	DMA Interrupt Pending 1	R	0x0	0x13420304	32
DDR1	DMA Doorbell 1 Register	RW	0x0	0x13420308	32
DDRS1	DMA Doorbell Set 1 Register	W	0x0	0x1342030C	32
DCKE1	DMA Clock Enable 1 Register	W	0x0	0x13420310	32
DCKES1	DMA Clock Enable 1 Set Register	W	0x0	0x13420314	32
DCKEC1	DMA Clock Enable 1 Clear Register	W	0x0	0x13420318	32
DMAC2	DMA Control 2 Register	R/W	0x0	0x13420400	32
DIRQP2	DMA Interrupt Pending 2	R	0x0	0x13420404	32
DDR2	DMA Doorbell 2 Register	RW	0x0	0x13420408	32
DDRS2	DMA Doorbell Set Register	W	0x0	0x1342040C	32
DCKE2	DMA Clock Enable 2 Register	W	0x0	0x13420410	32
DCKES2	DMA Clock Enable 2 Set Register	W	0x0	0x13420414	32
DCKEC2	DMA Clock Enable 2 Clear Register	W	0x0	0x13420418	32

9.2.1 DMA Source Address (DSAn, n = 0 ~ 11)

DSA0, DSA1, DSA2, 0x13420000, 0x13420020, 0x13420040,
DSA3, DSA4, DSA5, 0x13420060, 0x13420080, 0x134200A0,
DSA6, DSA7, DSA8, 0x13420100, 0x13420120, 0x13420140,
DSA9, DSA10, DSA11 0x13420160, 0x13420180, 0x134201A0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SA																															
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
31:0	SA	Source physical address.	RW

9.2.2 DMA Target Address (DTAn, n = 0 ~ 11)

DTA0, DTA1, DTA2, 0x13420004, 0x13420024, 0x13420044,
DTA3, DTA4, DTA5, 0x13420064, 0x13420084, 0x134200A4,
DTA6, DTA7, DTA8, 0x13420104, 0x13420124, 0x13420144,
DTA9, DTA10, DTA11 0x13420164, 0x13420184,0x134201A4

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TA																															
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
31:0	TA	Target physical address.	RW

9.2.3 DMA Transfer Count (DTCn, n = 0 ~ 11)

DTC0, DTC1, DTC2, 0x13420008, 0x13420028, 0x13420048,
DTC3, DTC4, DTC5, 0x13420068, 0x13420088, 0x134200A0,
DTC6, DTC7, DTC8, 0x13420108, 0x13420128, 0x13420148,
DTC9, DTC10, DTC11 0x13420168, 0x13420188, 0x134201A8

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																TC															
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
31:24	Reserved	Write has no effect, read as zero.	R
23:0	TC	When Stride address transfer is disabled: TC hold the number of data unit to transfer and it counts down to 0 at the end. When Stride address transfer is enabled: TC composes of two parts: The lower 16 bits: the number of data unit for sub-block transfer The higher 8 bits: the number of sub-block And both the two parts count down to 0 at the end.	RW

9.2.4 DMA Request Types (DRTn, n = 0 ~ 11)

DRT0, DRT1, DRT2, **0x1342000c, 0x1342002c, 0x1342004c,**
DRT3, DRT4, DRT5, **0x1342006c, 0x1342008c, 0x134200Ac,**
DRT6, DRT7, DRT8, **0x1342010c, 0x1342012c, 0x1342014c,**
DRT9, DRT10, DRT11 **0x1342016c, 0x1342018c, 0x134201Ac**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																										RT					
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
31:6	Reserved	Write has no effect, read as zero.	R
5:0	RT	Transfer request type.	RW

Table 9-2 Transfer Request Types

RT5-0	Description
000000	Reserved.
000001	Reserved.
000010	Reserved.
000011	Reserved.
000100	Reserved.
000101	Reserved.
000110	Reserved.
000111	Reserved.
001000	Auto-request. (ignore RDIL3-0, external address → external address)
001001	TSSI receive-fifo-full transfer request. (TS fifo → external address)
001010	Reserved.

001011	Reserved.
001100	External request with DREQn. (external address ←→ external device with DACKn)
001101	Reserved.
001110	UART3 transmit-fifo-empty transfer request. (external address → UTHR)
001111	UART3 receive-fifo-full transfer request. (URBR → external address)
010000	UART2 transmit-fifo-empty transfer request. (external address → UTHR)
010001	UART2 receive-fifo-full transfer request. (URBR → external address)
010010	UART1 transmit-fifo-empty transfer request. (external address → UTHR)
010011	UART1 receive-fifo-full transfer request. (URBR → external address)
010100	UART0 transmit-fifo-empty transfer request. (external address → UTHR)
010101	UART0 receive-fifo-full transfer request. (URBR → external address)
010110	SSI transmit-fifo-empty transfer request.
010111	SSI receive-fifo-full transfer request.
011000	AIC transmit-fifo-empty transfer request.
011001	AIC receive-fifo-full transfer request.
011010	MSC transmit-fifo-empty transfer request.
011011	MSC receive-fifo-full transfer request.
011100	TCU channel n. (overflow interrupt, external address→external address space)
011101	SADC transfer request. (SADC → external address)
011110	MSC1 transmit-fifo-empty transfer request.
011111	MSC1 receive-fifo-full transfer request.
100000	SSI1 transmit-fifo-empty transfer request.
100001	SSI1 receive-fifo-full transfer request.
100010	PM transmit-fifo-empty transfer request.
100011	PM receive-fifo-full transfer request.
100100	MSC2 transmit-fifo-empty transfer request.
100101	MSC2 receive-fifo-full transfer request.
101000	I2C transmit-fifo-empty transfer request.
101001	I2C receive-fifo-full transfer request.
101010	I2C1 transmit-fifo-empty transfer request.
101011	I2C1 receive-fifo-full transfer request.
101100	Reserved.
101101	Reserved.
111110	I2C1 transmit-fifo-empty transfer request.
111111	I2C1 receive-fifo-full transfer request.
Other	Reserved.

NOTES:

1 Only auto request can be concurrently selected in all channels with different source and target address.

2 For on-chip device DMA request except TCU, the corresponding source or target address that map to on-chip device must be set as fixed.

9.2.5 DMA Channel Control/Status (DCSn, n = 0 ~ 11)

DCS0, DCS1, DCS2, **0x13420010, 0x13420030, 0x13420050,**
DCS3, DCS4, DCS5, **0x13420070, 0x13420090, 0x134200B0**
DCS6, DCS7, DCS8, **0x13420110, 0x13420130, 0x13420150,**
DCS9, DCS10, DCS11 **0x13420170, 0x13420190, 0x134201B0**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	NDES	DES8	Reserved						CDOA								Reserved											AR	TT	HLT	Reserved	CTE
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
31	NDES	Descriptor or No-Descriptor Transfer Select. 0: Descriptor Transfer; 1: No-descriptor Transfer.	RW
30	DES8	Descriptor 8 Word. 0: 4-word descriptor; 1: 8-word descriptor.	RW
29:24	Reserved	Write has no effect, read as zero.	R
23:16	CDOA	Copy of offset address of last completed descriptor from that in DMA command register. Software could know which descriptor is just completed combining with count terminate interrupt resulted by DCSn.CT. (Ignored in No-Descriptor Transfer)	RW
15:5	Reserved	Write has no effect, read as zero.	R
4	AR	Address Error. 0: no address error; 1: address error.	RW
3	TT	Transfer Terminate. 0: No-Link Descriptor or No-Descriptor DMA transfer does not end 1: No-Link Descriptor or No-Descriptor DMA transfer end	RW
2	HLT	DMA halt. 0: DMA transfer is in progress; 1: DMA halt.	RW
1	Reserved	Write has no effect, read as zero.	R
0	CTE	Channel transfer enable. 0: disable; 1: enable.	RW

9.2.6 DMA Channel Command (DCMn, n = 0 ~ 11)

DCM0, DCM1, DCM2, **0x13420014, 0x13420034, 0x13420054,**
DCM3, DCM4, DCM5, **0x13420074, 0x13420094, 0x134200B4,**
DCM6, DCM7, DCM8, **0x13420114, 0x13420134, 0x13420154,**
DCM9. DCM10, DCM11 **0x13420174, 0x13420194, 0x134201B4**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	EACKS	EACKM	ERDM		Reserved				SAI	DAI	Reserved		RDIL				SP		DP		Reserved		TSZ		Reserved					STDE	TIE	LINK
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
31	EACKS	External DACK Output Level Select. 0: active high; 1: active low.	RW
30	EACKM	External DACK Output Mode Select. 0: output in read cycle; 1: output in write cycle.	RW
29:28	ERDM	External DREQ Detection Mode Select. 00: Low level detection 01: Falling edge detection 10: High level detection 11: Rising edge detection	RW
27:24	Reserved	Write has no effect, read as zero.	R
23	SAI	Source Address Increment. 0: no increment; 1: increment.	RW
22	DAI	Target Address Increment. 0: no increment; 1: increment.	RW
19:16	RDIL	Request Detection Interval Length. Set the number of transfer unit between two requests detection in single mode. Please refer to following Table 9-3.	RW
15:14	SP	Source port width. 00: 32-bit; 01: 8-bit; 10: 16-bit; 11: reserved.	RW
13:12	DP	Target port width. 00: 32-bit; 01: 8-bit; 10: 16-bit; 11: reserved.	RW
11	Reserved	Write has no effect, read as zero.	R
10:8	TSZ	Transfer Data Size of a data unit. 000: 32-bit; 001: 8-bit; 010: 16-bit; 011: 16-byte; 100: 32-byte; 101: 64-byte; others: reserved.	RW
7:3	Reserved	Write has no effect, read as zero.	R
2	STDE	Stride Disable/Enable. 0: address stride disable; 1: address stride enable.	RW
1	TIE	Transfer Interrupt Enable (TIE).	RW

		0: disable interrupt; 1: enable interrupt when TT is set to 1.	
0	LINK	Descriptor Link Enable. 0: disable; 1: enable. (Ignored in No-Descriptor Transfer)	RW

Table 9-3 Detection Interval Length

RDIL	Description
0	Interval length is 0.
1	Interval length is 2 transfer unit.
2	Interval length is 4 transfer unit.
3	Interval length is 8 transfer unit.
4	Interval length is 12 transfer unit.
5	Interval length is 16 transfer unit.
6	Interval length is 20 transfer unit.
7	Interval length is 24 transfer unit.
8	Interval length is 28 transfer unit.
9	Interval length is 32 transfer unit.
10	Interval length is 48 transfer unit.
11	Interval length is 60 transfer unit.
12	Interval length is 64 transfer unit.
13	Interval length is 124 transfer unit.
14	Interval length is 128 transfer unit.
15	Interval length is 200 transfer unit.

9.2.7 DMA Descriptor Address (DDAn, n = 0 ~ 11)

This register is ignored in No-Descriptor Transfer.

DDA0, DDA1, DDA2, **0x13420018, 0x13420038, 0x13420058,**
DDA3, DDA4, DDA5, **0x13420078, 0x13420098, 0x134200B8,**
DDA6, DDA7, DDA8, **0x13420118, 0x13420138, 0x13420158,**
DDA9, DDA10, DDA11 **0x13420178, 0x13420198, 0x134201B8**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DBA																				DOA								Reserved			
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
31:12	DBA	Descriptor Base Address.	RW
11:4	DOA	Descriptor Offset Address.	RW
3:0	Reserved	Write has no effect, read as zero.	R

9.2.8 DMA Stride Address (DSDn, n = 0 ~ 11)

This register is ignored in No-Descriptor Transfer.

When address stride transfer is enabled in Descriptor mode, after a sub-block defined in DTCRn is finished transferring, the source or target stride address will be added up to the corresponding source or target address and the transfer will keep going until the transfer ends which means TC in DTCRn reach 0.

DSD0, DSD1, DSD2, **0x1342001C, 0x1342003C, 0x1342005C,**
DSD3, DSD4, DSD5, **0x1342007C, 0x1342009C, 0x134200BC**
DSD6, DSD7, DSD8, **0x1342011C, 0x1342013C, 0x1342015C,**
DSD9, DSD10, DSD11 **0x1342017C, 0x1342019C, 0x134201BC**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TSD																SSD															
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
31:16	TSD	Target Stride Address.	RW
15:0	SSD	Source Stride Address.	RW

9.2.9 DMA Control

DMAC1 controls channel 0~5 and DMAC2 controls channel 6~11.

DMAC1 **0x13420300**
DMAC2 **0x13420400**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FMSC	FSSI	FTSSI	FUART	FAIC	Reserved																	PM	Reserved					HLT	AR	Reserved	DMAE
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
31	FMSC	MSC Fast DMA mode. 0: normal DMA transfer; 1: fast DMA transfer.	RW
30	FSSI	SSI Fast DMA mode. 0: normal DMA transfer; 1: fast DMA transfer.	RW
29	FTSSI	TSSI Fast DMA mode. 0: normal DMA transfer; 1: fast DMA transfer.	RW
28	FUART	UART Fast DMA mode.	RW

		0: normal DMA transfer; 1: fast DMA transfer.	
27	FAIC	AIC Fast DMA mode. 0: normal DMA transfer; 1: fast DMA transfer.	RW
26:10	Reserved	Write has no effect, read as zero.	R
9:8	PM	Channel priority mode. 00: CH0, CH1 > CH2, CH3, CH4 01: CH1, CH2 > CH0, CH3, CH4 10: CH2, CH3 > CH0, CH1, CH4 11: CH3, CH4 > CH0, CH1, CH2 For example, when PM == 2'b00, it means set1 includes ch0 and ch1 and set2 includes ch2~ch4, set 1 has the higher priority than set 2, within one set, channel priority is round robin, that is: ch0→ch1→ch2→ch0→ch1→ch3→ch0→ch1→ch4→ch0→ch1	RW
7:4	Reserve	Write has no effect, read as zero.	R
3	HLT	Global halt status, halt occurs in any channel, the bit should set to 1. 0: no halt 1: halt occurred	RW
2	AR	Global address error status, address error occurs in any channel, the bit should be set to 1. 0: no address error 1: address error occurred	RW
1	Reserved	Write has no effect, read as zero.	R
0	DMAE	Global DMA transfer enable. 0: disable DMA channel transfer 1: enable DMA channel transfer	RW

NOTE: FMSC/FSSI/FTSSI/FUART/FAIC bit either in DMAC1 or in DMAC2 is set, the corresponding dma transfer for MSC(MSC1, MSC2), SSI(SSI1), UART0~3, AIC is in fast dma mode.

9.2.10 DMA Interrupt Pending (DIRQP)

DMAC supports total 12 pending interrupt, 6 of them are in DIRQP and the other 6 are in DIRQP2.

DIRQP **0x13420304**

DIRQP2 **0x13420404**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																										CIRQ5	CIRQ4	CIRQ3	CIRQ2	CIRQ1	CIRQ0
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
31:6	Reserved	Write has no effect, read as zero.	R
5:0	CIRQn	CIRQn (n=0~5) denotes pending status for corresponding channel. 0:no abnormal situation or normal DMA transfer is in progress 1: abnormal situation occurred or normal DMA transfer done	RW

9.2.11 DMA Doorbell (DDR)

DDR supports channel 0~5 and DDR2 supports channel 6~11.

DDR **0x13420308**

DDR2 **0x13420408**

Bit	31:6	5	4	3	2	1	0
	Reserved	DB5	DB4	DB3	DB2	DB1	DB0
RST	0	0	0	0	0	0	0

Bits	Name	Description	RW
31:6	Reserved	Write has no effect, read as zero.	R
5:0	DBn	DMA Doorbell for each channel, n=0~5, for example DB0 is for DMA channel 0. Software set it to 1 and hardware clears it to 0. 0: disable DMA controller to fetch the first descriptor or DMA controller clears it to 0 as soon as it starts to fetch the descriptor 1: Write 1 to DDS will set the corresponding DBn bit to 1 and enable DMA controller to fetch the first descriptor For example, write 0x00000001 to DDS, DB0 bit is set to 1 and enable DMA channel 0 to fetch the first descriptor. Write 0 to DDS, no meaning.	R

9.2.12 DMA Doorbell Set (DDRS)

DDRS supports channel 0~5 and DDRS2 supports channel 6~11.

DDRS **0x1342030c**

DDRS2 **0x1342040c**

Bit	31:6	5	4	3	2	1	0
	Reserved	DBS5	DBS4	DBS3	DBS2	DBS1	DBS0
RST	0	0	0	0	0	0	0

Bits	Name	Description	RW
31:6	Reserved	Write has no effect, read as zero.	R
5:0	DBSn	DMA Doorbell Set for each channel. 0: ignore 1: Set the corresponding DBn bit to 1	W

9.2.13 DMA Clock Enable (DCKE)

DCKE supports channel 0~5 and DCKE2 supports channel 6~11.

DCKE **0x13420310**

DCKE2 **0x13420410**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																										DCKE5	DCKE4	DCKE3	DCKE2	DCKE1	DCKE0
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
31:6	Reserved	Write has no effect, read as zero.	R
5:0	DCKEn	DMA Clock Enable for each channel. 0: ignore 1: Set the corresponding DCKEn bit to 1	RW

9.2.14 DMA Clock Enable Set (DCKES)

DCKES supports channel 0~5 and DCKES2 supports channel 6~11.

DCKES **0x13450314**

DCKES2 **0x13450414**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																										DCKES5	DCKES4	DCKES3	DCKES2	DCKES1	DCKES0
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
31:3	Reserved	Write has no effect, read as zero.	R
2:0	DCKESn	DMA Clock Enable Set for each channel. 0: ignore 1: Set the corresponding DCKESn bit to 1 to enable corresponding channel clock	W

9.2.15 DMA Clock Clear Set (DCKEC)

DCKEC supports channel 0~5 and DCKEC2 supports channel 6~11.

DCKEC **0x13450318**

DCKEC2 **0x13450418**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																										DCKEC5	DCKEC4	DCKEC3	DCKEC2	DCKEC1	DCKEC0
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
31:3	Reserved	Write has no effect, read as zero.	R
2:0	DCKECn	DMA Clock Enable Clear for each channel. 0: ignore 1: Set the corresponding DCKECn bit to 1 to disable corresponding channel clock	W

9.3 DMA manipulation

9.3.1 Descriptor Transfer

9.3.1.1 Normal Transfer

To do proper Descriptor DMA transfer, do as following steps:

1 First of all, open channel clock by setting DCKESn register for corresponding channel.

2 Check whether the status of DMA controller are available, that is, for global control (DMAC), ensure that DMAC.AR=0 and DMAC.HLT=0; while for expected channels, ensure that DCSn.AR=0, DCSn.HLT=0, DCSn.TT=0, DTCn=0 and DCSn.INV=0.

3 Select 4 word or 8 word descriptor by DCSn.DES8.

4 For Descriptor transfer, guarantee DCSn.NDES=0.

5 Initiate channel request register DRTn.

6 Build descriptor in memory. Write the first descriptor address in DDAn and the address must be 16Bytes aligned in 4word descriptor and 32Bytes aligned in 8word descriptor. The descriptor address includes two parts: Base and Offset address. If the descriptor is linked, the 32-bit address of next descriptor is composed of 20-bit Base address in DDAn and 8-bit Offset address in DES3.DOA and the four LSB is 0x0. See Table 9-4 for the detailed 4-word descriptor structure.

NOTE: if stride address transfer is enabled, the address must be 32Bytes aligned because DES4 needs to read out.

7 Set 1 to the corresponding bit in DDR to initiate descriptor fetch.

8 Set DMAC.DMAE=1 and expected DCSn.CTE=1 to launch DAM transfer.

9 Hardware clears the corresponding bit in DDR as soon as it starts to fetch the descriptor.

10 Waits for dma request from peripherals to start dma transfer.

11 After DMAC completes the current descriptor dma transfer, if DES0.Link=0, it sets DCSn.TT to 1. If the interrupt enabled, it will generates the corresponding interrupts.

12 If DES0.LINK=1, after DMAC completes the current descriptor dma transfer and return to fetch the next descriptor and continues dma transfer until completes the descriptor dma transfer which DES0.LINK=0.

13 When transfer end, clr DCSn.CTE to 0 to close the channel, and then clear DCSn.TT bits.

Table 9-4 Descriptor Structure

Word	Bit	Name	Function
1st (DES0)	31	EACKS	External DMA DACKn output polarity select
	30	EACKM	External DMA DACKn output Mode select
	29-28	ERDM	External DMA request detection Mode
	27	EOPM	External DMA End of process mode
	26-24	Reserved	--
	23	SAI	Source Address Increment
	22	DAI	Target Address Increment
	21-20	Reserved	--
	19-16	RDIL	Request Detection Interval Length
	15-14	SP	Source port width
	13-12	DP	Target port width
	11	Reserved	--
	10-8	TSZ	Transfer Data Size
	7-3	Reserved	--
	2	STDE	Stride transfer enable
	1	TIE	Transfer Interrupt Enable
	0	LINK	Descriptor Link Enable
2nd (DES1)	31-0	DSA	Source Address
3rd (DES2)	31-0	DTA	Target Address
4th (DES3)	31-24	DOA	Descriptor Offset address
	23-0	DTC	Transfer Counter
5th (DES4)	31-16	TSD	Target Stride Address
	15-0	SSD	Source Stride Address
6th(DES5)	31-6	Reserved	--
	5-0	DRT	DMA Request Type
7th(DES6)	31-0	Reserved	--
8th(DES7)	31-0	Reserved	--

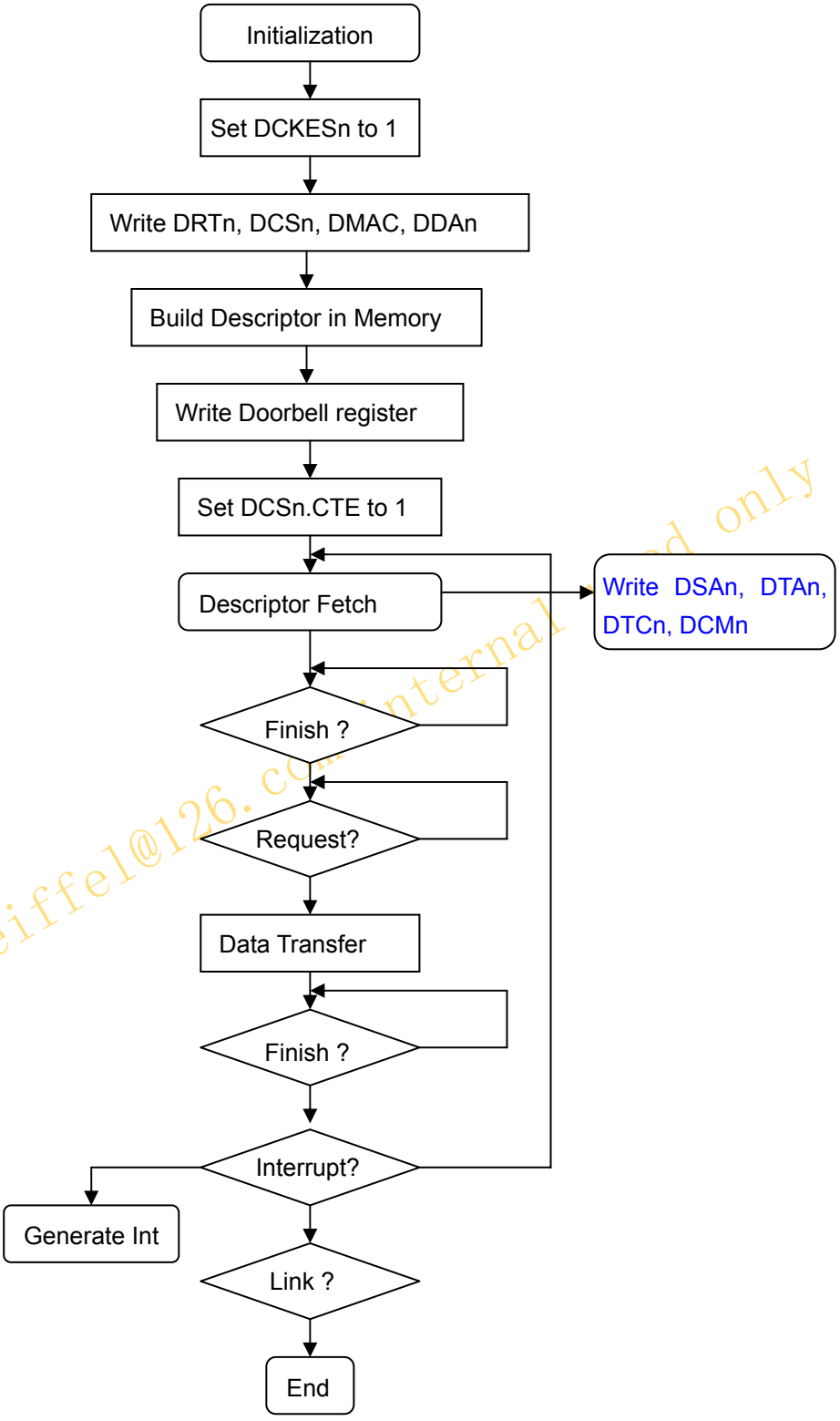


Figure 9-1 Descriptor Transfer Flow

9.3.1.2 Stride Address Transfer

During transfer, source or target address can be not continuous and the source and target stride offset address are showed in DSDn registers.

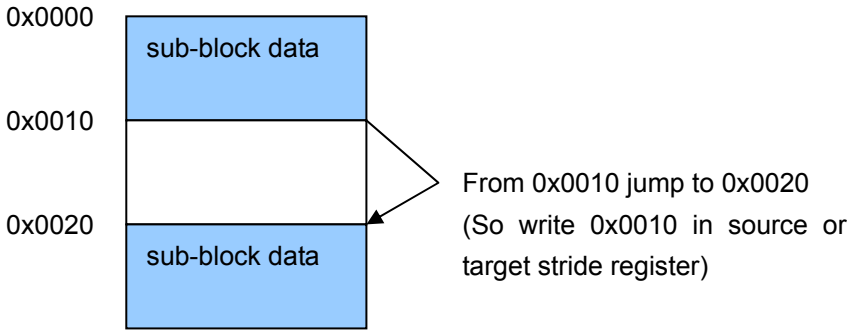


Figure 9-2 Example for Stride Address Transfer

9.3.2 No-Descriptor Transfer

To do proper DMA transfer, do as following steps:

1 First of all, check whether the status of DMA controller are available, that is, for global control (DMAC), ensure that DMAC.AR=0 and DMAC.HLT=0; while for expected channels, ensure that DCSn.AR=0, DCSn.HLT=0 and DCSn.TT=0 and DTCn=0.
2 For each channel n, initialize DSAn, DTAn, DTCn, DRTn, DCSn, DCMn properly.
3 Set DMAC.DMAE=1 and expected DCSn.CTE=1 and DCSn.NDES=1 to launch DMA transfer.

For a channel with auto-request (DRTn.RT=0x8), the transfer begins automatically when the DCSn.CTE bit and DMAC.DMAE bit are set to 1. While for a channel with other request types, the transfer does not start until a transfer request is issued and detected.

For any channel n, The DTCn value is decremented by 1 for each successful transaction of a data unit. When the specified number of transfer data unit has been completed (DTCn = 0), the transfer ends normally. Meanwhile corresponding bit of DIRQP is set to 1. If DCMn.TIE bit is set to 1, an interrupt request is sent to the CPU. However, during the transfer, if a DMA address error occurs, the transfer is suspended, both DCSn.AR and DMAC.AR are set to 1 as well as corresponding bit of DIRQP. Then an interrupt request is sent to the CPU despite of DCMn.TIE.

Sometimes, for example, an UART parity error occurs for a channel that is transferring data between such UART and another terminal. In the case, both DCSn.HLT and DMAC.HLT are set to 1 and the transfer is suspended. Software should identify halt status by checking such two bits and re-configure DMA to let DMA rerun properly later.

9.4 DMA Requests

DMA transfer requests are normally generated from either the data transfer source or target, but also they can be issued by on-chip peripherals that are neither the source nor the target. There are two DMA transfer request types: auto-request, and on-chip peripheral request. For any channel n, its transfer request type is determined through DRTn.

9.4.1 Auto Request

When there is no explicit transfer request signal available, for example, memory-to-memory transfer or memory to some on-chip peripherals like GPIO, the auto-request mode allows the DMA to automatically generate a transfer request signal internally. Therefore, when DMA initialization done, once the DMAC.DMAE and DCSn.CTE are set to 1, the transfer begins immediately in channel n which DRTn=0x8.

9.4.2 On-Chip Peripheral Request

In the mode, transfer request signals come from on-chip peripherals. All request types except 0x8 (value of DRT) belong to the mode. **NOTE:** the transfer byte number for one request detection according to DCMn.RDIL must be equal or less than the byte number according to receive or transmit trigger value of source or target devices.

9.5 Channel Priorities

There are two dma cores, each one supports 6 channels dma transfer. The two cores have the same priority.

In each core, there are two sets: set 1 has the higher priority than set 2, within each set priority is round robin.

Table 9-5 Relationship among DMA Transfer connection, Request & Transfer Mode

Transfer Connection	Request Mode	Transfer Mode	Data Size (bits)	Channel
External memory or memory-mapped external device and on-chip peripheral module	Auto on-chip	Single	8/16/32 16-byte/32-byte/64-byte	0~5

9.6 Examples

9.6.1 Memory-to-memory auto request No-Descriptor Transfer

Suppose you want to do memory move between two different memory regions through channel 3, for example, moving 1KB data from address 0x20001000 to 0x20011000, do as following steps:

1 Check if (DMAC.AR==0 && DMAC.HLT==0 && DCS3.AR==0 && DCS3.HLT==0 && DCS3.NDES=1 && DTC3==0).
2 If above condition is true, set value 0 to DCS3.CTE to disable the channel 3 temporarily.
3 Set source address 0x20001000 to DSA3 and target address 0x20011000 to DTA3.
4 Suppose the data unit is word, set transfer count number 256 (1024/4) to DTC3.
5 Set auto-request (0x8) to DRT3.
6 Up to now, only the most important channel control register DCM3 is left, set it carefully:
   – Set value 1 to SAI and DAI*1.
   – Ignore RDIL because in the case there is no explicit request signal can be detected.
   – Set word size (0) to SP and DP*2.
   – Set value 1 to TIE to let CPU do some post process after the transfer done.
7 Set value 1 to DCS3.CTE and DMAC.DMAE to launch the transfer in channels 3.
8 When the transfer terminates normally (DTC3==0 && DCS3.TT==1), DIRQP.CIRQ3 will automatically be set value 1 and an interrupt request will be sent to CPU.
9 When CPU grants the interrupt request, in the corresponding IRQ handler, software must clear the DCS3.TT to value 0, and the behavior will automatically clear DIRQP.CIRQ3.

NOTES:

1 Either source or target is a FIFO, must not enable corresponding address increment.
2 When either source or target need be accessed through EMC (external memory controller), the real port with of the device is encapsulated by EMC, so you can set any favorite port with for it despite of the real one.

10 AHB Bus Arbiter

10.1 Overview

AHB bus arbiter is responsible for AHB bus transactions' arbitrating to provide a fair chance for each AHB master to possess the AHB bus. The refined arbiter adopts a new arbitrating technique to fulfill the back-to-back feature of AHB protocol. In detail, total 4 priority master groups are supported, a master belonging to higher priority group will be granted first. Moreover, in each group, round-robin strategy is used. Every AHB master can dynamically asserts different priority value (0 ~ 3) accompanying with each individual bus transaction to inform arbiter to arbitrate it in corresponsive master group, thus, more balanced bus workload may be capitalized.

There are three sets of AHB buses including AHB0 and AHB2, and they all instance this arbiter.

10.2 AHB Extension

To get better DDR performance, a little protocol extension is devised, additional AHB master owned signals are added for AHB protocol, they are:

emergency – *tell AHB arbiter to try to grant the current granted master's next time's BUSREQ when no other pending bus request with higher priority exists. AHB master can assert this hint signal when its continuous bus transactions locate in the same DDR bank and the same DDR row, or it is a real-time device thus interrupting its several continuous bus transactions may cause HW error. The signal must keep active when an AHB master wants to perform several bus transactions with above features back-to-back. Its 0->1 transition should be active accompanying with BUSREQ of first bus transaction, and at least at the last AP phase of the last time's bus transaction, it should be set to inactive 0.*

curr_len[6:0] – *data phase beats of a flexible fixed-length bus transaction. 0 denotes inactive; 2 ~ 64 is available. The signal must be active/inactive at the NONSEQ phase.*

next_len[6:0] – *data phase beats of next bus transaction with a flexible fixed-length. 0 denotes inactive; 2 ~ 64 is available. The signal must be active/inactive at the NONSEQ phase.*

offset[6:0] – *start address' offset between next bus transaction and current granted bus transaction. 0 ~ 127 is available. The signal must be active/inactive at the NONSEQ phase.*

10.3 Register Descriptions

The base addresses for memory-mapped registers of arbiter are listed below:

AHB0: 0x13000000

AHB2: 0x13400000

Table 10-1 AHB Bus Arbiter Registers List

Register Name	Offset	Size	R/W	Reset Value	Description
RPIOR	0x00	32	R/W	0x00000000	Default master priorities.
CTRL	0x04	32	R/W	0x00000000	AHB monitor control.
CLKL	0x08	32	R/W	undefined	AHB clock counter low.
EVENT0L	0x0C	32	R/W	undefined	AHB bus event 0 counter low.
EVENT1L	0x10	32	R/W	undefined	AHB bus event 1 counter low.
EVENTH	0x14	32	R/W	undefined	AHB bus event & clock counter high.
WATCHCTRL	0x18	32	RW	0x00000000	AHB bus watch control.
WATCHADDR	0x1C	32	RW	undefined	AHB bus watch address.
WATCHAMSK	0x20	32	RW	undefined	AHB bus watch address mask.
WATCHDATA	0x24	32	RW	undefined	AHB bus watch data.
WATCHDMSK	0x28	32	RW	undefined	AHB bus watch data mask.

10.3.1 Priority Order Register

HARB_PRIOR **offset 0**

Bit	31 30 29 28 27 26 25 24 23 22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved	PRIMX		PRIM9		PRIM8		PRIM7		PRIM6		PRIM5		PRIM4		PRIM3		PRIM2		PRIM1		PRIM0	
Rst	? ? ? ? ? ? ? ? ? ?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0

Bits	Name	Description	R/W
31:22	Reserved	Writing has no effect, read as zero.	R
21:0	PRI	PRIM0 ~ PRIMX. Default master's priority. Any AHB master can determine how to assert its priority. If an AHB master does not provide its priority aggressively, Its default static priority can be set to relative bit fields by SW. Value 0 ~ 3 are available, and 0 is the lowest priority.	RW

10.3.2 Monitor Control Register

HARB_MC offset 4

Bit	31:24	23:20	19:16	15:12	11:8	7:3	2	1	0
	Reserved	M1	M0	EV1	EV0	Reserved	EV1E	EV0E	CLKE
Rst	?	?	?	?	?	?	0	0	0

Bits	Name	Description	R/W
31:24	Reserved	Writing has no effect, read as zero.	R
23:20	M1	Monitored Master ID in monitor channel 1[*1].	RW
19:16	M0	Monitored Master ID in monitor channel 0[*1].	RW
15:12	EV1	AHB bus event encoding for monitor channel 1[*2].	RW
11:8	EV0	AHB bus event encoding for monitor channel 0[*2].	RW
7:3	Reserved	Write has no effect, read as zero.	R
2	EV1E	Enable monitor channel 1. 0: disable; 1: enable.	RW
1	EV0E	Enable monitor channel 0. 0: disable; 1: enable.	RW
0	CLKE	AHB clock counting enable. 0: disable; 1: enable.	RW

NOTES:

1 [*1] denotes the masterID encoding is described in the 68H Table 1-3 AHB0 Master-ID.

2 [*2] the event encoding is described in the 2569H Table 1-2 AHB Bus Monitor Events.

Table 10-2 AHB Bus Monitor Events

Events	Full Name	Comment
0	bus transaction cycles	exclude idle cycles.
1	bus transaction times	count NONSEQ times.
2	grant latency[*3]	count pending request (not granted) cycles.
3	critical grant latency trigger[*4]	Once the grant latency for a bus transaction exceeds the critical value preset in the counter low register, the associative counter high register will accumulate 1.
4	single beat transaction times	BURST type is SINGLE.
5	fixed length burst transaction times	BURST type is INCR4/8/16/32 or WRAP4/8/16/32.
6	INCR burst transaction times	BURST type is INCR.
7	critical transaction cycles trigger[*5]	Once the active transaction cycles for a bus transaction exceeds the critical value preset in the counter low register, the associative counter high register will accumulate 1.
8~15	reserved	

NOTE: *3, *4, *5 denotes that such events are undefined when masterID is ALL.

Table 10-3 AHB0 Master-ID

Masters	Full Name
0	CIM
1	LCD
2	IPU
3	AXI-to-AHB BRIDGE
4	-
5	-
6	CORE0
7	-
8	-
9	AOSD
10	AHB2-to-AHB0 BRIDGE
11~14	-
15	ALL (events triggered by any master should be monitored)

Table 10-4 AHB2 Master-ID

Masters	Full Name
0	USB
1	-
2	UHC
3	BDMA
4	ETHC
5	-
6	CPU-to-AHB2 BRIDGE
10	DMA
7~9, 11~14	-
15	ALL (events triggered by any master should be monitored)

10.3.3 AHB Clock Counter Low Register

HARB_CLKL offset 8

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CLKL																															
RST	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?

Bits	Name	Description	R/W
31:0	CLKL	Record the low 32 bits of AHB clock counter.	RW

10.3.4 Event0 Low Register

HARB_EVENT0L offset 12

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	EVENT0L																															
RST	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?

Bits	Name	Description	R/W
31:0	EVENT0L	Record the low 32 bits of event 0 counter.	RW

10.3.5 Event1 Low Register

HARB_EVENT1L offset 16

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	EVENT1L																															
RST	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?

Bits	Name	Description	R/W
31:0	EVENT1L	Record the low 32 bits of event 1 counter.	RW

10.3.6 Event High Register

HARB_EVENTH offset 20

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CLKH																EVENT1H								EVENT0H							
RST	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?

Bits	Name	Description	R/W
31:16	CLKH	Record the high 16 bits of AHB clock counter.	RW
15:8	EVENT1H	Record the high 8 bits of event 1 counter.	RW
7:0	EVENT0H	Record the high 8 bits of event 0 counter.	RW

Note that fields of EVENTH register will not overflow automatically. For example, when EVENT1H reaches 0xFF during monitoring, it remains the value until software modifies it.

10.3.7 AHB Watch Control Register

WATCH_CTRL **offset 24**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	WF	RF	MasterID														Reserved							IRQE	Reserved					DE	WE	RE
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	?	?	?	?	?	?	?	0	?	?	?	?	?	0	0	0

Bits	Name	Description	R/W
31	WF	0: no write watch point detected; 1: a write watch point is detected.	RW
30	RF	0: no read watch point detected; 1: a read watch point is detected.	RW
29:16	MasterID	0: ID of the master just triggering watch point, one-hot encoding.	RW
15:9	Reserved	Writing has no effect, read as zero.	R
8	IRQE	Interrupt enable. 1: if WF or RF is set value 1, an interrupt request arises immediately. In this version, AHB0 interrupt is not available.	RW
7:3	Reserved	Writing has no effect, read as zero.	R
2	DE	Watch data enable. 1: enable.	RW
1	WE	Switch of triggering watch point by write. 1: enable.	RW
0	RE	Switch of triggering watch point by read. 1: enable.	RW

10.3.8 AHB Watch Address Register

WATCH_ADDR **offset 28**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ADDR																															
RST	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?

Bits	Name	Description	R/W
31:0	ADDR	Watch address to be monitored.	RW

10.3.9 AHB Watch Address Mask Register

WATCH_AMSK offset 32

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MASK																															
RST	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?

Bits	Name	Description	R/W
31:0	MASK	If a bit is set 0, which means in WATCH_ADDR, corresponding bit position should be monitored, otherwise, the bit position should be ignored.	RW

10.3.10 AHB Watch Data Register

WATCH_DATA offset 36

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DATA																															
RST	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?

Bits	Name	Description	R/W
31:0	DATA	Read or Write data to be monitored.	RW

10.3.11 AHB Watch Data Mask Register

WATCH_DMSK offset 40

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MASK																															
RST	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?

Bits	Name	Description	R/W
31:0	MASK	If a bit is set 0, which means in WATCH_DATA, corresponding bit position should be monitored, otherwise, the bit position should be ignored.	RW

11 Clock Reset and Power Controller

11.1 Overview

The Clock & Power management block consists of three parts: Clock control, PLL control, and Power control, Reset control.

The Clock control logic can generate the required clock signals including CCLK for CPU, HCLK for the AHB0 and DDR，AUX_CCLK and H1CLK for VPU , H2CLK for the AHB2 bus peripherals, PCLK for the APB bus peripherals. The Chip has two Phase Locked Loops (PLL): for CCLK, AUX_CCLK, H0CLK, H1CLK, H2CLK and PCLK, GPUCLK, SSICLK, MSCLK, LPCLK, USBCLK, I2SCLK. The clock control logic can make slow clocks without PLL and connect/disconnect the clock to each peripheral block by software, which will reduce the power consumption.

For the power control logic, there are various power management schemes to keep optimal power consumption for a given task. The power management block can activate four modes: NORMAL mode, DOZE mode, IDLE mode, SLEEP mode.

Support power supply shut down for 3 power domain separately. Software may separately shut down AHB1 module. When in Sleep mode, software may shut down J1. Thus, the chip may best reduce leakage current.

For reset control logic, the hardware reset and hibernate reset is extended to more 40ms. It controls or distributes all of the system reset signals.

11.2 Clock Generation UNIT

The clock generation unit (CGU) contains one PLL driven by an external oscillator and the clock generation circuit from which the following clocks are derived:

Signal	Description
CCLK	Fast clock for internal operations such as executing instructions from the cache. It can be gated during doze and idle mode when all the criteria to enter a low power are met.
AUX_CCLK	AUX_CPU Clock.
H0CLK	AHB0 and DDR Bus Clock.
H1CLK	VPU Clock.
H2CLK	AHB2 Speed Bus Clock.
PCLK	APB Speed Bus Clock.
CKO	DDR or SDRAM Clock.
LPCLK	LCD pixel clock.
TVECLK	TV encoder 27M clock.
CIM_MCLK	Clock output from CIM module.
CIM_PCLK	Clock input to CIM module.
GPUCLK	GPU clock.
I2SCLK	I2S codec clock.
BITCLK	AC97 bit clock.
PCMCLK	PCM clock.
MSCCLK	MSC clock.
SSICLK	SSI clock.
TSSICLK	TSSI clock.
EXCLK	12M clock output for UART I2C TCU USB2.0-PHY AUDIO CODEC.

Feature:

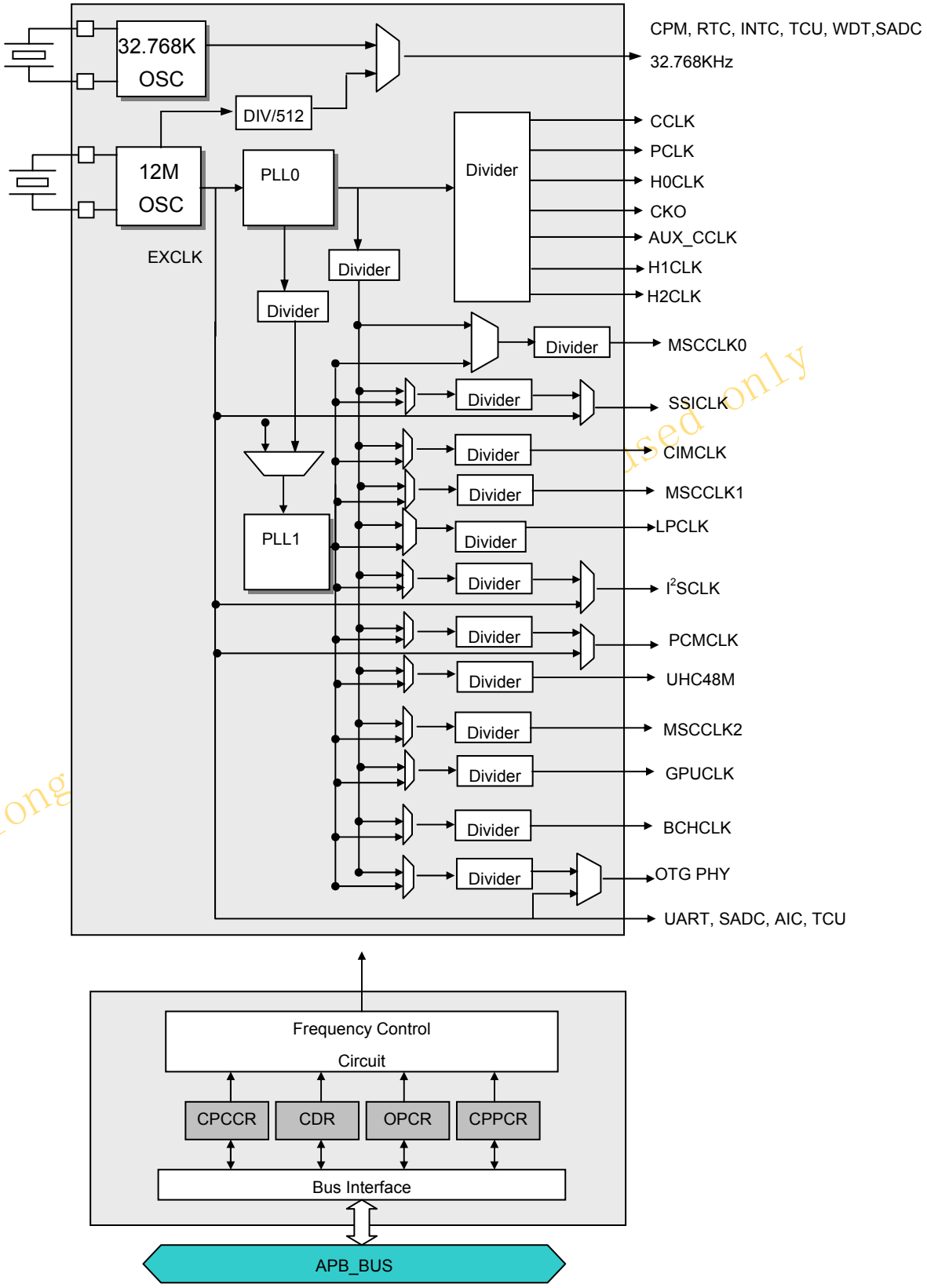
- On-chip 2MHz~27MHZ oscillator circuit
- On-chip 32.768KHZ oscillator circuit
- One two-chip phase-locked loops (PLL) with programmable multiplier
- CCLK, H0CLK, H1CLK, PCLK, H2CLK, CKO and LPCLK, GPUCLK, MSCCLK , UHCCLK, SSICLK frequency can be changed separately for software by setting registers
- SSI clock supports 50M clock
- MSC clock supports 50M clock
- Functional-unit clock gating
- Shut down power supply for J1, VPU

11.2.1 Pin Description

Name	I/O	Description
RTCLK_XI	Input	32.768KHZ Oscillator input signal
RTCLK_XO	Output	32.768KHZ Oscillator output signal
EXCLK	Input	Oscillator input signal
EXCLKO	Output	Oscillator output signal
CIM_MCLK	Output	Clock output from CIM module signal
CIM_PCLK	Input	Clock input to CIM module signal
LPCLK	Output	LCD pix clock signal
CKO	Output	DDR clock signal
TSSICLK	Input	TSSI clock signal
BITCLK	Inout	I2S/AC97 bit clock
PCMCLK	Inout	PCM bit clock
MSC_CLK	Output	Clock output For MMC/SD Card signal
SSI_CLK	Output	Clock output from SSI module signal

11.2.2 CGU Block Diagram

Following figure illustrates a block diagram of CGU.



11.2.3 Clock Overview

There is an internal PLL in this chip. PLL input clock is an external input clock EXCLK. Theoretically, EXCLK can be 2MHz ~ 27MHz.

CCLK is CPU clock. It is usually the fastest clock in the chip. This clock represents the chip speed.

AUX_CCLK is for AUX_CPU clock, it is should equal to half of CCLK or 1/3 of CCLK.

H0CLK is for on chip high speed peripherals connected to AHB0 bus.

H1CLK is for on chip high speed peripherals connected to VPU bus.

H2CLK is for on chip high speed peripherals connected to AHB2 bus.

PCLK is for on chip slow speed peripherals connected to APB bus.

CCLK, AUX_CCLK, H0CLK, H1CLK, H2CLK, PCLK are synchronous clocks that may have different frequencies. They are from the same clock source, the on chip PLL output clock in most cases. H0CLK , H1CLK, H2CLK frequency can be equal to CCLK/2 or divided CCLK by an integer. PCLK frequency can be equal to H2CLK/2 or divided CCLK by an integer.

AC97 in AIC module needs a 12.288MHz BIT clock. It is input from the external AC97 CODEC chip or other clock source. I2S and PCM clock are generated from PLL output clock.

Besides PLL input, EXCLK also provides device clock or one of device clocks for many peripherals, such as, UART, TCU, SSI, SADC and WDT.

Device clock of MSC (MMC/SD) is taken from software divided PLL0/PLL1 output clock.

USB device and host controllers need a 48MHz USB clock. USB clock can be selected by software divided PLL0/PLL1 output clock.

Device clock of SSI is taken from software divided PLL output clock. or 12M from oscillator.

LCD's pixel clock is generated from PLL output clock, which are divided by one independent dividers.

GPU clock is generated from PLL output clock, which is divided by one independent dividers.

BCH clock is generated from PLL output clock, which is divided by one independent dividers.

The slowest clock is RTCLK, which is usually 12M/512 or 32768Hz.

11.2.4 CGU Registers

All CGU register 32bit access address is physical address.

Table 11-1 CGU Registers Configuration

Name	description	RW	Reset Value	Address	Access Size
CPCCR	Clock Control Register	RW	0x01010110	0x10000000	32
CPPCR	PLL Control Register0	RW	0x????0020	0x10000010	32
CPPSR	PLL switch and status register	RW	0x80000000	0x10000014	32
CPPCR1	PLL Control Register1	RW	0x????0002	0x10000030	32
CPSPR	CPM Scratch Pad Register	RW	0x????????	0x10000034	32
CPSPPR	CPM Scratch Protected Register	RW	0x0000a5a5	0x10000038	32
USBPCR	USB Parameter control register	RW	0x429919b8	0x1000003C	32
USBRDT	USB Reset Detect Timer Register	RW	0x02000096	0x10000040	32
USBVBFIL	USB jitter filter Register	RW	0x00ff0080	0x10000044	32
USBPCR1	USB Parameter control register 1	RW	0x00000004	0x10000048	32
USBCDR	OTG PHY clock divider Register	RW	0x00000000	0x10000050	32
I2SCDR	I2S device clock divider Register	RW	0x00000000	0x10000060	32
LPCDR	LCD pix clock divider Register	RW	0x00000000	0x10000064	32
MSC0CDR	MSC0 clock divider Register	RW	0x00000000	0x10000068	32
UHCCDR	UHC 48M clock divider Register	RW	0x00000000	0x1000006C	32
SSICDR	SSI clock divider Register	RW	0x00000000	0x10000074	32
CIMCDR	CIM MCLK clock divider Register	RW	0x00000000	0x1000007C	32
PCMCDR	PCM device clock divider Register	RW	0x00000000	0x10000084	32
GPUCDR	GPU clock divider Register	RW	0x00000000	0x10000088	32
MSC1CDR	MSC1 clock divider Register	RW	0x80000003	0x100000A4	32
MSC2CDR	MSC2 clock divider Register	RW	0x80000003	0x100000A8	32
BCHCDR	BCH clock divider Register	RW	0x00000000	0x100000AC	32
CPM_INTR	CPM interrupt Register	RW	0x00000000	0x100000B0	32
CPM_INTRE	CPM interrupt Enable Register	RW	0x00000000	0x100000B4	32

11.2.4.1 Clock Control Register

The Clock Control Register (CPCCR) is a 32-bit read/write register, which controls CCLK, AUX_CCLK, H0CLK, H1CLK, H2CLK and PCLK division ratios. It is initialized to 0x01010110 by any reset. Only word access can be used on CPCCR.

CPCCR **0x10000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved	MEM	Reserved		H1DIV				Reserved	CE	PCS	Reserved	H2DIV				C1DIV				PDIV				H0DIV				CDIV			
RST	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0

Bits	Name	Description	RW
31	Reserved	Writing has no effect, read as zero.	R
30	MEM	0: mobile ddr 1: ddr or ddr2 memory	RW
29:28	Reserved	Writing has no effect, read as zero.	R
27:24	H1DIV	Divider for VPU Bus Clock Frequency. Specified the H1CLK division ratio.	RW
22	CE	change enable. If CE is 1, writes on CDIV, C1DIV, H2DIV, H0DIV, PDIV, H1DIV, USBCDR, LPCDR, CIMCDR, MSCCDR, SSICDR, BCHCDR UHCCDR, GPUCDR , PCMCDR , I2SCDR will start a frequency changing sequence immediately. When CE is 0, writes on CDIV, C1DIV, H2DIV, HDIV, PDIV, H0DIV, USBCDR, LPCDR, CIMCDR, MSCCDR, SSICDR, UHCDR, PCMCDR, GPUCDR, I2SCDR will not start a frequency changing sequence immediately. The division ratio is actually updated in PLL multiple ratio changing sequence or PLL Disable Sequence. 0: Division ratios are updated in PLL multiple ratio changing sequence or PLL Disable Sequence 1: Division ratios are updated immediately	RW
21	PCS	PLL out clock source clock selection. It supplies source clock for MSC I2S LCD UHC OTG SSI PCM GPU GPS BCH. 0: divider clock source is PLL output 1: divider clock source is PLL output divided by 2 Software should set the bit according to various needs.	RW
20	Reserved	Writing has no effect, read as zero.	R
19:16	H2DIV	Divider for AHB2 Clock Frequency. Specified the H2CLK division ratio.	RW

Bit 27~24: H1DIV				Description
0	0	0	0	X1
0	0	0	1	X1/2
0	0	1	0	X1/3
0	0	1	1	X1/4
0	1	0	0	X1/6
0	1	0	1	X1/8
0	1	1	0	X1/12
Other Value				Reserved

			Bit 19~16: H2DIV	Description
			0 0 0 0	X1
			0 0 0 1	X1/2
			0 0 1 0	X1/3
			0 0 1 1	X1/4
			0 1 0 0	X1/6
			0 1 0 1	X1/8
			0 1 1 0	X1/12
			Other Value	Reserved
15:12	C1DIV	Divider for AUX CPU Frequency. Specified the AUX_CLK division ratio.		RW
			Bit 15~12: C1DIV	Description
			0 0 0 0	X1
			0 0 0 1	X1/2
			0 0 1 0	X1/3
			0 0 1 1	X1/4
			0 1 0 0	X1/6
			0 1 0 1	X1/8
			0 1 1 0	X1/12
			Other Value	Reserved
11:8	PDIV	Divider for Peripheral Clock Frequency. Specified the PCLK division ratio.		RW
			Bit 11~8: PDIV	Description
			0 0 0 0	X1
			0 0 0 1	X1/2
			0 0 1 0	X1/3
			0 0 1 1	X1/4
			0 1 0 0	X1/6
			0 1 0 1	X1/8
			0 1 1 0	X1/12
			Other Value	Reserved
7:4	H0DIV	Divider for AHB0 Clock Frequency. Specified the H0CLK division ratio.		RW
			Bit 7~4: H0DIV	Description
			0 0 0 0	X1
			0 0 0 1	X1/2
			0 0 1 0	X1/3
			0 0 1 1	X1/4
			0 1 0 0	X1/6
			0 1 0 1	X1/8
			0 1 1 0	X1/12
			Other Value	Reserved
3:0	CDIV	Divider for CPU Clock Frequency. Specifies the CCLK division ratio.		RW
			Bit 3~0: CDIV	Description

			0	0	0	0	X1		
			0	0	0	1	X1/2		
			0	0	1	0	X1/3		
			0	0	1	1	X1/4		
			0	1	0	0	X1/6		
			0	1	0	1	X1/8		
			0	1	1	0	X1/12		
			Other Value				Reserved		

11.2.4.2 I2S device clock divider Register

I2S device clock divider Register (I2SCDR) is a 32-bit read/write register that specifies the divider of I2S device clock . This register is initialized to 0x00000000 only by any reset. Only word access can be used on I2SCDR.

I2SCDR **0x10000060**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	I2CS	I2PCS	Reserved																					I2SCDR								
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
31	I2CS	I2S Clock Source Selection. Selects the I2S clock source between PLL output and pin EXCLK. 0: I2S clock source is EXCLK 1: I2S clock source is PLL output divided by I2SDIV If EXCLK is 12M, please don't change the bit.	R
30	I2PCS	0: select PLL0 clock output 1: select PLL1 clock output	
29:9	Reserved	Writing has no effect, read as zero.	R
8:0	I2SCDR	Divider for I2S Frequency. Specified the I2S device clock division ratio, which varies from 1 to 512 (division ratio = I2SCDR + 1). When EXCLK is 12M, don't care the bit.	RW

11.2.4.3 USB clock divider Register

USB clock divider Register (USBCDR) is a 32-bit read/write register that specifies the divider of OTG PHY clock . This register is initialized to 0x00000000 only by any reset. Only word access can be used on USBCDR.

USBCDR 0x10000050

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	UCS	UPCS	Reserved																						USBCDR							
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
31	UCS	OTG PHY Clock Source Selection. Selects the OTG PHY clock source between PLL output and pin EXCLK. 0: OTG clock source is pin EXCLK 1: OTG clock source is PLL output If EXCLK is 12M, please don't change the bit.	RW
30	UPCS	0: select PLL0 clock output 1: select PLL1 clock output	
29:9	Reserved	Writing has no effect, read as zero.	R
7:0	USBCDR	Divider for OTG PHY Clock Frequency. When OTG PHY clock source is PLL (UCS bit is 1), this field specified the OTG PHY clock division ratio, which varies from 1 to 256 (division ratio = USBCDR + 1).	RW

11.2.4.4 LCD pix clock divider Register

LCD pix clock divider Register (LPCDR) is a 32-bit read/write register that specifies the divider of LCD pixel clock (LPCLK). This register is initialized to 0x00000000 only by any reset. Only word access can be used on LPCDR.

LPCDR 0x10000064

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved	LTCS	LPCS	Reserved																					LPCDR							
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
31	Reserved	Writing has no effect, read as zero.	R
30	LTCS	LCD TV Encoder or Panel pix clock Selection. 0: pix clock is used as LCD PANEL 1: pix clock is used as TV ENCODER	RW
29	LPCS	0: select PLL0 clock output 1: select PLL1 clock output	RW
29:11	Reserved	Writing has no effect, read as zero.	R
10:0	LPCDR	Divider for Pixel Frequency. Specified the LCD pixel clock (LPCLK) division ratio, which varies from 1 to 2048 (division ratio = LPCDR + 1).	RW

11.2.4.5 MSC0 device clock divider Register

MSC0 device clock divider Register (MSC0CDR) is a 32-bit read/write register that specifies the divider of MSC device clock . This register is initialized to 0x00000000 only by any reset. Only word access can be used on MSC0CDR.

MSCCDR **0x10000068**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MCSG	MPCS	Reserved																							MSCCDR							
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
31	MCSG	MSC Clock Source Gate. Whether gate MSC clock divider. 0: clock divider is on, clock is running 1:clock divider is off , clock is stopped	RW
30	MPCS	0: select PLL0 clock output 1: select PLL1clock output	RW
29:7	Reserved	Writing has no effect, read as zero.	R
6:0	MSCCDR	Divider for MSC Frequency. Specified the MSC device clock division ratio, which varies from 1 to 128 (division ratio = MSCCDR + 1).	RW

11.2.4.6 MSC1 device clock divider Register

MSC1 device clock divider Register (MSC1CDR) is a 32-bit read/write register that specifies the divider of MSC device clock . This register is initialized to 0x80000003 only by any reset. Only word access can be used on MSC1CDR.

MSCCDR **0x100000A4**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MCSG	MPCS	Reserved																							MSCCDR							
RST	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

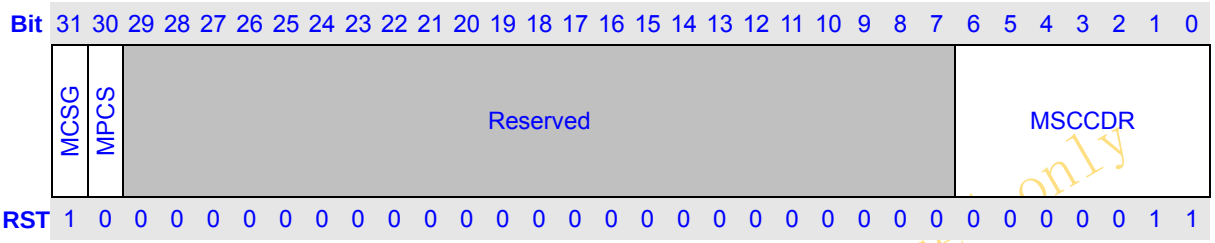
Bits	Name	Description	RW
31	MCSG	MSC Clock Source Gate. Whether gate MSC clock divider. 0: clock divider is on, clock is running 1: clock divider is off , clock is stopped	RW
30	MPCS	0: select PLL0 clock output 1: select PLL1clock output	RW

29:7	Reserved	Writing has no effect, read as zero.	R
6:0	MSCCDR	Divider for MSC Frequency. Specified the MSC device clock division ratio, which varies from 1 to 128 (division ratio = MSCCDR + 1).	RW

11.2.4.7 MSC2 device clock divider Register

MSC2 device clock divider Register (MSC2CDR) is a 32-bit read/write register that specifies the divider of MSC device clock . This register is initialized to 0x80000003 only by any reset. Only word access can be used on MSC2CDR.

MSCCDR **0x100000A8**

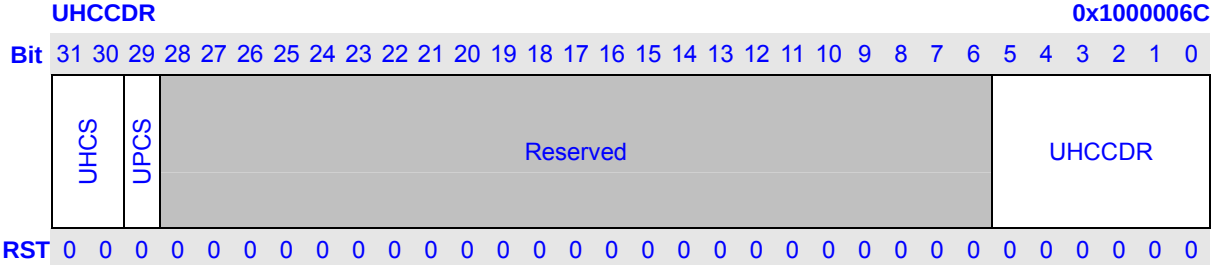


Bits	Name	Description	RW
31	MCSG	MSC Clock Source Gate. Whether gate MSC clock divider. 0: clock divider is on, clock is running 1:clock divider is off , clock is stopped	RW
30	MPCS	0: select PLL0 clock output 1: select PLL1clock output	RW
29:7	Reserved	Writing has no effect, read as zero.	R
6:0	MSCCDR	Divider for MSC Frequency. Specified the MSC device clock division ratio, which varies from 1 to 128 (division ratio = MSCCDR + 1).	RW

11.2.4.8 UHC device clock divider Register

UHC device clock divider Register (UHCCDR) is a 32-bit read/write register that specifies the divider of UHC 48M device clock . This register is initialized to 0x00000000 only by any reset. Only word access can be used on UHCCDR.

UHCCDR **0x1000006C**



Bits	Name	Description	RW
31:30	UHCS	00: UHC clock source is PLL divider output 01: UHC clock source is PLL divider output 10: UHC clock source is OTG_PHY 11: UHC clock source is external PIN	RW
29	UHPCS	0: select PLL0 clock output 1: select PLL1 clock output	RW
30:6	Reserved	Writing has no effect, read as zero.	R
3:0	UHCCDR	Divider for UHC Frequency. Specified the UHC 48M device clock division ratio, which varies from 1 to 64 (division ratio = UHCCDR + 1).	RW

11.2.4.9 SSI device clock divider Register

SSI device clock divider Register (SSICDR) is a 32-bit read/write register that specifies the divider of SSI device clock . This register is initialized to 0x00000000 only by any reset. Only word access can be used on SSICDR.

SSICDR 0x10000074

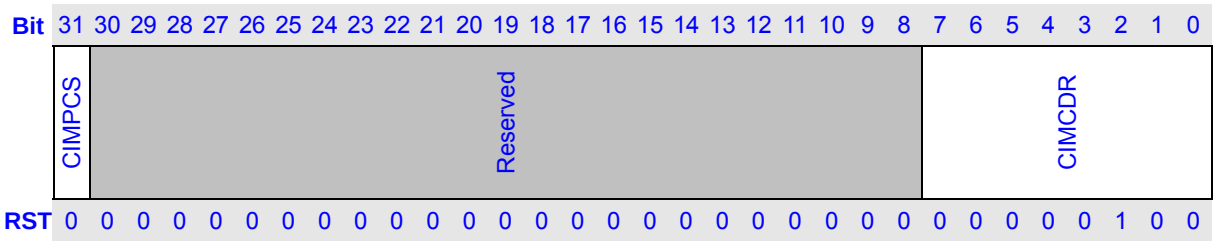
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SCS	SPCS	Reserved																							SSICDR							
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
31	SCS	SSI Clock Source Selection. Selects the SSI clock source between PLL output and pin EXCLK. 0: SSI clock source is EXCLK 1: SSI clock source is PLL output divided by SSICDR	R
30	SPCS	0: select PLL0 clock output 1: select PLL1 clock output	RW
29:7	Reserved	Writing has no effect, read as zero.	R
6:0	SSICDR	Divider for SSI Frequency. Specified the SSI device clock division ratio, which varies from 1 to 128 (division ratio = SSICDR + 1).	RW

11.2.4.10 CIM MCLK clock divider Register

CIM mclk clock divider Register (CIMCDR) is a 32-bit read/write register that specifies the divider of CIM mclk clock (CIM_MCLK). This register is initialized to 0x00000000 only by any reset. Only word access can be used on CIMCDR.

CIMCDR 0x1000007C

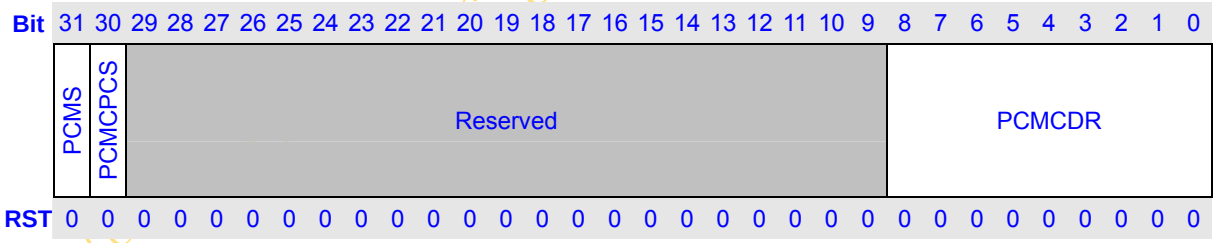


Bits	Name	Description	RW
31	CIMPCS	0: select PLL0 clock output 1: select PLL1 clock output	RW
30:8	Reserved	Writing has no effect, read as zero.	R
7:0	CIMCDR	Divider for CIM MCLK Frequency. Specified the CIM MCLK clock (CIM_MCLK) division ratio, which varies from 1 to 256 (division ratio = CIMCDR + 1).	RW

11.2.4.11 PCM device clock divider Register

PCM device clock divider Register (PCMCDR) is a 32-bit read/write register that specifies the divider of PCM device clock . This register is initialized to 0x00000000 only by any reset. Only word access can be used on PCMCDR.

PCMCDR 0x10000084



Bits	Name	Description	RW
31	PCMS	PCM source clock Selection. 0: PCM source clock is pin EXCLK 1: PCM source clock is PLL divider output	RW
30	PCMPCS	0: select PLL0 divider output 1: select PLL1 divider output	
30:9	Reserved	Writing has no effect, read as zero.	R
8:0	PCMCDR	Divider for PCM Frequency. Specified the PCM device clock division ratio, which varies from 1 to 512 (division ratio = PCMCDR + 1).	RW

11.2.4.12 GPU clock divider Register

GPU clock divider Register (GPUCDR) is a 32-bit read/write register that specifies the divider of GPU clock . This register is initialized to 0x00000000 only by any reset. Only word access can be used on GPUCDR.

GPUCDR **0x10000088**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GPCS	Reserved																										GPUCDR				
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
31	GPCS	0: select PLL0 divider output 1: select PLL1 divider output	RW
30:3	Reserved	Writing has no effect, read as zero.	R
4:0	GPUCDR	Divider for GPU Frequency. Specified the GPU clock division ratio, which varies from 1 to 32 (division ratio = GPUCDR + 1).	RW

11.2.4.13 BCH clock divider Register

BCH clock divider Register (BCHCDR) is a 32-bit read/write register that specifies the divider of BCH clock . This register is initialized to 0x00000000 only by any reset. Only word access can be used on BCHCDR.

GPUCDR **0x100000AC**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BPCS	BCHM	Reserved																												BCHCDR	
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
31	BPCS	0: select PLL0 divider output 1: select PLL1 divider output	RW
30	BCHM	0: hardware frequency change 1: software frequency change	RW
30:3	Reserved	Writing has no effect, read as zero.	R
1:0	BCHCDR	Divider for BCH Frequency. Specified the BCH clock division ratio, which varies from 1 to 4 (division ratio = BCHCDR + 1).	RW

11.2.4.14 CPM interrupt Register

CPM_INTR 0x100000B0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																														VBUS_INTR	ADEV_INTR
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
31:3	Reserved	Writing has no effect, read as zero.	R
1	VBUS_INTR	B device insert interrupt.	R
0	ADEV_INTR	A device insert interrupt.	R

11.2.4.15 CPM interrupt enable Register

CPM_INTRE 0x100000B4

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																														VBUS_INTRE	ADEV_INTRE
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
31:3	Reserved	Writing has no effect, read as zero.	R
1	VBUS_INTRE	B device insert interrupt enable.	RW
0	ADEV_INTRE	A device insert interrupt enable.	RW

11.2.4.16 CPM Scratch Pad Protected Register

The Scratch Pad Protected Register is reset to 0x0000a5a5. When CPSPPR value equals to 0x00005a5a, software can write the CPSPR.

CPSPPR 0x10000038

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																CPSPPR															
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	1	0	1	1	0	1	0	0	1	0	1

Bits	Name	Description	RW
31:16	Reserved	Writing has no effect, read as zero.	R
15:0	CPSPPR	The value is only = 0x00005a5a, software can write the CPSPR.	RW

11.2.4.17 CPM Scratch Pad Register

The Scratch Pad Register is a 32-bit read/write register that allows software to preserve some critical data . It is not initialized by poweron and WDT reset.

CPPSR 0x10000034

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CPPSR																															
RST	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?

11.2.4.18 USB Parameter Control Register

The USBPCR is a 32-bit read/write register that allows software to control OTG PHY some functions. It is initialized to 0x429919b8.

USBPCR 0x1000003C

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	USB_MODE	AVLD_REG	IDPULLUP_MASK		INCR_MASK	CLK12_EN	COMONONN	VBUSVLDEXT	VBUSVLDEXTSEL	POR	SIDDQ	OTGDISABLE	COMPDISTUNE			OTGTUNE		SQRXTUNE		TXFSLSTUNE		TXPREMTUNE	TXHSXVTUNE		TXVREFTUNE						
RST	0	1	0	0	0	0	1	0	1	0	0	1	1	0	0	1	0	0	0	1	1	0	0	1	1	0	1	1	1	0	0	0

Bits	Name	Description	RW
31	USB_MODE	0: work as USB device; 1: work as OTG.	RW
30	AVLD_REG	This bit is used to set “avalid”(VBUS above A-device session threshold) signal.	RW
29:28	IDPULLUP_MASK	These 2 bits control “idpullup” signal in otg mode. 2’b1x: “idpullup” always active 2’b01: “idpullup” always active when usb suspend	RW

		2'b00: use "idpullup" from otg controller	
27	INCR_MASK	This bit controls whether the ahb interface enhancement for "incr transfer" takes effect. Set this bit to 0 will active the enhancement.	RW
26	CLK12_EN	OTG PHY reference clock enable.	RW
25	COMMONONN	This bit is the OTG PHY common block power down control signal. 0: The common blocks remain powered in suspend mode 1: The common blocks are powered down in suspend mode	RW
24	VBUSVLDEXT	This bit controls OTG PHY VBUSVLDEXT signal.	RW
23	VBUSVLDEXTSEL	This bit controls OTG PHY VBUSVLDEXTSEL signal.	RW
22	POR	This bit controls OTG PHY power on reset.	RW
21	SIDDQ	This bit is the OTG PHY analog blocks power down signal.	RW
20	OTG_DISABLE	This bit is the power control for otg block in OTG PHY.	RW
19:17	COMPDISTUNE	These bits control disconnect threshold adjustment.	RW
		3'b111 +4.5%	
		3'b110 +3%	
		3'b101 +1.5%	
		3'b100 Default	
		3'b011 -1.5%	
		3'b010 -3%	
		3'b001 -4.5%	
		3'b000 -6%	
16:14	OTGTUNE	These bits control VBUS valid threshold adjustment.	RW
		3'b111 +4.5%	
		3'b110 +3%	
		3'b101 +1.5%	
		3'b100 Default	
		3'b011 -1.5%	
		3'b010 -3%	
		3'b001 -4.5%	
		3'b000 -6%	
13:11	SQRXTUNE	These bits control squelch threshold adjustment.	RW
		3'b111 -20%	
		3'b110 -15%	
		3'b101 -10%	
		3'b100 -5%	
		3'b011 default	
		3'b010 +5%	
		3'b001 +10%	
		3'b000 +15%	
10:7	TXFSLSTUNE	These bits control FS/LS source impedance adjustment.	RW

		4'b1111	-5%	
		4'b0111	-2.5%	
		4'b0011	Default	
		4'b0001	+2.5%	
		4'b0000	+5%	
6	TXPREEMPHTUNE	This bit controls HS transmitter Pre-emphasis enable. 0: disable; 1: enable.		RW
5:4	TXHSXVTUNE	These bits adjust the voltage at which dp and dm signals cross while transmitting in HS mode.		RW
		2'b11	Default	
		2'b10	+15mv	
		2'b01	-15mv	
		2'b00	reserved	
3:0	TXVREFTUNE	These bits control HS DC voltage level adjustment.		RW
		4'b1111	+12.5%	
		4'b1110	+11.25%	
		4'b1101	+10%	
		4'b1100	+8.75%	
		4'b1011	+7.5%	
		4'b1010	+6.255	
		4'b1001	+5%	
		4'b1000	+3.75%	
		4'b0111	+2.5%	
		4'b0110	+1.25%	
		4'b0101	Default	
		4'b0100	-1.25%	
		4'b0011	-2.5%	
		4'b0010	-3.75%	
		4'b0001	-5%	
		4'b0000	-6.25%	

11.2.4.19 USB Reset Detect Timer Register

USBRDT **0x10000040**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved					HB_MASK	VBFIL_LD_EN	IDDIG_EN	IDDIG_REG	USBRDT																						
RST	?	?	?	?	?	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	1	1	0

Bits	Name	Description	RW
31:27	Reserved	Writing has no effect, read as zero.	R
26	HB_MASK	Halfword/Byte transfer support mask. 0: enable 1: mask	RW
25	VBFIL_LD_EN	VBUS filter data load enable.	RW
24	IDDIG_EN	This bit indicates using IDDIG_REG to control “iddig” signal.	RW
23	IDDIG_REG	This bit controls “iddig” when IDDIG_REG_EN = 1’b1.	RW
22:0	USBRDT	These bits control USB reset detect time.	RW

11.2.4.20 USB VBUS Jitter Filter Register

USBVBFIL 0x10000044

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	IDDIGFIL																USBVBFIL															
RST	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0

Bits	Name	Description	RW
31:16	IDDIGFIL	These bits controls iddig jitter filter time.	RW
15:0	USBVBFIL	These bits controls VBUS jitter filter time.	RW

11.2.4.21 USB Parameter Control Register1

The USBPCR1 is a 32-bit read/write register that allows software to control OTG PHY some functions. It is initialized to 0x00000004.

USBPCR1 0x10000048

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																									TXRISETUNE	UHC_pdbar	UHC_ibspot			UHC_xp	
RST	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	1	0	0

Bits	Name	Description	RW
6	TXRISETUNE	These bit adjust the rise/fall times of the HS waveform. 1’b0 — default 1’b1 — -8%	RW
5	UHC_pdbar	Power down Mode. Enables power down state. 0: power down	RW

		1: power on	
4:2	UHC_ibspot	Current option. 3'b000 — 6u 3'b001 — 5u (default) 3'b010 — 4u	RW
1:0	UHC_xp	Cross-point control of DP, DM. 2'b0x — default cross-point: VDD/2 2'b11 — cross-point up: VDD/2 + 400mV 2'b10 — cross-point down: VDD/2 - 400mV	RW

11.2.4.22 PLL Control Register0

The PLL Control Register (CPPCR) is a 32-bit read/write register, which controls PLL multiplier, on/off state and stabilize time. It is initialized to 0x????0020 only by any reset. Only word access can be used on CPPCR.

CPPCR **0x10000010**

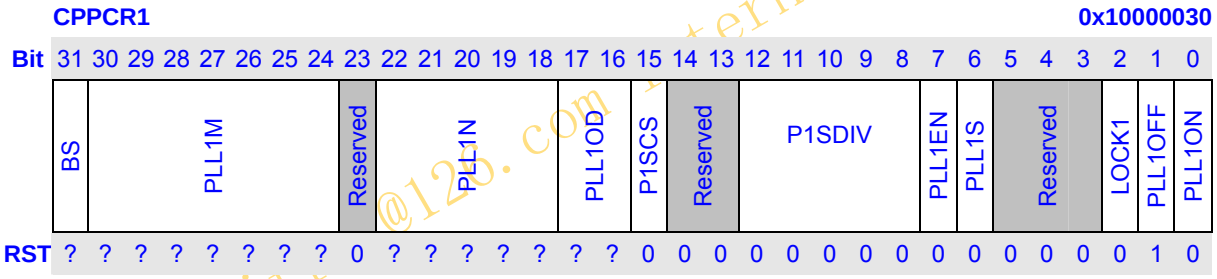
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BS				PLLM				Reserved			PLLN			PLLOD		LOCK0	Reserved				PLLS	PLLBP	PLLEN	PLLST							
RST	?	?	?	?	?	?	?	?	0	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0

Bits	Name	Description	RW
31	BS	0: low band 1: high band	RW
30:24	PLLM	the PLL feedback 7-bit divider.	RW
23	Reserved	Writing has no effect, read as zero.	R
22:18	PLLN	the PLL input 5-bit divider.	RW
17:16	PLLOD	00: divide by 1 01: divide by 2 10: divide by 4 11: divide by 8	RW
15	LOCK0	0: the PLL output is stable 1: the PLL output is not stable Software should clear this bit to 0, when this bit equal to 1, it indicates that PLL hadn't stable previously , it is only used to debug.	RW
14:11	Reserved	Writing has no effect, read as zero.	R
10	PLLS	PLL Stabilize Flag. 0: PLL is off or not stable 1: PLL is on and stable	R

9	PLLBP	PLL Bypass. If PLLEN is 1, set this bit to1 will bypass PLL. The PLL is still running background but the source of associated dividers is switched to 12-M. If PLLEN is 0, set this bit to 1 has no effect. If PLLEN is 1, clear this bit to 0 will switch the source of associated dividers to PLL output.	RW
8	PLLEN	PLL Enable. When PLLEN is set to 1, PLL starts to lock phase. After PLL stabilizes, PLLS bit is set. If PLLBP is 0, the source of associated dividers, is switched to PLL output. When PLLEN is clear to 0, PLL is shut off and the source of associated dividers is switched to 12-MHz in spite of PLLBP bit.	RW
7:0	PLLST	PLL Stabilize Time. Specifies the PLL stabilize time by unit of RTCCLK (approximate 32kHz) cycles. It is used when change PLL multiplier or change PLL from off to on. It is initialized to H'20.	RW

11.2.4.23 PLL Control Register1

The PLL Control Register (CPPCR1) is a 32-bit read/write register, which controls PLL multiplier, on/off state and stabilize time. It is initialized to 0x????0002 only by any reset. Only word access can be used on CPPCR1.



Bits	Name	Description	RW
31	BS	0: low band 1: low band	RW
30:24	PLL1M	the PLL1 feedback 7-bit divider.	RW
23	Reserved	Writing has no effect, read as zero.	R
22:18	PLL1N	the PLL1 input 5-bit divider.	RW
17:16	PLL1OD	00: divide by 1 01: divide by 2 10: divide by 4 11: divide by 8	RW
15	P1SCS	0: select EXCLK as PLL1 input clock 1: select PLL0 divided clock as PLL1 input clock	RW
12:8	P1SDIV	PLL0 clock dividers as PLL1 input clock.	RW
8	Reserved	Writing has no effect, read as zero.	R
7	PLL1EN	PLL1 Enable. When PLL1EN is set to 1, PLL1 starts to lock phase. After PLL1 stabilizes, PLL1S bit is set. When PLL1EN is clear to 0, PLL1 is	RW

		shut off.	
6	PLL1S	PLL1 Stabilize Flag. 0: PLL1 is off or not stable 1: PLL 1is on and stable	R
5:3	Reserved	Writing has no effect, read as zero.	R
2	LOCK1	0: the PLL output is stable 1: the PLL output is not stable Software should clear this bit to 0, when this bit equal to 1, it indicates that PLL hadn't stable previously , it is only used to debug.	RW
1	PLLOFF	0 : PLL1 doesn't enter shut off state 1: PLL1 is in shut off state	R
0	PLLON	0: PLL1 doesn't enter on state 1: PLL1 is in on state	R

11.2.4.24 PLL Switch and Status Register

The PLL Switch and Status Register (CPPSR) is a 32-bit read/write register, which controls the clock switch ,frequency change mode and reflect the PLL and clock switch Status .It is initialized to 0x80000000 by any reset. Only word access can be used on CPPSR.

CPPSR **0x10000014**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PLLOFF	PLLBP	PLLON	PS	FS	CS	Reserved																							SM	PM	FM
RST	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
31	PLLOFF	0 : PLL doesn't enter shut off state 1: PLL is in shut off state	R
30	PLLBP	0: PLL doesn't enter by pass state 1: PLL is in by pass state	R
29	PLLON	0: PLL doesn't enter on state 1: PLL is in on state	R
28	PS	0: disable PLL or no change PLL parameters 1: enable PLL or change PLL parameters have finished The bit is asserted to 1 auto by hardware . when software concerns this bit, at first software write 0 to the bit, then read the status bit until to 1.	RW
27	FS	Indicate the change frequency has finished . the bit only reflect CDIV, C1DIV, H0DIV, H1DIV, PDIV , H2DIV change. 0: no change 1: change clock parameters have finished	RW

		when software concerns this bit, at first software write 0 to the bit, then read the status bit until to 1.	
26	CS	Indicate the clock switch has finished, the bit reflects when PLL switch to EXCLK or EXCLK to PLL. 0: no clock switch 1: clock switch has finished. when software concerns this bit, at first software write 0 to the bit, then read the status bit until to 1.	RW
25:3	Reserved	Writing has no effect, read as zero.	R
2	SM	0: hardware control 1: when frequency changes, above clocks are all stopped	RW
1	PM	Clock switch mode. When PLL switch to EXCLK or EXCLK switch to PLL. 0: slow mode 1: fast mode	RW
0	FM	Clock frequency change mode. Only to CDIV, C1DIV. 0: slow mode 1: fast mode	RW

11.2.5 PLL Operation

The PLL developed as a macro cell for clock generator. It can generate a stable high-speed clock from a slower clock signal. The output frequency is adjustable and can be up to 1500MHz. The PLL integrates a phase frequency detector (PFD), a low pass filter (LPF), a voltage controlled oscillator (VCO) and other associated support circuitry. All fundamental building blocks as well as fully programmable dividers are integrated on the core. It is useful for clock multiplication of stable crystal oscillator sources and for de-skew clock signals.

The PLL block diagram is shown in following figure:

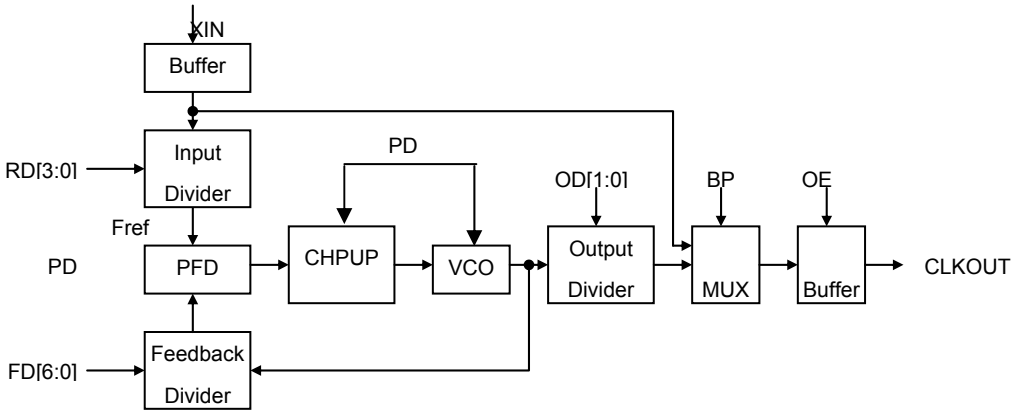


Figure 11-1 Block Diagram of PLL

11.2.5.1 PLL Configuration

PLL Divider Value Setting

There are 3 divider values (N, M and NO) to set the PLL output clock frequency CLKOUT:

1 Input Divider Value N.

N = PLLN of CPPCR

2 Feedback Divider Value M.

M = PLLM of CPPCR

3 Output Divider Value NO.

Output Divider Setting (OD)	Output Divider Value (NO)
0	1
1	2
2	4
3	8

4 The PLL output frequency, FOUT, is determined by the ratio set between the value set in the input divider and the feedback divider. PLL output frequency FOUT is calculated from the following equations:

NF = 1+ M0 + M1*2 + M2*4 + M3*8 + M4*16 + M5*32 + M6*64

NR = 1+ N0*1 + N1*2 + N2*4 + N3*8 + N4*16

NO = $2^{od0+2od1}$

FREF = FIN / NR

FVCO = FOUT * NO

FOUT = FIN * NF / (NR*NO), where FREF is the comparison frequency for the PFD.

For proper operation in normal mode, the following constraints must be satisfied:

For high-band,

10 MHz ⩽ FREF ⩽ 50 MHz

500 MHz ⩽ FVCO ⩽ 1000 MHz

62.5 MHz ⩽ FOUT ⩽ 1000 MHz

For low-band:

10 MHz ⩽ FREF ⩽ 50 MHz

300 MHz ⩽ FVCO ⩽ 600 MHz

37.5 MHz ⩽ FOUT ⩽ 600 MHz

Electrical Characteristics

Junction Temperature = -40 - 125°C, Operating Voltage = typical ± 10% (unless specified otherwise)

PARAMETER	SYM	CONDITION		MIN	TYP	MAX	UNIT
Comparison Frequency	F_{REF}	$F_{REF} = F_{IN} / NR$ [1]		10	--	50	MHz
Input Clock Frequency [2]	F_{IN}	$F_{IN} = NR * F_{REF}$		10	--	400	MHz
Output Clock Frequency	F_{OUT}	$F_{OUT} = F_{VCO} / NO$ [1]	High-band	62.5	--	1000	MHz
			Low-band	37.5		600	
VCO Operating Range	F_{VCO}	$F_{VCO} = F_{REF} * NF$ [1]	High-band	500	--	1000	MHz
			Low-band	300		600	

PARAMETER	SYM	CONDITION			MIN	TYP	MAX	UNIT
Period Jitter (pk-pk) [3, 4]	--	Clean Power	High-band	OD=0	--	--	3.50	% UI
				OD=1			2.00	
				OD=2			1.35	
				OD=3			0.75	
			Low-band	OD=0			2.35	
				OD=1			1.50	
				OD=2			1.00	
				OD=3			0.65	
Cycle-to-Cycle Jitter (max) [3, 4]	--	Clean Power	High-band	OD=0	--	--	3.20	% UI
				OD=1			2.00	
				OD=2			1.30	
				OD=3			0.65	
			Low -band	OD=0			2.10	
				OD=1			1.40	
				OD=2			1.00	
				OD=3			0.55	
Long-term Jitter (pk-pk) [3, 4]	--	Clean Power / 1024-cycle Delay	High-band	OD=0	--	--	490	ps
				OD=1			525	
				OD=2			560	
				OD=3			565	
			Low -band	OD=0			615	
				OD=1			600	
				OD=2			710	
				OD=3			770	
TIE (Time Interval Error) Jitter (rms) [3, 4]	--	Clean Power	High-band	OD=0	--	--	3.00	% UI
				OD=1			1.41	
				OD=2			0.70	
				OD=3			0.36	
			Low -band	OD=0			2.10	
				OD=1			1.03	
				OD=2			0.52	
				OD=3			0.26	

11.2.6 Implementing the Dividers

In Normal Mode, in order for the PLL to function properly, it is necessary to set suitable integer values for the dividers (NR for the input divider, NF for the feedback divider and NO for the output divider). The divider values are set using digital binary inputs of R[4:0], F[6:0] and OD[1:0].

1 Input Divider Value (**NR**).
NR = 16*R4 + 8*R3 + 4*R2 + 2*R1+ R0 + 1 = R[4:0] + 1

2 Feedback Divider Value (**NF**).
NF = 64*F6 + 32*F5 + 16*F4 + 8*F3 + 4*F2 + 2*F1 + F0 + 1 = F[6:0] + 1

3 Output Divider Value (NO).

OD[1:0]	0	1	2	3
NO	1	2	4	8

11.2.7 Programming the Output Clock Frequency

FREF = FIN / NR
FVCO = FOUT * NO
FOUT = FIN * NF / (NR*NO), where FREF is the comparison frequency for the PFD.

For proper operation in normal mode, the following constraints must be satisfied:

For high-band,

10 MHz ⩽ FREF ⩽ 50 MHz
500 MHz ⩽ FVCO ⩽ 1000 MHz
62.5 MHz ⩽ FOUT ⩽ 1000 MHz

For low-band:

10 MHz ⩽ FREF ⩽ 50 MHz
300 MHz ⩽ FVCO ⩽ 600 MHz
37.5 MHz ⩽ FOUT ⩽ 600 MHz

Example I: To synthesize a 800 MHz output clock in high-band with an input frequency FIN = 100 MHz.

1 Set normal mode operation in high-band.
PD = 0, BP = 0, BS = 1

2 Set NR to obtain FREF within the PFD comparison frequency range.
Let FREF = 25 MHz;
Since FREF = FIN / NR,
NR = FIN / FREF = 100 MHz / 25 MHz = 4;
R[4:0] = NR – 1 = 3 = 000112.

3 Set NO and ensure FVCO within the VCO operating range.
Set NO = 1;
FVCO = FOUT * NO;
FVCO = 800 MHz * 1 -> within the VCO operating range in high-band;
-> OD[1:0] = 0 = 002.

4 Set NF to obtain the FOUT frequency.
NF = FOUT * NR * NO / FIN;
NF = 800 MHz * 4 * 1 / 100 MHz = 32;
-> F[6:0] = NF – 1 = 31 = 00111112.

Example II: To synthesize a 500 MHz output clock with an input frequency FIN = 100 MHz. Choose low-band for a better jitter performance.

1 Set normal mode operation in low-band.
PD = 0, BP = 0, BS = 0.

2 Set NR to obtain FREF within the PFD comparison frequency range.
Let FREF = 25 MHz;
Since FREF = FIN / NR,
NR = FIN / FREF = 100 MHz / 25 MHz = 4;
-> R[4:0] = NR – 1 = 3 = 000112.

3 Set NO and ensure FVCO within the VCO operating range.
Set NO = 1;
FVCO = FOUT * NO;
FVCO = 500 MHz * 1 -> within the VCO operating range in low-band;
-> OD[1:0] = 0 = 002.

4 Set NF to obtain the FOUT frequency.
NF = FOUT * NR * NO / FIN;
NF = 500 MHz * 4 *1 / 100 MHz = 20;
-> F[6:0] = NF – 1 = 19 = 00100112.

11.2.8 Main Clock Division Change Sequence

Main clock (CCLK, AUX_CLK, H0CLK, PCLK, H1CLK, H2CLK) frequencies can be changed separately or simultaneously by changing division ratio. Following conditions must be obeyed:

1 CCLK must be integral multiple of H0CLK, H1CLK, H2CLK.
2 AUX_CCLK must be CCLK/2 or equal to H1CLK.
3 H0CLK must be equal to H2CLK or twice of H2CLK.
4 H2CLK must be equal to PCLK or twice of PCLK.
5 H1CLK must be equal to H0CLK or twice of H0CLK.

Don’t violate this limitation, otherwise unpredictable error may occur.

In normal mode, if CE bit of CPCCR is 1, changing CDIV, C1DIV, H0DIV, H2DIV, PDIV ,H1DIV, BCHCDR(BCHM=0) will start a Division Change Sequence immediately. If CE bit of CPCCR is 0, changing above all will not start Division Change Sequence.

11.2.9 Change Other Clock Frequencies

The divider of LCD pixel clock (LPCLK), I2S device clock, SSI device clock, MSC device clock ,USB clock, UHC clock, PCM clock, GPS , BCH and GPU clock can be changed by programming LPCDR , I2SCDR, SSICDR, MSCCDR , USBCDR, UHCCDR, PCMCDR, GPSCDR, BCHCDR(BCHM=1) and GPUCDR respectively.

Change LPCDR I2SCDR SSICDR MSCCDR , USBCDR, UHCCDR, PCMCDR, GPSCDR, BCHCDR(BCHM=1), GPUCDR as following steps:

1 Stop related devices with clock-gate function. Clock supplies to the devices are stopped.
2 Change LPCDR, I2SCDR, SSICDR, MSCCDR, USBCDR, UHCCDR, PCMCDR, GPSCDR, BCHCDR, GPUCDR . If CE is 1, clock frequencies are changed immediately. If CE is 0, clock frequencies are not changed until PLL Multiplier Change Sequence is started.
3 Cancel above clock-gate function.

11.2.10 Change Clock Source Selection

USB, I2S device clocks, PCM device clock, LCD pix clock , MSC clock and SSI clock can be selected from two sources. Before change clock source, corresponding devices should be stopped using clock-gate function.

1 When USB clock source is changed (UCS bit of USBCDR), USB clock should be stopped.
2 When UHC clock source is changed (UHPCS, UHCS bit of UHCCDR), UHC should be stopped.
3 When I2S clock source is changed (I2CS bit of CPCCR), AIC should be stopped.
4 When LCD pix clock source is changed (LSCS LTCS bit of LPCDR), LCD should be stopped.
5 When MSC clock source is changed (MPCS of MSCCDR), MSC should be stopped.
6 When SSI clock source is changed (SCS ,SPCS of SSICDR), SSI should be stopped.
7 When BCH clock source is changed (BPCS of BCHCDR), BCH should be stopped.

When UCS, I2CS , LSCS, LTCS, MCS, SCS, BPCS(BCHM=1) bit is changed, clock source is changed immediately.

When PCS of CPPCR is changed, the corresponding module clock should be stopped.

When P1SCS of CPPCR1 or P1SDIV is changed, the corresponding module should be stopped.

11.2.11 Two PLL Source Selection

USB, I2S, PCM, GPS, GPU, LCD, UHC, MSC, SSI, BCH, CIM source clock can be selected from PLL0 or PLL1. Before change clock source, corresponding devices should be stooped using clock-gate function.

11.2.12 EXCLK Oscillator

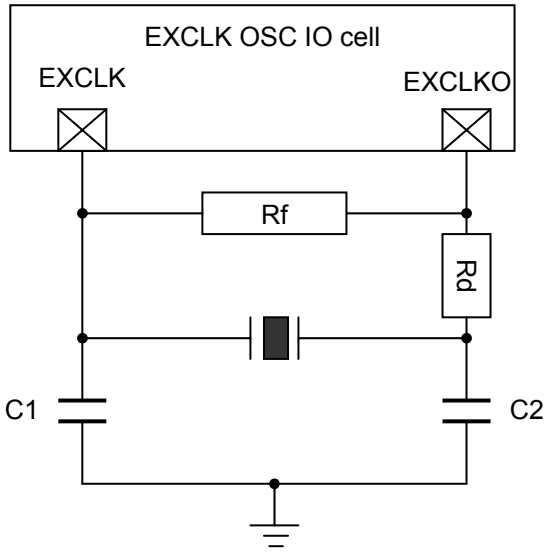


Figure 11-2 Oscillating circuit for fundamental mode

To turn on the oscillator, the oscillating circuit must provide the negative resistance (-Re) at least five times the equivalent series resistance (ESR) of the crystal sample. For larger -Re value, faster turn on the crystal. Higher gm provides larger -Re therefore can start-up the crystal with higher ESR for the same load capacitance (CL). However, it's required higher power consumption.

There are two key parameters to turn on oscillator. Which are CL and the maximum ESR at the target frequency? By reducing the CL, the -Re can be increased thus; shorter turn on time can be achieved. However, if CL is too small, the deviation from the target frequency will increase because of the capacitance variation. So, a trade-off relationship between short turn on time and small frequency deviation in deciding CL value. The smaller ESR of the crystal sample will reduce turn on time but the price is higher. The typical CL and ESR values for difference target frequencies are listed in Table 11-2.

Table 11-2 Typical CL and the corresponding maximum ESR

Target Frequency (Hz)	2M ~ 3M	3M ~ 6M	6M ~ 10M	10M ~ 20M
CL (pf)	25	20	16	12
Maximum ESR (ohm)	1K	400	100	80

Figure 11-2 shows the oscillating circuit is connected with the oscillator I/O cell. Components feedback resistor (Rf), damping resistor (Rd), C1 and C2 are used to adjust the turn on time, keep stability and accurate of the oscillator.

Rf is used to bias the inverter in the high gain region. It cannot be too low or the loop may not oscillate. For mega Hertz range applications, Rf of 1Mohm is applied.
Rd is used to increase stability, low power consumption, suppress the gain in high frequency region and also reduce -Re of the oscillator. Thus, proper Rd cannot be too large to cease the loop oscillating.

C1 and C2 are deciding regard to the crystal or resonator CL specification. In the steady state of oscillating, CL is defined as (C1*C2)/(C1+C2). Actually, the I/O ports, bond pad, and package pin all contribute the parasitic capacitance to C1 and C2. Thus, CL can be rewrite to (C1'*C2')/(C1'+C2'), where C1'=(C1+Cin,stray) and C2'=(C2+Cout,stray). In this case, the required C1 and C2 will be reduced.

Notice, this oscillating circuit is for parallel resonate but not series resonate. Because C1, C2, Rd and Rf are varying with the crystal specifications; therefore there is no single magic number of all the applications.

11.3 Power Manager

In the Low-Power mode, part or whole processor is halted. This will reduce power consumption. The Power Management Controller contains low-power mode control and reset sequence control.

11.3.1 Low-Power Modes and Function

The processor supports six low-power modes and function:

- NORMAL mode

  In Normal mode, all peripherals and the basic blocks including power management block, the CPU core, the bus controller, the memory controller, the interrupt controller, DMA, and the external master may operate completely. But, the clock to each peripheral, except the basic blocks, can be stopped selectively by software to reduce the power consumption.

- DOZE mode

  DOZE mode is entered by setting DOZE bit of LCR to 1. In DOZE mode, clock is burst to CPU core and the clock duty is set by DUTY field of LCR. DOZE mode is canceled by reset, interrupt or clearing DOZE bit to 0. Continuous clock is supplied immediately after DOZE mode is canceled. The other Clocks except CCLK run continuously in DOZE mode.

- IDLE mode

  In IDLE mode, the clock to the CPU core is stopped except the bus controller, the memory controller, the interrupt controller, and the power management block. To exit the IDLE mode, the any interrupts should be activated.

- SLEEP mode

  In SLEEP mode, all clocks except RTC clock are disabled. PLL is disabled also. SLEEP mode is canceled by reset or interrupt. When SLEEP mode is canceled, PLL is restarted , the PLL needs clock stabilization time (PLL lock time). This PLL stabilization time is automatically inserted by the internal logic with lock time count register. and all clocks start operating after PLL stability time.

- CLOCK GATE function

  CLOCK GATE function is used to gate specified on-chip module when it is not used. Set specified CLKG0~40 bits in CLKGR will enter specified CLK gate function. CLOCK gate function is canceled by reset or clearing specified CLKGR0~40 to 0.

- Power down Mode

  In order to reduce power leakage, software may shut down power supply for AHB1 and GPS module. When system enters into SLEEP mode, the software may shut down power for J1 according to OPCR.PD bit.

11.3.2 Register Description

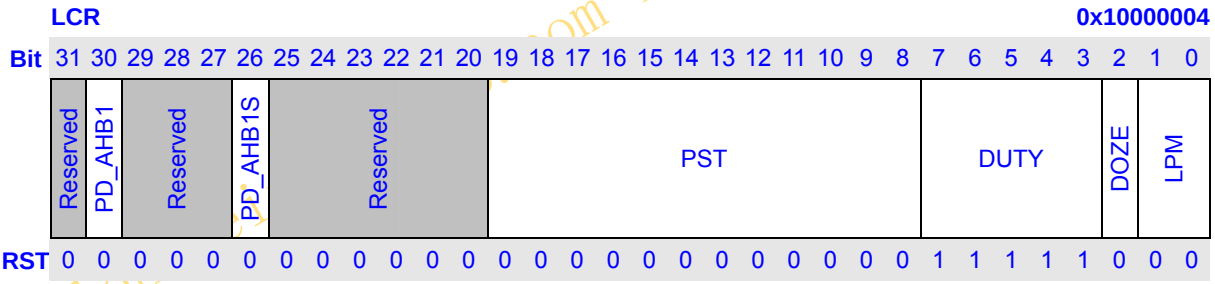
All PMC register 32bit access address is physical address.

Table 11-3 Power/Reset Management Controller Registers Configuration

Name	description	RW	Initial Value	Address	Access Size
LCR	Low Power Control Register	RW	0x000000F8	0x10000004	32
PSWC0ST	Power Switch Chain0 Start Time	RW	0x00000000	0x10000090	32
PSWC1ST	Power Switch Chain1 Start Time	RW	0x00000000	0x10000094	32
PSWC2ST	Power Switch Chain2 Start Time	RW	0x00000000	0x10000098	32
PSWC3ST	Power Switch Chain3 Start Time	RW	0x00000000	0x1000009c	32
CLKGR0	Clock Gate Register0	RW	0x3FFFFFE0	0x10000020	32
OPCR	Oscillator and Power Control Register	RW	0x00001570	0x10000024	32
CLKGR1	Clock Gate Register1	RW	0x0000017F	0x10000028	32

11.3.2.1 Low Power Control Register

The Low Power Control Register (LCR) is a 32-bit read/write register that controls low-power mode status. It is initialized to 0x000000F8 by any reset.



Bits	Name	Description	RW
31	Reserved	Writing has no effect, read as zero.	R
30	PD_AHB1	Power Down Module AHB1. 0: not shut down power supply to AHB1 1: shut down power supply to AHB1	RW
29:27	Reserved	Writing has no effect, read as zero.	R
26	PD_AHB1S	AHB1 power down status. 0: AHB1 module not shut down 1: AHB1 module has entered shut down mode	R
25:20	Reserved	Writing has no effect, read as zero.	R
19:8	PST	Power stability Time. Specifies the Power stabilize time by unit of RTCCLK (approximate 32kHz) cycles.	RW
7:3	DUTY	CPU Clock Duty. Control the CPU clock duty in doze mode. When	RW

		the DUTY field is 0x1F, the clock is always on and when it is zero, the clock is always off. Set the DUTY field to 0 when the CPU will be disabled for an extended amount of time. 00000: 0/31 duty-cycle 00001: 1/31 duty-cycle 00010: 2/31 duty-cycle … 11111: 31/31 duty-cycle	
2	DOZE	Doze Mode. Control the doze mode. When doze mode is canceled, this bit is cleared to 0 automatically. 0: Doze mode is off 1: Doze mode is on	RW
1:0	LPM	Low Power Mode. Specifies which low-power mode will be entered when SLEEP instruction is executed. Bit 1~0: **00**: IDLE mode will be entered when SLEEP instruction is executed **01**: SLEEP mode will be entered when SLEEP instruction is executed **10**: Reserved **11**: Reserved	RW

11.3.2.2 Power Switch Chain0 Start Time Register

PSWC0ST **0x10000090**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																				PSWC0ST											
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.3.2.3 Power Switch Chain1 Start Time Register

PSWC1ST **0x10000094**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																				PSWC1ST											
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.3.2.4 Power Switch Chain2 Start Time Register

PSWC2ST																														0x10000098
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																				PSWC2ST											
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.3.2.5 Power Switch Chain3 Start Time Register

PSWC3ST																														0x1000009C
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																				PSWC3ST											
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

NOTE: The Start Time by the unit of PCLK cycles.

11.3.2.6 Clock Gate Register0

The Clock Gate Register (CLKGR0) is a 32-bit read/write register that controls the CLOCK GATE function of peripherals. It is reset to 0x2FFFFFE0.

CLKGR																														0x10000020
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CLKGR0																															
RST	0	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0

Bits	Name	Description	RW
31:0	CLKGR0	Clock gate Bits. Controls the clock supplies to some peripherals. If set, clock supplies to associated devices are stopped, and registers of the device cannot be accessed also.	RW

Bit	Module	Description
31	AHB_MON	
30	DDR	
29	IPU	After reset period, the clock is stopped.
28	LCD	
27	TVE	After reset period, the clock is stopped.

		26	CIM	After reset period, the clock is stopped.
		25	MDMA	After reset period, the clock is stopped.
		24	UHC	After reset period, the clock is stopped.
		23	MAC	After reset period, the clock is stopped.
		22	GPS	After reset period, the clock is stopped.
		21	DMAC	After reset period, the clock is stopped.
		20	SSI2	After reset period, the clock is stopped.
		19	SSI1	After reset period, the clock is stopped.
		18	UART3	After reset period, the clock is stopped.
		17	UART2	After reset period, the clock is stopped.
		16	UART1	After reset period, the clock is stopped.
		15	UART0	After reset period, the clock is stopped.
		14	SADC	After reset period, the clock is stopped.
		13	KBC	After reset period, the clock is stopped.
		12	MSC2	After reset period, the clock is stopped.
		11	MSC1	After reset period, the clock is stopped.
		10	OWI	After reset period, the clock is stopped.
		9	TSSI	After reset period, the clock is stopped.
		8	AIC	After reset period, the clock is stopped.
		7	SCC	After reset period, the clock is stopped.
		6	I2C1	After reset period, the clock is stopped.
		5	I2C0	After reset period, the clock is stopped.
		4	SSI0	
		3	MSC0	
		2	OTG	
		1	BCH	
		0	NEMC	

11.3.2.7 Clock Gate Regiater1

The Clock Gate Register (CLKGR1) is a 32-bit read/write register that controls the CLOCK GATE function of peripherals. It is reset to 0x0000FFFF.

CLKGR1 **0x10000028**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																CLKGR															
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	Name	Description	RW
31:16	Reserved	Writing has no effect, read as zero.	R
15:0	CLKGR1	Clock gate Bits. Controls the clock supplies to some peripherals. If set,	

clock supplies to associated devices are stopped, and registers of the device cannot be accessed also.

Bit	Module	Description
15	I2C2	After reset period, the clock is stopped.
14	AUX	After reset period, the clock is stopped.
13	I2S2CH	After reset period, the clock is stopped.
12	OSD	After reset period, the clock is stopped.
11	Reserved	Writing has no effect, read as zero.
10	PCM1	After reset period, the clock is stopped.
9	GPU	After reset period, the clock is stopped.
8	PCM0	After reset period, the clock is stopped.
7	VPU	After reset period, the clock is stopped.
6	CABAC	After reset period, the clock is stopped.
5	SRAM	After reset period, the clock is stopped.
4	DCT	After reset period, the clock is stopped.
3	ME	After reset period, the clock is stopped.
2	DBLK	After reset period, the clock is stopped.
1	MC	After reset period, the clock is stopped.
0	BDMA	After reset period, the clock is stopped.

NOTE: CLKGR1(6-1) bits has no effect , it don't care.

11.3.2.8 Oscillator and Power Control Register (OPCR)

The Oscillator and Power Control Register is a 32-bit read/write register that specifies some special controls to oscillator and analog block. It is initialized to 0x00001570 by reset.

OPCR **0x10000024**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	IDLE_DIS	AHB1_ST	AHB1_STP	AHB1_ACK	Reserved												O1ST								SPENDN	GPSEN	SPENDH	O1SE	PD	ERCS	Reserved	
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	1	1	1	0	0	0	0	0

Bits	Name	Description	RW
31	IDLE_DIS	0: when CPU enters idle mode, CPUI clock is stopped 1: When CPU enters idle mode, CPUI clock is not stopped	RW
30	AHB1_ST	0: AHB1 does not enter soft reset mode 1: AHB1 enters soft reset mode	RW
29	AHB1_STP	Request for AHB1 to Stop bus transfer.	RW
28	AHB1_ACK	AHB1 Stop Ack.	R

27:16	Reserved	Writing has no effect, read as zero.	R
15:8	O1ST	EXCLK Oscillator Stabilize Time. This filed specifies the EXCLKoscillator stabilize time by unit of 16 RTCCLK periods (oscillator stable time O1ST × 16 / 32768) cycles. It is initialized to H'15.	RW
7	SPENDN	force OTG phy to enter suspend mode. 0: OTG phy has forced to entered SUSPEND mode 1: OTG phy hasn't forced to entered SUSPEND mode	R
6	GPSEN	0: Disable GPS module 1: Enable GPS module	RW
5	SPENDH	Force UHC phy to enter suspend mode. 0: UHC phy hasn't forced to entered SUSPEND mode 1: UHC phy has forced to entered SUSPEND mode	RW
4	O1SE	EXCLK Oscillator Sleep Mode Enable. This filed controls the state of the EXCLK oscillator in Sleep mode. 0: EXCLK oscillator is disabled in Sleep mode 1: EXCLK oscillator is enabled in Sleep mode	RW
3	PD	The fief controls the state P0 in Sleep mode. 0: The P0 not power down in Sleep mode 1: The P0 power down in Sleep mode	RW
2	ERCS	EXCLK/512 clock and RTCLK clock selection. 0: select EXCLK/512 division ration clock 1: select RTCLK clock the clock only output to CPM INTC SSI TCU etc.	RW
1:0	Reserved	Writing has no effect, read as zero.	R

11.3.3 Doze Mode

Firstly, software should set the DUTY bits of LCR. Then set DOZE bit of LCR to 1 to enter doze mode. When slot controller of PMC indicates that the CPU clock's time-slot has expired, CPU is halted but its register contents are retained. During doze mode, program can modify clock duty-cycle according to core resource requirement. Clock control is in increments of approximately 3% (1/31).

Doze is exited by software, interrupt, reset or SLEEP instruction.

11.3.4 IDLE Mode

In normal mode, when LPM bits in LCR are 0 and SLEEP instruction is executed, the processor enters idle mode. CPU is halted but its register contents are retained All critical application must be finished and peripherals must be configured to generate interrupts when they need CPU attention.

The procedure of entering sleep mode is shown blow:

1 Set LPM bits in LCR to 0.
2 Executes SLEEP instruction.

3 When current operation of CPU core has finished and CPU core is idle, CCLK supply to CPU core is stopped.

IDLE mode is exited by an interrupt (IRQ or on-chip devices) or a reset.

11.3.5 SLEEP Mode

In normal mode, when LPM bits in LCR is 1 and SLEEP instruction is executed, the processor enter SLEEP mode. CPU and on-chip devices are halted, except some wakeup-logic. PLL is shut off. Clock output from CKO pin is also stopped. SDRAM content is preserved by driving into self-refresh state. CPU registers and on-chip devices registers contents are retained.

Before enter SLEEP mode, software should ensure that all peripherals are not running. The procedure of entering SLEEP mode is shown blow:

1 Set LPM bit in LCR to 1.
2 Execute a SLEEP instruction.
3 When current access on system bus complete, the arbiter will not grant any following request. EMC will drive SDRAM from auto-refresh mode to self-refresh mode.
4 When system bus is idle state and SDRAM is self-refresh mode, internal clock supplies are stopped.
5 SLEEP mode can be exited by an interrupt (IRQ or on-chip devices), WDT reset or a poweron reset via the RESETP pin.

11.3.6 Power Down Mode

When PD_AHB1/PD_GPS bit in LCR is 1 , the processor enters shut down AHB1/GPS module power sequence.

When PD_AHB1S/PD_GPSS bit in LCR is 1, it indicates that the AHB1/GPS module has been shut off. The leakage current of AHB1/GPS is reduced almost to 0.

When enter sleep mode, when PD bits in OPCR is 1, the J1 power supply would be shut off. The leakage current of J1 is reduced almost to 0.

The procedure of entering Power Down mode is shown blow:

1 set proper values for PSWC0ST, PSWC1ST, PSWC2ST, PSWC3ST.
2 set PD_AHB1/PD_GPS bit in LCR to 1.
3 wait until PD_AHB1S/PD_GPSS = 1.
4 When need for supply power for AHB1/GPS, set PD_AHB1/PD_GPS in LCR to 0.
5 wait until PD_AHB1S/PD_GPSS = 0.
6 the hardware auto generate RESET signal to the module, the software must again config the module as the same to POWER ON Reset.

11.4 Reset Control Module

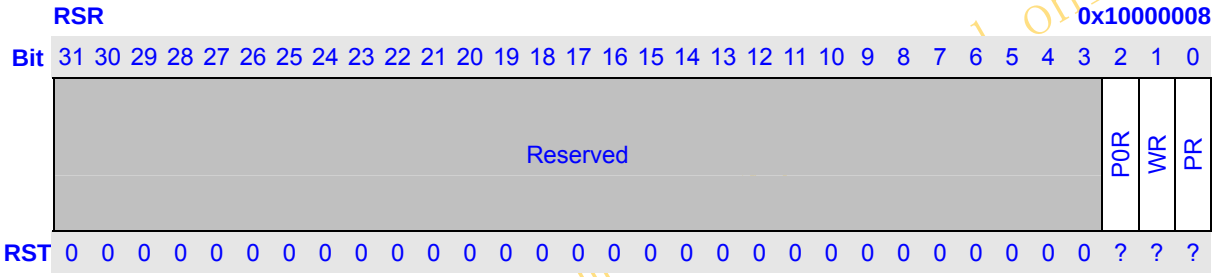
11.4.1 Register Description

All RCM register 32bit access address is physical address.

Name	description	RW	Initial Value	Address	Access Size
RSR	Reset Status Register	RW	0x????????	0x10000008	32

11.4.1.1 Reset Status Register (RSR)

The Reset Status Register (RSR) is a 32-bit read/write register which records last cause of reset. Each RSR bit is set by a different source of reset. Please refer to Reset Sequence Control for reset sources description.



Bits	Name	Description	RW
31:2	Reserved	Writing has no effect, read as zero.	R
2	P0R	P0 power up Reset. It indicates that P0 has been shut down, now it has been power up. When P0 reset is detected, P0R is set and remains set until software clears it or another reset occurs. This bit can only be written with 0. Write with 1 will be ignored. 0: P0 reset has not occurred since the last time the software clears this bit 1: P0 reset has occurred since the last time the software clears this bit	RW
1	WR	WDT Reset. When a WDT reset is detected, WR is set and remains set until software clears it or another reset occurs. This bit can only be written with 0. Write with 1 will be ignored. 0: WDT reset has not occurred since the last time the software clears this bit 1: WDT reset has occurred since the last time the software clears this bit	RW
0	PR	Power On Reset. When a poweron reset via PRESET pin is detected, PR is set and remains set until software clears it or another reset occurs. This bit can only be written with 0. Write with 1 is ignored. 0: Power on reset has not occurred since the last time the software clears	RW

		this bit 1: Power on reset has occurred since the last time the software clears this bit	

11.4.2 Power On Reset

Power on reset is generated when PRESET pin is driven to low. Internal reset is asserted immediately. All pins return to their reset states. The Power on reset is extended to 40MS.

PRESET pin must be held low until power stabilizes and the EXCLK oscillator stabilize. CPU and peripherals are clocked by EXCLK oscillator output directly. PLL is reset to off state. All internal modules are initialized to their predefined reset states.

11.4.3 WDT Reset

WDT reset is generated when WDT overflow. Internal reset is asserted within two RTCCLK cycles. All pins return to their reset states.

Then WDT reset source is cleared because of internal reset. The internal reset is asserted for about 10 milliseconds. CPU and peripherals are clocked by EXCLK oscillator output directly. PLL is reset to off state.

12 Real Time Clock

12.1 Overview

The Real-Time Clock (RTC) unit can be operated in either chip main power is on or the main power is down but the RTC power is still on. In this case, the RTC power domain consumes only a few micro watts power.

The RTC contains a 32768Hz oscillator, the real time and alarm logic, and the power down and wakeup control logic.

12.1.1 Features

RTC module has following features:

- Embedded 32768Hz oscillator for 32k clock generation with an external 32k crystal
- RTCLK selectable from the oscillator or from the divided clock of EXCLK, so that 32k crystal can be absent if the hibernating mode is not needed
- 32-bits second counter
- Programmable and adjustable counter to generate accurate 1 Hz clock
- Alarm interrupt, 1Hz interrupt
- Stand alone power supply, work in hibernating mode
- Power down controller
- Alarm wakeup
- External pin wakeup with up to 2s glitch filter

12.1.2 Signal Descriptions

RTC has 5 signal IO pins and 1 power pin. They are listed and described in.

Pin Names	Pin Loc	IO	IO Cell Char.	Pin Description	Power
RTCLK		AI	32768Hz	RTCLK: 32768 clock input or OSC input	VDD_{RTC}
RTCLKO		AO		RTCLKO: OSC output	VDD_{RTC}
PWRON		AO	~2mA, Open-Draw	PWRON: Power on/off control of main power	VDD_{RTC}
WKUP_		AI	Schmitt	WKUP_: Wake signal after main power down	VDD_{RTC}
PPRST_		AI	Schmitt	PPRST_: RTC power on reset and RESET-KEY reset input	VDD_{RTC}
VDDRTC		P		VDDRTC: 3.3V power for RTC and hibernating mode controlling that never power down	-
CLK32K PD14		O IO	8mA pullup-pe	32768Hz clock output PD14: GPIO group D bit14. When main power down, this pin is controlled by RTC register	VDD_{RTC}

Pin Names	Pin Loc	IO	IO Cell Char.	Pin Description	Power
LDO_CAP		AIO		LDO_CAP: Capacitor pin for RTC LDO	

RTCLK/RTCLKO pins. We have an embedded oscillator for 32768Hz crystal. These two pins are the crystal XTALI and XTALO connection pins. If an input clock is used instead, please input it to RTCLK pin.

If do not use any clock, hibernate mode will be NOT available any more, and the time will lose if power down.

PWRON pin: this pin is used to control the main power on/off. Output high voltage means on and 0 means off.

WKUP_ pin: hibernating mode wakeup input. (Default low active)

PPRST_ pin: This pin should be set to low voltage only in two cases.

– When RTC power is turned on. (so that whole chip is power on)

– A RESET-KEY is pressed.

CLK32PD14 pin: output 32.768KHz RTC clock and can used as GPIO.

LDO_CAP pin: needed one 1nF decoupling capacitor on the PCB board.

12.2 Register Description

Table 12-1 Registers for real time clock

Name	Description	RW	Reset Value	Address	Access Size
RTCCR	RTC Control Register	RW	0x00000081*1*2	0x10003000	32
RTCSR	RTC Second Register	RW	0x????????	0x10003004	32
RTCSAR	RTC Second Alarm Register	RW	0x????????	0x10003008	32
RTCGR	RTC Regulator Register	RW	0x0???????	0x1000300C	32

NOTES:

1 *1: Unless otherwise stated, the reset value is for PPRST_ and Hibernating wakeup reset. WDT reset doesn't change the value.

2 *2: The reset value can be either of 0x00000081, 0x00000091, 0x00000089, 0x00000099.

Table 12-2 Registers for hibernating mode

Name	Description	RW	Reset Value	Address	Access Size
HCR	Hibernate Control Register	RW	0x00000000*1	0x10003020	32
HWFCR	Wakeup filter counter Register in Hibernate mode	RW	0x0000???0	0x10003024	32
HRCR	Hibernate reset counter Register in Hibernate mode	RW	0x00000??0	0x10003028	32
HWCR	Wakeup control Register in Hibernate mode	RW	0x00000008*1	0x1000302C	32
HWRSR	Wakeup Status Register in Hibernate mode	RW	0x00000000*1	0x10003030	32
HSPR	Scratch pattern register	RW	0x????????	0x10003034	32
WENR	Write enable pattern register	RW	0x00000000	0x1000303C	32
CKPCR	Configure the CLK32K pin value	RW	0x00000010	0x10003040	32
PMCR	Identify the RTC battery has been removed	RW	0x00000000	0x10003044	32

NOTE:

*1: Unless otherwise stated, the reset value is for PPRST_ and Hibernating wakeup reset. WDT reset doesn't change the value.

All these registers, include those for real time clock and for hibernating mode control, except otherwise stated, are implemented in RTCLK clock domain. When write to these registers, it needs about 1 ~ 2 RTCLK cycles to actually change the register's value and needs another RTCLK cycle to allow the

next write access. A bit RTCCR.WRDY is used to indicate it. When RCR.WRDY is 1, it means the previous write is finished, a right value can be read from the target register, and a new write access can be issued. So before any write access, please make sure RCR.WRDY = 1.

12.2.1 RTC Control Register (RTCCR)

RTCCR contains bits to configure the real time clock features. Unless otherwise stated, the reset value is for PPRST_ and Hibernating wakeup reset. WDT reset doesn't change the value.

RTCCR **0x10003000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																								WRDY	1HZ	1HZIE	AF	AIE	AE	SELEXC	RTCE
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1*[1]	0	0*[1]	?	0	?	0*[1]	1

NOTE:

*[1]: These bits are reset in all resets: PPRST_ input pin reset, hibernating reset and WDT reset.

Bits	Name	Description	RW
31:7	Reserved	Writing has no effect, read as zero.	R
7	WRDY	Write ready flag. It is 0 when a write is currently processing and the value has not been written to the writing target register. No write to any RTC registers can be issued in this case, or the result is undefined. The read value from the target register is also undefined. The reading is meaningful and another write can be issued when it is 1. Please reference to descriptions in 12.2 for some more details. This bit is read only and write to it is ignored.	R
6	1HZ	1Hz flag. This bit is set by hardware once every 1 second through the 1Hz pulse if the real time clock is enabled (RTCCR.RTCE = 1). This bit can be cleared by software. Write 1 to this bit is ignored.	RW
5	1HZIE	1Hz interrupt enable. Writing to this bit takes effect immediately without delay. 1HZIE: 0 — 1Hz interrupt is disabled. 1HZIE: 1 — 1Hz interrupt is enabled. RTC issues interrupt when 1HZ bit is set.	RW
4	AF	Alarm flag. This bit is set by hardware when alarm match (RTCSR = RTCSAR) is found and alarm is enabled (RTCCR.AE = 1) and the real time clock is enabled (RTCCR.RTCE = 1). This bit can be cleared by software. Write 1 to this bit is ignored. Writing to this bit takes effect immediately.	RW
3	AIE	Alarm interrupt enable.	RW

		AIE / **Description** 0 / Alarm interrupt is disabled. 1 / Alarm interrupt is enabled. RTC issues interrupt when AF is set.	
2	AE	Alarm enable. **AE** / **Description** 0 / Alarm function is disabled. 1 / Alarm function is enabled.	RW
1	SELEXC	The divided EXCLK is selected as RTCLK in rtc-hiber module. **SELEXC** / **Description** 0 / OSC32K or RTCLK input clock is selected as RTCLK in rtc-hiber module. 1 / The divided EXCLK is selected as RTCLK in rtc-hiber module. NOTE: If do not use any 32Khz clock (either input clock or using crystal), hibernate mode will be NOT available any more, and the time will lose if power down. CPM.OPCR.ERCS must be 0, when using SELEXC = 1. When the main chip power down, SELEXC will be 0 in internal circuit, in this time, RTCLK will use OSC32K clock.	RW
0	RTCE	Real time clock enable. **RTCE** / **Description** 0 / Real time clock function is disabled. 1 / Real time clock function is enabled.	RW

12.2.2 RTC Second Register (RTCSR)

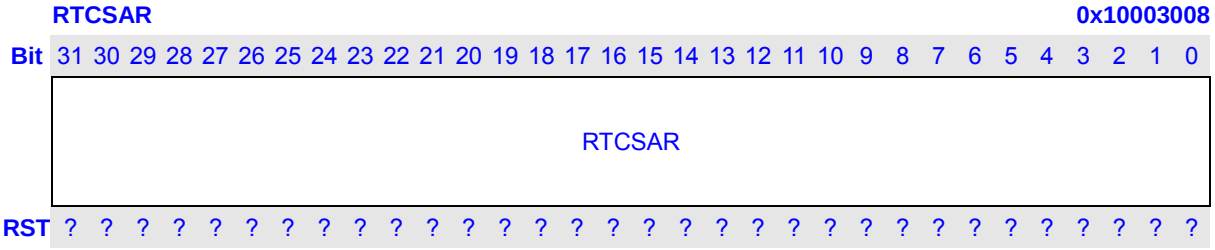
RTCSR is a 32-bit width second counter. It can be read and write by software. It is increased by 1 at every 1Hz pulse if the real time clock is enabled (RTCCR.RTCE = 1). When read, it should be read continued more than once and take the value if the adjacent results are the same. RTCSR is not initialized by any reset.

RTCSR **0x10003004**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RTCSR																															
RST	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?

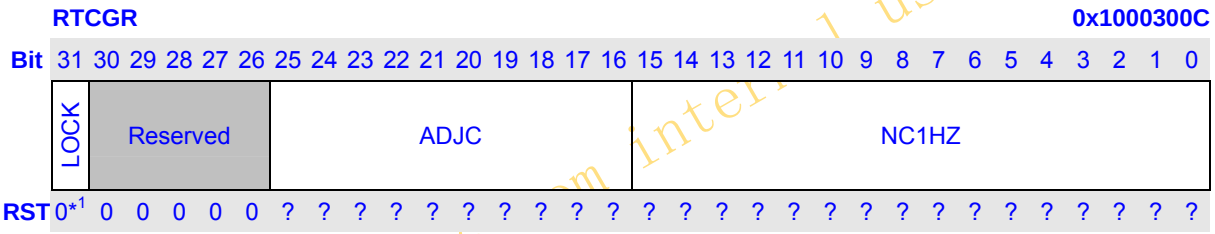
12.2.3 RTC Second Alarm Register (RTCSAR)

RTCSAR serves as a second alarm register. Alarm flag (RTCCR.AF) is set to 1 when the RTCSR equals the RTCSAR in the condition of alarm is enabled (RTCCR.AE = 1) and the real time clock is enabled (RTCCR.RTCE = 1). RTCSAR can be read and write by software and is not initialized by any reset.



12.2.4 RTC Regulator Register (RTCGR)

RTCGR is serves as the real time clock regulator, which is used to adjust the interval of the 1Hz pulse.



NOTE:

*[1]: This bit is reset in all resets: PPRST_ input pin reset, hibernating reset and WDT reset.

Bits	Name	Description	RW
31	LOCK	Lock bit. This bit is used to safeguard the validity of the data written into the RTCGR register. Once it is set, write to RTCGR is ignored. This bit can only be set by software and cleared by (any type of) resets. LOCK: 0 – Write to RTCGR is allowed. LOCK: 1 – Write to RTCGR is forbidden.	RW
30:26	Reserved	Writing has no effect, read as zero.	R
25:16	ADJC	This field specifies how many times it needs to add one 32kHz cycle for the 1Hz pulse interval in every 1024 1Hz pulses. In other word, among every 1024 1Hz pulses, ADJC number of them are trigged in every (NC1HZ + 2) 32kHz clock cycles, (1024 – ADJC) number of them are trigged in every (NC1HZ + 1) 32kHz clock cycles.	RW
15:0	NC1HZ	This field specifies the number plus 1 of the working 32kHz clock cycles are contained in the 1Hz pulse interval. In other word, 1Hz pulse is trigged every (NC1HZ + 1) 32kHz clock cycles, if RTCGR.ADJC = 0.	RW

12.2.5 Hibernate Control Register (HCR)

HCR contains the bit to control the main chip power on/off.

HCR **0x10003020**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																															PD
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
31:1	Reserved	Writing has no effect, read as zero.	R
0	PD	Power down or power on bit. Besides writing by CPU, this bit will be set to 1 if an unknown reason main power supply off is detected. This bit controls the PWRON pin level. When co-working with some external components, this bit is used for power management of this chip. It is supposed when 1 is written to this bit, the main power supply of the chip, except RTC power, will be shut down immediately. After this bit is set to 1, all registers in RTC module, except RTCCR.1HZ and RTCCR.1HZIE, cannot be changed by write access. This bit is cleared by reset pin reset and hibernating reset. The later one is asserted by wakeup procedure.	RW

PD	PWRON	Description
0	VDDRTC	No power down, keep power on.
1	0 V	Power down enable, turn power off.

12.2.6 HIBERNATE mode Wakeup Filter Counter Register (HWFCR)

The HIBERNATE mode Wakeup Filter Counter Register (HWFCR) is a 32-bit read/write register .It filter the glitch generated by a dedicated wakeup pin. The HWFCR is not initialized by any reset.

HWFCR **0x10003024**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																HWFCR											Reserved					
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0

Bits	Name	Description	RW
31:16	Reserved	Writing has no effect, read as zero.	R
15:5	HWFCR	Wakeup pin effective minimum time in number of 32 RTCLK cycles, used as glitch filter logic. Maximum of 2 seconds if the RTCLK is 32768Hz If this value is configured to 0, and the pin keeps low longer than 15	RW

		RTCLK periods, it wakes up RTC from Hibernate.	
4:0	Reserved	Writing has no effect, read as zero.	R

12.2.7 Hibernate Reset Counter Register (HRCR)

The Hibernate Reset Counter Register is a 32-bit read/write register that specifies hibernate reset assertion time. The HRCR is initialized by PPRST_ .

HRCR **0x10003028**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																HRCR											Reserved					
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	?	?	?	?	?	?	?	0	0	0	0	0

Bits	Name	Description	RW
31:12	Reserved	Writing has no effect, read as zero.	R
11:5	HRCR	HIBERNATE Reset waiting time. Number of 32 RTCLK cycles. Maximum 125 ms if the RTCLK is 32768Hz. If this value is configured to 0, it will generate 31 RTCLK HIBERNATE Reset.	RW
4:0	Reserved	Writing has no effect, read as zero.	R

12.2.8 HIBERNATE Wakeup Control Register (HWCR)

The HIBERNATE Wakeup Control Register is a 32-bit read/write register that controls real time clock alarm wake up enable. The reset value is for PPRST_ and Hibernating wakeup reset. WDT reset doesn't change the value.

HWCR **0x1000302C**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																												EPDET	WKUPVL	Reserved	EALM
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0

Bits	Name	Description	RW
31:4	Reserved	Writing has no effect, read as zero.	R
3	EPDET	Power detect enable. 0: disable 1: enable (default)	RW
2	WKUPVL	RTC wakeup pin valid level.	RW

		0: Low level sensitive (default) 1: High sensitive	
1	Reserved	Writing has no effect, read as zero.	R
0	EALM	RTC Alarm wakeup enable. 0: disable 1: enable	RW

12.2.9 HIBERNATE Wakeup Status Register (HWRSR)

The HIBERNATE Wakeup Status Register is a 32-bit read/write register that reflects wakeup status bits.

HWRSR **0x10003030**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																							APD	Reserved		HR	PPR	Reserved		PIN	ALM
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	?	0	0	1*2	1*1	0	0	?	?

NOTES:

1 *1: This reset value only for PPRST_. It is undefined in case of other resets.

2 *2: This reset value only for HRST_. It is undefined in case of other resets.

Bits	Name	Description	RW
31:9	Reserved	Writing has no effect, read as zero.	R
8	APD	Accident power down. When the software has not set to HIBERNATE state, the core power is down, then an accident power down is detected. APD is set and remains set until software clears it. This bit can only be written with 0. Write with 1 is ignored. **HR** / **Description** 0: Accident power down has not occurred since the last time the software clears this bit. 1: Accident power down has occurred since the last time the software clears this bit.	RW
7:6	Reserved	Writing has no effect, read as zero.	R
5	HR	Hibernate Reset. When a Hibernate reset detected, HR is set and remains set until software clears it or another reset occurs. This bit can only be written with 0. Write with 1 is ignored. **HR** / **Description** 0: Hibernate reset has not occurred since the last time the software clears this bit. 1: Hibernate reset has occurred since the last time the	RW

		software clears this bit.	
4	PPR	PAD PIN Reset. When a PPRST_ is detected, PPR is set and remains set until software clears it or another reset occurs. This bit can only be written with 0. Write with 1 is ignored. **PPR** / **Description** 0: PPRST_ reset has not occurred since last time the software clears this bit. 1: PPRST_ reset has occurred since last time the software clears this bit.	RW
3:2	Reserved	Writing has no effect, read as zero.	R
1	PIN	Wakeup Pin Status bit. The bit is cleared when chip enters hibernating mode. It is set when exit the hibernating mode by wakeup pin. This bit can only be written with 0. Write with 1 is ignored.	RW
0	ALM	RTC Alarm Status bit. The bit is cleared when chip enters hibernating mode. It is set when exit the hibernating mode by alarm. This bit can only be written with 0. Write with 1 is ignored.	RW

12.2.10 Hibernate Scratch Pattern Register (HSPR)

This is a scratch register used to hold a pattern. The software can check the pattern is kept to know whether RTC power has ever been down and whether it is needed to setup the real time clock.

HSPR 0x10003034

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PAT																															
RST	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?

Bits	Name	Description	RW
31:0	PAT	The pattern.	RW

12.2.11 Write Enable Pattern Register (WENR)

This is a scratch register used to hold a pattern. The software can check the pattern is kept to know whether RTC power has ever been down and whether it is needed to setup the real time clock.

WENR 0x1000303C

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	WEN	Reserved															WENPAT															
RST	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?

Bits	Name	Description	RW
31	WEN	The write enable flag. If the WENPAT is 0xA55A then this bit will be 1. When the WEN changes to 1, the RTCCR, RTCSR, RTCSAR, RTCGR, HCR, HWFCR, HRCR, HWCR, HWRSR, HSPR registers could be changed. But RTCCR.SELEXC, RTCCR.HZIE, RTCCR.WRDY may change in any time. This bit is read only and write to it is ignored. There is an exception, when system does NOT have RTC 32Khz crystal. MUST write 1 to RTCCR.SELEXC before write to any value to any other registers. **WEN** / **Description** 0 — Other RTC registers is locked, write these registers will be ignored. 1 — Other RTC registers can be changed.	R
30:16	Reserved	Writing has no effect, read as zero.	R
15:0	WENPAT	The write enable pattern. Before writing any value to RTCCR, RTCSR, RTCSAR, RTCGR, HCR, HWFCR, HRCR, HWCR, HWRSR, HSPR registers, write 0xA55A to WENPAT to set these register writable. If this value is ok, WEN will change to 1. But RTCCR.SELEXC and RTCCR.HZIE are writable in any time. These bits are write-only, always read as 0.	W

12.2.12 CLK32K Pin control register (CKPCR)

This is a CLK32K pin control register used to configure the CLK32K pin value. The CKPCR is initialized by PPRST_ .

CKPCR **0x10003040**

Bit	31–7	6	5	4	3	2	1	0
	Reserved		CK32RD	CK32PULL	Reserved	CK32CTL		CK32D
RST	?	?	?	1	0	0	0	0

Bits	Name	Description	RW
31:3	Reserved	Writing has no effect, read as zero.	R
5	CK32RD	Read this bit will return CLK32K pin status.	R
4	CK32PULL	Pull up configures. **CK32PULL** / **Description** 0 — Pull Up is enabled.	RW

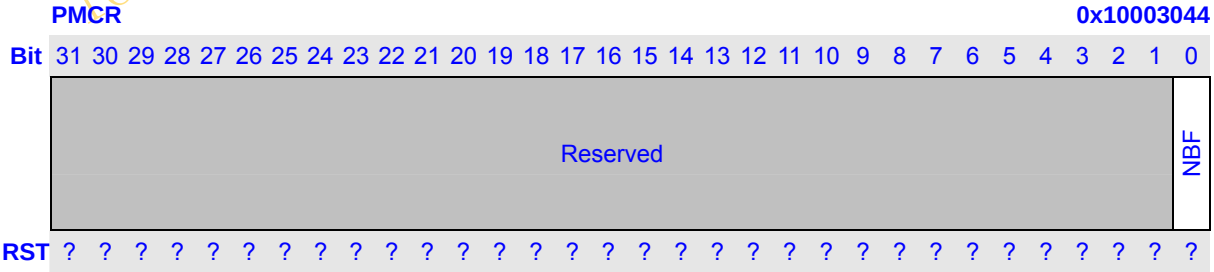
		1	Pull Up is disabled.
3	Reserved	Writing has no effect, read as zero.	R
2:1	CK32CTL	Output RTCLK to CLK32K pin.	RW

CK32CTL	Description
00	CLK32K pin is set to general input. The pin value can be read by CK32RD bit or GPIO PD14 bit. The input pin should not be left floating, if pull up (CK32PULL) is disabled. The CLK32K pin is only set as input in HIBERNATE mode.
01	CLK32K pin is set to general output. The pin output value is set by CK32D bit.
10	GPIO PD14 controls CLK32K pin. The pin output is set by GPIO PD14 bit. The CLK32K pin is output CK32D bit in HIBERNATE mode.
11	Output RTCLK to CLK32K pin. If this set, CLK32K pin will always output 32K clock, even in HIBERNATE mode.

0	CK32D	When CK32CTL is configured to general output or input (HIBERNATE), Write to this pin will output to CLK32K pin, if configured.	RW

12.2.13 PMCR Power Monitor register (PMCR)

This is a register used to identify the RTC battery has been removed. The software can check the register to know whether RTC battery has ever been down and whether it is needed to setup the real time clock. The PMCR is not initialized by any reset.



Bits	Name	Description	RW
31:1	Reserved	Writing has no effect, read as zero.	R
0	NBF	No RTC battery flag. Write 1 to this register will update this flag. Write 0 to this register will be clear this flag.	RW

12.3 Time Regulation

Because of the inherent inaccuracy of crystal and other variables, the time counter may be inaccurate. This requires a slight adjustment. The application processor, through the RTCGR, lets you adjust the 1Hz time base to an error of less than 1ppm. Such that if the Hz clock were set to be 1Hz, there would be an error of less than 5 seconds per month.

To determine the value programmed into the RTCGR, you must first measure the output frequency at the oscillator multiplex (approximately 32 kHz) using an accurate time base, such as a frequency counter. This clock is externally visible by selecting the alternate function of GPIO.
To gain access to the clock, program this pin as an output and then switch to the alternate function. To trim the clock, divide the output of the oscillator by an integer value and fractional adjust it by periodically deleting clocks from the stream driving this integer divider.

After the true frequency of the oscillator is known, it must be split into integer and fractional portions. The integer portion of the value (minus one) is loaded into the NC1HZ field of the RTCGR.

The fractional part of the adjustment is done by periodically deleting clocks from the clock stream driving the Hz divider. The trim interval period is hardwired to be 1024 1Hz clock cycles (approximately 17 minutes). The number of clocks (represented by ADJC field of RTCGR) are deleted from the input clock stream per trim interval. If ADJC is programmed to be zero, then no trim operations occur and the RTC is clocked with the raw 32 kHz clock. The relationship between the Hz clock frequency and the nominal 32 kHz clock (f1 and f32K, respectively) is shown in the following equation.

$$f1 = \frac{2^{10} \times (NC1HZ + 1)}{2^{10} \times (NC1HZ + 1) + ADJC} \times \frac{f32k}{NC1HZ + 1}$$

f1 = actual frequency of 1Hz clock
f32k = frequency of either 32.768KHz crystal output or 3.6864MHz crystal output further divided down to 32.914KHz

12.3.1 HIBERNATE Mode

First make sure RTCCR.SELEXC is 0.

When Software writes 1 to PD bit of HCR, the system at once enters HIBERNATE mode. The powers of CORE and IO are disconnected by PWRON pin, no power consumption to core and IO. When a wakeup event occurs, the core enters through a hibernate reset. Only CPM wake up logic and RTC is operating in HIBERNATE mode.

12.3.1.1 Procedure to Enter HIBERNATE mode

Before enter HIBERNATE mode, software must complete following steps:

1 Finish the current operation and preserve all data to flash.

2 Configure the wake-up sources properly by configure HWCR.

3 Set HIBERNATE MODE. (Set PD bit in HCR to 1)

12.3.1.2 Procedure to Wake-up from HIBERNATE mode

1 The internal hibernate reset signal will be asserted if one of the wake-up sources is issued.

2 Check RSR to determine what caused the reset.

3 Check PIN/ALM bits of HWRSR in order to know whether or not the power-up is caused by which wake-up from HIBERNATE mode.

4 Configure the SDRAM memory controller.

5 Recover the data from flash.

12.4 Clock select

There could be two clock input to RTC internal clock called rtclk. One is OSC32k clock; the other is EXCLK/512.

The software MUST make sure the RTC run in valid clock configuration.

Table 12-3 Clock select registers

RTCCR.SELEXC	CPM.ERCS	Description	Valid
0	0	RTC use OSC32K clock.	OK
0	1		OK
1	0	RTC use EXCLK/512 clock.	OK
1	1	RTC will lost clock. (Not Valid)	NO

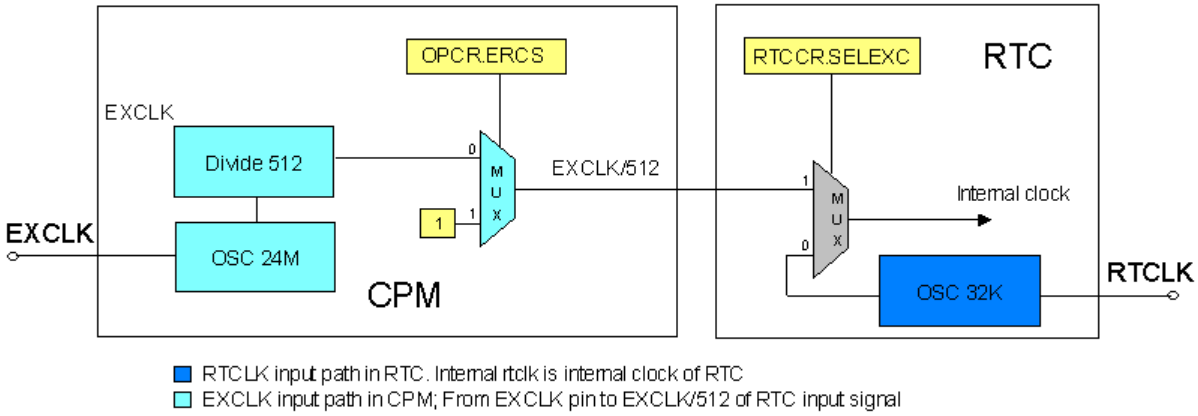


Figure 12-1 RTC clock selection path

Changing RTCLK sequence:

1 There are both 32KHz crystal and 24Mhz EXCLK crystal connected, so RTCLK input path has 32Khz clock.

In this case, there is no need to change internal clock, so do NOT change SELEXC all the time.

2 There is no 32KHz crystal connected but only 24Mhz EXCLK crystal connected, so RTCLK input path has no clock.

In this case, should flow the sequence below to change internal clock:

a Set OPCR.ERCS of CPM to 1; close EXCLK/512 to RTC.

b Set CLKGR.RTC of CPM to 1; close PCLK to RTC.

c Set RTCCR.SELEXC to 1; change internal clock to EXCLK/512.

d Wait two clock period of clock.

e Clear OPCR.ERCS of CPM to 0; open EXCLK/512 to RTC.

f Clear CLKGR.RTC of CPM to 0; open PCLK to RTC.

g Configure all RTC registers but RTCCR.SELEXC.

h Check RTCCR.SELEXC == 1.

i IF YES, finish this sequence; IF NO, do step (1) again.

NOTE: If using HIBERNATE mode, MUST have both 32KHz crystal (or input 32Khz clock) and 24Mhz EXCLK crystal connected, or RTC time will be insignificant.

13 Interrupt Controller

13.1 Overview

This chapter describes the interrupt controller included in the XBurst Processor, explains its modes of operation, and defines its registers. The interrupt controller controls the interrupt sources available to the processor and contains the location of the interrupt source to allow software to determine source of all interrupts. It also determines whether the interrupts cause an IRQ to occur and masks the interrupts.

Features:

- Total 64 interrupt sources
- Each interrupt source can be independently enabled
- Priority mechanism to indicate highest priority interrupt
- All the registers are accessed by CPU
- Unmasked interrupts can wake up the chip in sleep mode

13.2 Register Description

Table 13-1 lists the registers of Interrupt Controller. All of these registers are 32bit, and each bit of the register represents or controls one interrupt source that list in Table 13-1.

All INTC register 32bit access address is physical address.

Table 13-1 INTC Register

Name	Description	RW	Reset Value	Address	Access Size
ICSR0	Interrupt controller Source Register	R	0x00000000	0x10001000	32
ICMR0	Interrupt controller Mask Register	RW	0xFFFFFFFF	0x10001004	32
ICMSR0	Interrupt controller Mask Set Register	W	0x????????	0x10001008	32
ICMCR0	Interrupt controller Mask Clear Register	W	0x????????	0x1000100C	32
ICPR0	Interrupt controller Pending Register	R	0x00000000	0x10001010	32
ICSR1	Interrupt controller Source Register	R	0x00000000	0x10001020	32
ICMR1	Interrupt controller Mask Register	RW	0xFFFFFFFF	0x10001024	32
ICMSR1	Interrupt controller Mask Set Register	W	0x????????	0x10001028	32
ICMCR1	Interrupt controller Mask Clear Register	W	0x????????	0x1000102C	32
ICPR1	Interrupt controller Pending Register	R	0x00000000	0x10001030	32

13.2.1 Interrupt Controller Source Register (ICSR0)

This register contains all the interrupts' status. A "1" indicates that the corresponding interrupt is pending . A "0" indicates that the interrupt is not pending now. The register is read only.

ICSR0 0x10001000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	LCD	CIM	IPU	GPS	TCU0	TCU1	TCU2	DMA0	DMA1	I2C2	OTG	UHC	ETH	SADC	GPIO0	GPIO1	GPIO2	GPIO3	GPIO4	GPIO5	KBC	BDMA	TSSI	SSI0	SSI1	GPU	UART0	UART1	UART2	UART3	I2C0	I2C1
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits Of ICSR0	Description
0	The corresponding interrupt source is not pending.
1	The corresponding interrupt source is pending.

13.2.2 Interrupt Controller Source Register (ICSR1)

This register contains all the interrupts' status. A "1" indicates that the corresponding interrupt is pending . A "0" indicates that the interrupt is not pending now. The register is read only.

ICSR1 **0x10001020**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																CPM	Reserved	I2S2CH	HARB2	AOSD	Reserved	PCM1	PCM0	BCH	SCC	MSC0	MSC1	MSC2	AIC	OWI	RTC
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits Of ICSR1	Description
0	The corresponding interrupt source is not pending.
1	The corresponding interrupt source is pending.

13.2.3 Interrupt Controller Mask Register (ICMR0)

This register is used to mask the interrupt input sources and defines which active sources are allowed to generate interrupt requests to the processor. Its value can be changed either by writing ICMSR and ICMCR or by writing itself. The masked interrupts are invisible to the processor.

ICMR0 **0x10001004**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	LCD	CIM	IPU	GPS	TCU0	TCU1	TCU2	DMA0	DMA1	I2C2	OTG	UHC	ETH	SADC	GPIO0	GPIO1	GPIO2	GPIO3	GPIO4	GPIO5	KBC	BDMA	TSSI	SSI0	SSI1	GPU	UART0	UART1	UART2	UART3	I2C0	I2C1
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits Of ICMR0	Description
0	The corresponding interrupt is not masked.
1	The corresponding interrupt is masked.

13.2.4 Interrupt Controller Mask Register (ICMR1)

This register is used to mask the interrupt input sources and defines which active sources are allowed to generate interrupt requests to the processor. Its value can be changed either by writing ICMSR and ICMCR or by writing itself. The masked interrupts are invisible to the processor.

ICMR1 **0x10001024**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																CPM	Reserved	I2S2CH	HARB2	AOSD	Reserved	PCM1	PCM0	BCH	SCC	MSC0	MSC1	MSC2	AIC	OWI	RTC
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits Of ICMR1	Description
0	The corresponding interrupt is not masked.
1	The corresponding interrupt is masked.

13.2.5 Interrupt Controller Mask Set Register (ICMSR0)

This register is used to set bits in the interrupt mask register. This register is write only.

ICMSR0 **0x10001008**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	LCD	CIM	IPU	GPS	TCU0	TCU1	TCU2	DMA0	DMA1	I2C2	OTG	UHC	ETH	SADC	GPIO0	GPIO1	GPIO2	GPIO3	GPIO4	GPIO5	KBC	BDMA	TSSI	SSI0	SSI1	GPU	UART0	UART1	UART2	UART3	I2C0	I2C1
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits Of ICMSR0	Description
0	Ignore.
1	Will set the corresponding interrupt mask bit.

13.2.6 Interrupt Controller Mask Set Register (ICMSR1)

This register is used to set bits in the interrupt mask register. This register is write only.

ICMSR1 **0x10001028**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																CPM	Reserved	I2S2CH	HARB2	AOSD	Reserved	PCM1	PCM0	BCH	SCC	MSC0	MSC1	MSC2	AIC	OWI	RTC
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits Of ICMSR1	Description
0	Ignore.
1	Will set the corresponding interrupt mask bit.

13.2.7 Interrupt Controller Mask Clear Register (ICMCR0)

This register is used to clear bits in the interrupt mask register. This register is write only.

ICMCR0 **0x1000100C**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	LCD	CIM	IPU	GPS	TCU0	TCU1	TCU2	DMA0	DMA1	I2C2	OTG	UHC	ETH	SADC	GPIO0	GPIO1	GPIO2	GPIO3	GPIO4	GPIO5	KBC	BDMA	TSSI	SSI0	SSI1	GPU	UART0	UART1	UART2	UART3	I2C0	I2C1
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits Of ICMCR0	Description
0	Ignore.
1	Will clear the corresponding interrupt mask bit.

13.2.8 Interrupt Controller Mask Clear Register (ICMCR1)

This register is used to clear bits in the interrupt mask register. This register is write only.

ICMCR1 **0x1000102C**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																CPM	Reserved	I2S2CH	HARB2	AOSD	Reserved	PCM1	PCM0	BCH	SCC	MSC0	MSC1	MSC2	AIC	OWI	RTC
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits Of ICMCR1	Description
0	Ignore.
1	Will clear the corresponding interrupt mask bit.

13.2.9 Interrupt Controller Pending Register (ICPR0)

This register contains the status of the interrupt sources after masking. This register is read only.

ICPR0 **0x10001010**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	LCD	CIM	IPU	GPS	TCU0	TCU1	TCU2	DMA0	DMA1	I2C2	OTG	UHC	ETH	SADC	GPIO0	GPIO1	GPIO2	GPIO3	GPIO4	GPIO5	KBC	BDMA	TSSI	SSI0	SSI1	GPU	UART0	UART1	UART2	UART3	I2C0	I2C1
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits Of ICPR0	Description
0	The corresponding interrupt is not active or is masked.
1	The corresponding interrupt is active and is not masked to the processor.

NOTES:

1 Reserved bits in ICMR0, ICMSR0 and ICMCR0 are normal bits to be written into and read out.

2 Reserved bits in ICSR and ICPR are read-only and always 0.

13.2.10 Interrupt Controller Pending Register (ICPR1)

This register contains the status of the interrupt sources after masking. This register is read only.

ICPR1 0x10001030

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																CPM	Reserved	I2S2CH	HARB2	AOSD	Reserved	PCM1	PCM0	BCH	SCC	MSC0	MSC1	MSC2	AIC	OWI	RTC
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits Of ICPR1	Description
0	The corresponding interrupt is not active or is masked.
1	The corresponding interrupt is active and is not masked to the processor.

NOTES:

1 Reserved bits in ICMR1, ICMSR1 and ICMCR1 are normal bits to be written into and read out.

2 Reserved bits in ICSR1 and ICPR1 are read-only and always 0.

13.3 Software Considerations

The interrupt controller is reflecting the status of interrupts sources in the peripheral .

Software should perform the task - determine the interrupt source from in ICPRx. In this chip, pending interrupts have two levels in structure. Interrupting module in the system that contains more than one interrupt sources need software to determine how to service it by reading interrupt status registers within it.

In the interrupt handler, the serviced interrupt source needs to be cleared in the interrupting device. In order to make certain the cleared source request status has been reflected at the corresponding ICPRx bit, software should wait enough time before exiting interrupt state.

The procedure is described following:

1 Interrupt generated.
2 CPU query interrupt sources, saves the current environment and then goes to interrupt common service routine.
3 Get ICPRx.
4 Find the highest priority interrupt and vector it. (The software decides which one has the highest priority)
5 Mask the chosen interrupt by writing the register ICMSRx.
6 Enable the system interrupt to allow the interrupt nesting. (software decided)
7 Execute the interrupt handler and unmask it by writing the register ICMCRx when exit the handler.
8 CPU restores the saved environment and exits the interrupt state.

14 Timer/Counter Unit

14.1 Overview

The TCU (Timer/Counter with PWM output) contains 8 channels of 16-bit programmable timers (timers 0 to 5). They can be used as Timer or PWM.

TCU has the following features:

- There are two modes of TCU for the eight channels
  - TCU1: Channel 0, 3,4, 5, 6,and 7
  - TCU2: Channel 1,2
- Six independent channels, each consisting of
  - Counter
  - Data register (FULL and HALF)
  - Control register
- Independent clock for each counter, selectable by software
  - PCLK, EXTAL and RTCCLK can be used as the clock for counter
  - The division ratio of the clock can be set to 1, 4, 16, 64, 256 and 1024 by software
- FULL interrupt and HALF interrupt can be generated for each channel using the compare data registers
  - Timer 0-7 can be used as PWM (Set the initial signal level)
  - Timer 0,3-7 can be used as a counter to count external signal (like trackball)
  - Timer 5 has separated interrupt
  - Timer 0-4 and timer 6-7 has one interrupt in common
  - OST uses interrupt 0, Timer 5 uses interrupt 1, and Timer 1-4/ 6-7 uses interrupt 2
- The difference between TCU1 and TCU2
  - TCU1: It cannot work in sleep mode, but operated easily
  - TCU2: It can work in sleep mode, but operated more complicated than TCU1

14.2 Pin Description

Table 14-1 PWM Pins Description

Name	I/O	Description
PWM [7:0]	Output	PWM channel output signals.

14.3 Register Description

In this section, we will describe the registers in timer. Following table lists all the registers definition. All timer register's 32bit address is physical address. And detailed function of each register will be described below.

Name	Description	RW	Reset Value	Address	Access Size
TSTR	Timer Status Register	R	0x00000000	0x100020F0	32
TSTSR	Timer Status Set Register	W	0x????????	0x100020F4	32
TSTCR	Timer Status Clear Register	W	0x????????	0x100020F8	32
TSR	Timer STOP Register	R	0x00000000	0x1000201C	32
TSSR	Timer STOP Set Register	W	0x00000000	0x1000202C	32
TSCR	Timer STOP Clear Register	W	0x0000	0x1000203C	32
TER	Timer Counter Enable Register	R	0x0000	0x10002010	16
TESR	Timer Counter Enable Set Register	W	0x????	0x10002014	16
TECR	Timer Counter Enable Clear Register	W	0x????	0x10002018	16
TFR	Timer Flag Register	R	0x003F003F	0x10002020	32
TFSR	Timer Flag Set Register	W	0x????????	0x10002024	32
TFCR	Timer Flag Clear Register	W	0x????????	0x10002028	32
TMR	Timer Mask Register	R	0x00000000	0x10002030	32
TMSR	Timer Mask Set Register	W	0x????????	0x10002034	32
TMCR	Timer Mask Clear Register	W	0x????????	0x10002038	32
TDFR0	Timer Data FULL Register 0	RW	0x????	0x10002040	16
TDHR0	Timer Data HALF Register 0	RW	0x????	0x10002044	16
TCNT0	Timer Counter 0	RW	0x????	0x10002048	16
TCSR0	Timer Control Register 0	RW	0x0000	0x1000204C	16
TDFR1	Timer Data FULL Register 1	RW	0x????	0x10002050	16
TDHR1	Timer Data HALF Register 1	RW	0x????	0x10002054	16
TCNT1	Timer Counter 1	RW	0x????	0x10002058	16
TCSR1	Timer Control Register 1	RW	0x0000	0x1000205C	16
TDFR2	Timer Data FULL Register 2	RW	0x????	0x10002060	16
TDHR2	Timer Data HALF Register 2	RW	0x????	0x10002064	16
TCNT2	Timer Counter 2	RW	0x????	0x10002068	16
TCSR2	Timer Control Register 2	RW	0x0000	0x1000206C	16
TDFR3	Timer Data FULL Register 3	RW	0x????	0x10002070	16
TDHR3	Timer Data HALF Register 3	RW	0x????	0x10002074	16
TCNT3	Timer Counter 3	RW	0x????	0x10002078	16
TCSR3	Timer Control Register 3	RW	0x0000	0x1000207C	16
TDFR4	Timer Data FULL Register 4	RW	0x????	0x10002080	16

TDHR4	Timer Data HALF Register 4	RW	0x????	0x10002084	16
TCNT4	Timer Counter 4	RW	0x????	0x10002088	16
TCSR4	Timer Control Register 4	RW	0x0000	0x1000208C	16
TDFR5	Timer Data FULL Register 5	RW	0x????	0x10002090	16
TDHR5	Timer Data HALF Register 5	RW	0x????	0x10002094	16
TCNT5	Timer Counter 5	RW	0x????	0x10002098	16
TCSR5	Timer Control Register 5	RW	0x0000	0x1000209C	16
TDFR6	Timer Data FULL Register 6	RW	0x????	0x100020A0	16
TDHR6	Timer Data HALF Register 6	RW	0x????	0x100020A4	16
TCNT6	Timer Counter 6	RW	0x????	0x100020A8	16
TCSR6	Timer Control Register 6	RW	0x0000	0x100020AC	16
TDFR7	Timer Data FULL Register 7	RW	0x????	0x100020B0	16
TDHR7	Timer Data HALF Register 7	RW	0x????	0x100020B4	16
TCNT7	Timer Counter 7	RW	0x????	0x100020B8	16
TCSR7	Timer Control Register 7	RW	0x0000	0x100020BC	16
TCUMOD0	Timer control mode Register 0	RW	0x????	0x10002100	16
TCUMOD3	Timer control mode Register 3	RW	0x????	0x10002110	16
TCUMOD4	Timer control mode Register 4	RW	0x????	0x10002120	16
TCUMOD5	Timer control mode Register 3	RW	0x????	0x10002130	16
TFWD0	Timer fifo write data Register 0	RW	0x????????	0x10002104	32
TFWD3	Timer fifo write data Register 3	RW	0x????????	0x10002114	32
TFWD4	Timer fifo write data Register 4	RW	0x????????	0x10002124	32
TFWD5	Timer fifo write data Register 4	RW	0x????????	0x10002134	32
TFIFOSR0	Timer fifo state Register	R	0x??	0x10002108	6
TFIFOSR3	Timer fifo state Register	R	0x??	0x10002118	6
TFIFOSR4	Timer fifo state Register	R	0x??	0x10002128	6
TFIFOSR5	Timer fifo state Register	R	0x??	0x10002138	6

14.3.1 Timer Control Register (TCSR)

The TCSR is a 16-bit read/write register. It contains the control bits for each channel. It is initialized to 0x00 by any reset.

TCSR0, TCSR1, TCSR2, TCSR3, TCSR4, TCSR5 TCSR6, TCSR7

0x1000204C, 0x1000205C, 0x1000206C, 0x1000207C, 0x1000208C, 0x1000209C, 0x100020AC, 0x100020BC

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved				BYPASS	CLRZ	SD	INITL	PWM_EN	PWM_IN_EN	PRESCALE			EXT_EN	RTC_EN	PCK_EN
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
15:11	Reserved	Writing has no effect, read as zero.	R
11	BYPASS	PWM bypass mode. 1: If PCK_EN = 1, this channel output PIXCLK; If RTC_EN = 1, this channel output RTCCLK; If EXT_EN = 1, this channel output EXTAL; Only one of those XXX_EN is permit available during one time. 0: This BYPASS function disable. *Only when you want to let this PWM channel output some special clock (PIXCLK, RTCLK and EXTAL clock), you can set this register to 1. Otherwise keep it to 0. When you want to use BYPASS function, not forget offer clock supplies of relate channel (relate to register TSR, TSSR, TSCR).	RW
10	CLRZ	Clear counter to 0. It is only used in TCU2 mode. Writing 1 to this bit will clear the counter to 0. When the counter is finished setting to 0, it will be cleared by hardware. Writing 0 to this bit will be ignored.	RW
9	SD	Shut Down (SD) the PWM output. It is only used in TCU1 mode. 0: Graceful shutdown 1: Abrupt shutdown Graceful shutdown: The output level for PWM output will keep the level after the comparison match of FULL. Abrupt shutdown: The output level for PWM output will keep the level.	RW
8	INITL	Selects an initial output level for PWM output. 0: Low 1: High	RW
7	PWM_EN	PWM output pin control bit. 0: PWM pin output disable, and the PWM pin will be set to the initial level according to INITL 1: PWM pin output enable	RW
6	PWM_IN_EN	PWM input mode enable. Set to 1 to enable this function. In this function, PWM pin need to set as input in GPIO to receive external signal, EXT_EN, RTC_EN, PCK_EN need to set 0. And TCNT became a counter to count this signal's both edges. (This bit in TCSR1, 2 are reserved).	RW
5:3	PRESCALE	These bits select the TCNT count clock frequency. Don't change this field when the channel is running.	RW

Bit 2	Bit1	Bit 0	Description
0	0	0	Internal clock: CLK/1
0	0	1	Internal clock: CLK/4
0	1	0	Internal clock: CLK/16

		0	1	1	Internal clock: CLK/64	
		1	0	0	Internal clock: CLK/256	
		1	0	1	Internal clock: CLK/1024	
		110~111			Reserved	
2	EXT_EN	Select EXTAL as the timer clock input. 0: Disable 1: Enable				RW
1	RTC_EN	Select RTCCLK as the timer clock input. 0: Disable 1: Enable				RW
0	PCK_EN	Select PCLK as the timer clock input. 0: Disable 1: Enable				RW

NOTE: The input clock of timer and the PCLK should keep to the rules as follows:

Input clock of timer: IN_CLK	**Clock generated from the frequency divider (PRESCALE): DIV_CLK**
PCK_EN == 0, RTC_EN == 1 and EXT_EN == 0 (IN_CLK = RTCCLK)	$f_{DIV\_CLK} < ½ f_{PCLK}$
PCK_EN == 0, RTC_EN == 0 and EXT_EN == 1 (IN_CLK = EXTAL)	$f_{DIV\_CLK} < ½ f_{PCLK}$
PCK_EN == 1, RTC_EN == 0 and EXT_EN == 0 (IN_CLK = PCLK)	ANY

14.3.2 Timer Data FULL Register (TDFR)

The comparison data FULL registers TDFR is used to store the data to be compared with the content of the up-counter TCNT. This register can be directly read and written. (Default: indeterminate) But it is not suggested changing when counter is working in TCU2 mode.

TDFR0, TDFR1, TDFR2, TDFR3, TDFR4, TDFR5, TDFR6, TDFR7 — **0x10002040, 0x10002050, 0x10002060, 0x10002070, 0x10002080, 0x10002090, 0x100020A0, 0x100020B0**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TDFR															
RST	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?

14.3.3 Timer Data HALF Register (TDHR)

The comparison data HALF registers TDHR is used to store the data to be compared with the content of the up-counter TCNT. This register can be directly read and written. (Default: indeterminate) But it is not suggested changing when counter is working in TCU2 mode.

TDHR0, TDHR1, TDHR2, TDHR3, TDHR4, TDHR5, TDHR6, TDHR7

0x10002044, 0x10002054, 0x10002064, 0x10002074, 0x10002084, 0x10002094, 0x100020A4, 0x100020B4

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TDHR															
RST	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?

14.3.4 Timer Counter (TCNT)

TCNT is a 16-bit read/write register. The up-counter TCNT can be reset to 0 by software and counts up using the prescaler output clock. When TCNT count up to equal to TDFR, it will reset to 0 and continue to count up.

TCU1: The counter data can be read out at any time. The data can be written at any time. This makes it possible to change the interrupt and/or clock output cycles temporarily. (Default: indeterminate)

TCU2: The counter data can be read out at any time, but you should read TSTR.REALn to check whether the data is real data or not. The data can only be written before counter is started, and the counter clock is pclk. But it can be cleared to 0 by setting TCSR.CLRZ to 1, and if the counter is really cleared, TCSR.CLRZ will be set to 0 by hardware.

TCNT0, TCNT1, TCNT2, TCNT3, TCNT4, TCNT5, TCNT6, TCNT7

0x10002048, 0x10002058, 0x10002068, 0x10002078, 0x10002088, 0x10002098, 0x100020A8, 0x100020B8

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TCNT															
RST	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?

14.3.5 Timer Counter Enable Register (TER)

The TER is a 16-bit read-only register. It contains the counter enable control bits for each channel. It is initialized to 0x0000 by any reset. It can only be set by register TESR and TECR. Since the timer enable control bits are located in the same addresses, two or more timers can be started at the same time.

TER 0x10002010

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	OSTEN	Reserved							TCEN 7	TCEN 6	TCEN 5	TCEN 4	TCEN 3	TCEN 2	TCEN 1	TCEN 0
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
15	OSTEN	Enable the counter in OST. 0: Stop counting up 1: Begin counting up	
14:8	Reserved	Writing has no effect, read as zero.	R
7	TCEN 7	Enable the counter in timer 7. 0: Stop counting up 1: Begin counting up	R
6	TCEN 6	Enable the counter in timer 6. 0: Stop counting up 1: Begin counting up	R
5	TCEN 5	Enable the counter in timer 5. 0: Stop counting up 1: Begin counting up	R
4	TCEN 4	Enable the counter in timer 4. 0: Stop counting up 1: Begin counting up	R
3	TCEN 3	Enable the counter in timer 3. 0: Stop counting up 1: Begin counting up	R
2	TCEN 2	Enable the counter in timer 2. 0: Stop counting up 1: Begin counting up	R
1	TCEN 1	Enable the counter in timer 1. 1: Begin counting up 0: Stop counting up	R
0	TCEN 0	Enable the counter in timer 0. 0: Stop counting up 1: Begin counting up	R

14.3.6 Timer Counter Enable Set Register (TESR)

The TCCSR is a 32-bit write-only register. It contains the counter enable set bits for each channel. Since the timer enable control set bits are located in the same addresses, two or more timers can be started at the same time.

TESR 0x10002014

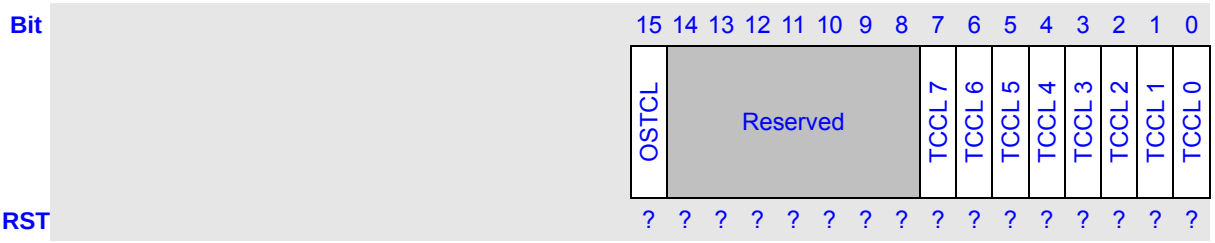
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	OSTST	Reserved							TCST 7	TCST 6	TCST 5	TCST 4	TCST 3	TCST 2	TCST 1	TCST 0
RST	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?

Bits	Name	Description	RW
15	OSTST	Set OSTEN bit of TER. 0: Ignore 1: Set OSTEN bit to 1	W
14:8	Reserved	Writing has no effect, read as zero.	R
7	TCST 7	Set TCEN 7 bit of TER. 0: Ignore 1: Set TCEN 5 bit to 1	W
6	TCST 6	Set TCEN 6 bit of TER. 0: Ignore 1: Set TCEN 5 bit to 1	W
5	TCST 5	Set TCEN 5 bit of TER. 0: Ignore 1: Set TCEN 5 bit to 1	W
4	TCST 4	Set TCEN 4 bit of TER. 1: Set TCEN 4 bit to 1 0: Ignore	W
3	TCST 3	Set TCEN 3 bit of TER. 0: Ignore 1: Set TCEN 3 bit to 1	W
2	TCST 2	Set TCEN 2 bit of TER. 1: Set TCEN 2 bit to 1 0: Ignore	W
1	TCST 1	Set TCEN 1 bit of TER. 0: Ignore 1: Set TCEN 1 bit to 1	W
0	TCST 0	Set TCEN 0 bit of TER. 0: Ignore 1: Set TCEN 0 bit to 1	W

14.3.7 Timer Counter Enable Clear Register (TECR)

The TECR is a 32-bit write-only register. It contains the counter enable clear bits for each channel. Since the timer enable clear bits are located in the same addresses, two or more timers can be stop at the same time.

TECR **0x10002018**



Bits	Name	Description	RW
15	OSTCL	Set OSTEN bit of TER. 0: Ignore 1: Set OSTEN 5 bit to 0	W
14:8	Reserved	Writing has no effect, read as zero.	R
7	TCCL 7	Set TCEN 7 bit of TER. 0: Ignore 1: Set TCEN 6 bit to 0	W
6	TCCL 6	Set TCEN 7 bit of TER. 0: Ignore 1: Set TCEN 6 bit to 0	W
5	TCCL 5	Set TCEN 5 bit of TER. 0: Ignore 1: Set TCEN 5 bit to 0	W
4	TCCL 4	Set TCEN 4 bit of TER. 1: Set TCEN 4 bit to 0 0: Ignore	W
3	TCCL 3	Set TCEN 3 bit of TER. 0: Ignore 1: Set TCEN 3 bit to 0	W
2	TCCL 2	Set TCEN 2 bit of TER. 1: Set TCEN 2 bit to 0 0: Ignore	W
1	TCCL 1	Set TCEN 1 bit of TER. 0: Ignore 1: Set TCEN 1 bit to 0	W
0	TCCL 0	Set TCEN 0 bit of TER. 0: Ignore 1: Set TCEN 0 bit to 0	W

14.3.8 Timer Flag Register (TFR)

The TFR is a 32-bit read-only register. It contains the comparison match flag bits for all the channels. It can also be set by register TFSR and TFCR. It is initialized to 0x00000000 by any reset.

TFR 0x10002020

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved								HFLAG 7	HFLAG 6	HFLAG 5	HFLAG 4	HFLAG 3	HFLAG 2	HFLAG 1	HFLAG 0	OSTFLAG	Reserved			FIFOFLAG5	FIFOFLAG4	FIFOFLAG3	FIFOFLAG0	FFLAG 7	FFLAG 6	FFLAG 5	FFLAG 4	FFLAG 3	FFLAG 2	FFLAG 1	FFLAG 0
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
31:24	Reserved	Writing has no effect, read as zero.	R
23:16	HFLAG 7~0	HALF comparison match flag. (TCNT = TDHR) 0: Comparison not match 1: Comparison match	R
15	OSTFLAG	OST comparison match flag. (OSTCNT = OSTDR) 0: Comparison not match 1: Comparison match	R
14:12	Reserved	Writing has no effect, read as zero.	R
11:8	FIFOFLAG 3~0	FIFO comparison match flag. (TCNT = TFWD) 0: Comparison not match 1: Comparison match	R
7:0	FFLAG 7~0	FULL comparison match flag. (TCNT = TDFR) 0: Comparison not match 1: Comparison match	R

14.3.9 Timer Flag Set Register (TFSR)

The TFSR is a 32-bit write-only register. It contains the comparison match flag set bits for all the channels.

TFSR 0x10002024

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved								HFST 7	HFST 6	HFST 5	HFST 4	HFST 3	HFST 2	HFST 1	HFST 0	OSTFST	Reserved			FIFOST5	FIFOST4	FIFOST3	FIFOST0	FFST 7	FFST 6	FFST 5	FFST 4	FFST 3	FFST 2	FFST 1	FFST 0
RST	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?

Bits	Name	Description	RW
31:24	Reserved	Writing has no effect, read as zero.	R
23:16	HFST 7~0	Set HFLAG n bit of TFR. 0: Ignore 1: Set HFLAG n bit to 1	W
15	OSTFST	Set OSTFLAG n bit of TFR. 0: Ignore 1: Set OSTFLAG n bit to 1	W
14:12	Reserved	Writing has no effect, read as zero.	R
11:8	FIFOST 3~0	Set FIFOFLAG n bit of TFR. 0: Ignore; 1: Set FIFOFLAG n bit to 1.	W
7:0	FFST 7~0	Set FFLAG n bit of TFR. 0: Ignore 1: Set FFLAG n bit to 1	W

14.3.10 Timer Flag Clear Register (TFCR)

The TFCR is a 32-bit write-only register. It contains the comparison match flag clear bits for all the channels.

TFCR 0x10002028

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved								HFCL 7	HFCL 6	HFCL 5	HFCL 4	HFCL 3	HFCL 2	HFCL 1	HFCL 0	OSTFCL	Reserved			FIFOCL5	FIFOCL4	FIFOCL3	FIFOCL0	FFCL 7	FFCL 6	FFCL 5	FFCL 4	FFCL 3	FFCL 2	FFCL 1	FFCL 0
RST	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?

Bits	Name	Description	RW
31:24	Reserved	Writing has no effect, read as zero.	R
23:16	HFCL 7~0	Set HFLAG n bit of TFR. 0: Ignore 1: Set FFLAG n bit to 0	W
15	OSTFCL	Set OSTFLAG n bit of TFR. 0: Ignore 1: Set OSTFLAG n bit to 0	W
14:12	Reserved	Writing has no effect, read as zero.	R
11:8	FIFOCL 3~0	Set FIFOFLAG n bit of TFR. 0: Ignore 1: Set FIFOFLAG n bit to 0	W
7:0	FFCL 7~0	Set FFLAG n bit of TFR. 0: Ignore	W

		1: Set FFLAG n bit to 0	

14.3.11 Timer Mast Register (TMR)

The TMR is a 32-bit read-only register. It contains the comparison match flag bits for all the channels. It is initialized to 0x003F003F by any reset. It can only be set by register TMSR and TMCR.

TMR **0x10002030**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved								HMASK 7	HMASK 6	HMASK 5	HMASK 4	HMASK 3	HMASK 2	HMASK 1	HMASK 0	OSTMASK	Reserved			FIFOMASK5	FIFOMASK4	FIFOMASK3	FIFOMASK0	FMASK 7	FMASK 6	FMASK 5	FMASK 4	FMASK 3	FMASK 2	FMASK 1	FMASK 0
RST	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1

Bits	Name	Description	RW
31:24	Reserved	Writing has no effect, read as zero.	R
23:16	HMASK 7~0	HALF comparison match interrupt mask. 0: Comparison match interrupt not mask 1: Comparison match interrupt mask	R
15	OSTMASK	OST comparison match interrupt mask. 0: Comparison match interrupt not mask 1: Comparison match interrupt mask	R
14:12	Reserved	Writing has no effect, read as zero.	R
11:8	FIFOMASK 3~0	FIFO comparison match interrupt mask. 0: Comparison match interrupt not mask 1: Comparison match interrupt mask	R
7:0	FMASK 7~0	FULL comparison match interrupt mask. 0: Comparison match interrupt not mask 1: Comparison match interrupt mask	R

14.3.12 Timer Mask Set Register (TMSR)

The TMSR is a 32-bit write-only register. It contains the comparison match flag set bits for all the channels.

TMSR **0x10002034**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved								HMST 7	HMST 6	HMST 5	HMST 4	HMST 3	HMST 2	HMST 1	HMST 0	OSTMST	Reserved			FIFOMST5	FIFOMST4	FIFOMST3	FIFOMST0	FMST 7	FMST 6	FMST 5	FMST 4	FMST 3	FMST 2	FMST 1	FMST 0
RST	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?

Bits	Name	Description	RW
31:24	Reserved	Writing has no effect, read as zero.	R
23:16	HMST 7~0	Set HMASK n bit of TMR. 0: Ignore 1: Set HMASK n bit to 1	W
15	OSTMST	Set OSTMASK n bit of TMR. 0: Ignore 1: Set OSTMASK n bit to 1	W
14:12	Reserved	Writing has no effect, read as zero.	R
11:8	FIFOMST 3~0	Set FIFOMST n bit of TMR. 0: Ignore 1: Set FMASK n bit to 1	W
7:0	FMST 7~0	Set FMASK n bit of TMR. 0: Ignore 1: Set FMASK n bit to 1	W

14.3.13 Timer Mask Clear Register (TMCR)

The TMCR is a 32-bit write-only register. It contains the comparison match flag clear bits for all the channels.

TMCR 0x10002038

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved								HMCL 7	HMCL 6	HMCL 5	HMCL 4	HMCL 3	HMCL 2	HMCL 1	HMCL 0	OSTMCL	Reserved			FIFOMCL5	FIFOMCL4	FIFOMCL3	FIFOMCL0	FMCL 7	FMCL 6	FMCL 5	FMCL 4	FMCL 3	FMCL 2	FMCL 1	FMCL 0
RST	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?

Bits	Name	Description	RW
31:22	Reserved	Writing has no effect, read as zero.	R
23:16	HMCL 7~0	Set HMASK n bit of TMR. 0: Ignore 1: Set HMASK n bit to 0	W
15	OSTMCL	Set OSTMASK n bit of TMR. 0: Ignore 1: Set OSTMASK n bit to 0	W
14:12	Reserved	Writing has no effect, read as zero.	R
11:8	FIFOMCL 3~0	Set FIFOMCL n bit of TMR. 0: Ignore 1: Set FIFOMCL n bit to 0	W
7:0	FMCL 7~0	Set FMASK n bit of TMR.	W

		0: Ignore 1: Set FMASK n bit to 0	

14.3.14 Timer Stop Register (TSR)

The TSR is a 32-bit read-only register. It contains the timer stop control bits for each channel, WDT and OST. It is initialized to 0x00000000 by any reset. It can only be set by register TSSR and TSCR. If set, clock supplies to timer n / WDT / OST is stopped, and registers of the timer / WDT / OST cannot be accessed also.

TSR 0x1000201C

Bit	31:17	16	15	14:8	7	6	5	4	3	2	1	0
	Reserved	WDTS	OSTS	Reserved	STOP 7	STOP 6	STOP 5	STOP 4	STOP 3	STOP 2	STOP 1	STOP 0
RST	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
31:17	Reserved	Writing has no effect, read as zero.	R
16	WDTS	0: The clock supplies to WDT is supplied 1: The clock supplies to WDT is stopped	R
15	OSTS	0: The clock supplies to OST is supplied 1: The clock supplies to OST is stopped	R
14:8	Reserved	Writing has no effect, read as zero.	R
7	STOP 7	0: The clock supplies to timer 7 is supplied 1: The clock supplies to timer 7 is stopped	R
6	STOP 6	0: The clock supplies to timer 6 is supplied 1: The clock supplies to timer 6 is stopped	R
5	STOP 5	0: The clock supplies to timer 5 is supplied 1: The clock supplies to timer 5 is stopped	R
4	STOP 4	0: The clock supplies to timer 4 is supplied 1: The clock supplies to timer 4 is stopped	R
3	STOP 3	0: The clock supplies to timer 3 is supplied 1: The clock supplies to timer 3 is stopped	R
2	STOP 2	0: The clock supplies to timer 2 is supplied 1: The clock supplies to timer 2 is stopped	R
1	STOP 1	0: The clock supplies to timer 1 is supplied 1: The clock supplies to timer 1 is stopped	R
0	STOP 0	0: The clock supplies to timer 0 is supplied 1: The clock supplies to timer 0 is stopped	R

14.3.15 Timer Stop Set Register (TSSR)

The TCSR is an 32-bit write-only register. It contains the timer stop set bits for each channel, WDT and OST. Since the timer stop control set bits are located in the same addresses, two or more timers can be started at the same time.

TSSR 0x1000202C

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved															WDTSS	OSTSS	Reserved							STPS 7	STPS 6	STPS 5	STPS 4	STPS 3	STPS 2	STPS 1	STPS 0
RST	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?

Bits	Name	Description	RW
31:17	Reserved	Writing has no effect, read as zero.	R
16	WDTSS	Set WDTS bit of TSR. 0: Ignore 1: Set WDTS bit to 1	W
15	OSTSS	Set OSTS bit of TSR. 0: Ignore 1: Set OSTS bit to 1	W
14:8	Reserved	Writing has no effect, read as zero.	R
7	STPS 7	Set STOP 7 bit of TSR. 0: Ignore 1: Set STOP 7 bit to 1	W
6	STPS 6	Set STOP 6 bit of TSR. 0: Ignore 1: Set STOP 6 bit to 1	W
5	STPS 5	Set STOP 5 bit of TSR. 0: Ignore 1: Set STOP 5 bit to 1	W
4	STPS 4	Set STOP 4 bit of TSR. 1: Set STOP 4 bit to 1 0: Ignore	W
3	STPS 3	Set STOP 3 bit of TSR. 0: Ignore 1: Set STOP 3 bit to 1	W
2	STPS 2	Set STOP 2 bit of TSR. 0: Ignore 1: Set STOP 2 bit to 1	W
1	STPS 1	Set STOP 1 bit of SR. 0: Ignore	W

		1: Set STOP 1 bit to 1	
0	STPS 0	Set STOP 0 bit of TSR. 0: Ignore 1: Set STOP 0 bit to 1	W

14.3.16 Timer Stop Clear Register (TSCR)

The TSCR is an 32-bit write-only register. It contains the timer stop clear bits for each channel, WDT and OST. Since the timer stop clear bits are located in the same addresses, two or more timers can be stop at the same time.

TSCR **0x1000203C**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved															WDTSC	OSTSC	Reserved							STPC 7	STPC 6	STPC 5	STPC 4	STPC 3	STPC 2	STPC 1	STPC 0
RST	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?

Bits	Name	Description	RW
31:17	Reserved	Writing has no effect, read as zero.	R
16	WDTSC	Set WDTS bit of TSR. 0: Ignore 1: Set WDTS bit to 0	W
15	OSTSC	Set OSTS bit of TSR. 0: Ignore 1: Set OSTS bit to 0	W
14:8	Reserved	Writing has no effect, read as zero.	R
7	STPC 7	Set STOP 7 bit of TSR. 0: Ignore 1: Set STOP 7 bit to 0	W
6	STPC 6	Set STOP 6 bit of TSR. 0: Ignore 1: Set STOP 6 bit to 0	W
5	STPC 5	Set STOP 5 bit of TSR. 0: Ignore 1: Set STOP 5 bit to 0	W
4	STPC 4	Set STOP 4 bit of TSR. 0: Ignore 1: Set STOP 4 bit to 0	W
3	STPC 3	Set STOP 3 bit of TSR. 0: Ignore 1: Set STOP 3 bit to 0	W

2	STPC 2	Set STOP 2 bit of TSR. 0: Ignore 1: Set STOP 2 bit to 0	W
1	STPC 1	Set STOP 1 bit of TSR. 0: Ignore 1: Set STOP 1 bit to 0	W
0	STPC 0	Set STOP 0 bit of TSR. 0: Ignore 1: Set STOP 0 bit to 0	W

14.3.17 Timer Status Register (TSTR)

The TSTR is a 32-bit read-only register. It contains the status of channel in TCU2 mode. The register can be written by setting register TSTSR and TSTCR.

TSR 0x100020F0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved													REAL2	REAL1	Reserved														BUSY2	BUSY1	Reserved
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
31:19	Reserved	Writing has no effect, read as zero.	R
18	REAL 2	0: The value read from counter 2 is a false value 1: The value read from counter 2 is a real value	R
17	REAL1	0: The value read from counter 1 is a false value 1: The value read from counter 1 is a real value	R
16:3	Reserved	Writing has no effect, read as zero.	R
2	BUSY 2	0: The counter 2 is ready now 1: The counter 2 is busy now	R
1	BUSY1	0: The counter 1 is ready now 1: The counter 1 is busy now	R
0	Reserved	Writing has no effect, read as zero.	R

14.3.18 er Status Set Register (TSTSR)

The TSTSR is a 32-bit write-only register. It contains the timer status set bits for each channel.

TSTSR **0x100020F4**

Bit	31:19	18	17	16:3	2	1	0
	Reserved	REALS 2	REALS 1	Reserved	BUSYS 2	BUSYS 1	Reserved
RST	0	0	0	0	0	0	0

Bits	Name	Description	RW
31:19	Reserved	Writing has no effect, read as zero.	R
18	REALS 2	Set REAL 2 bit of TSTR. 0: Ignore 1: Set REAL 2 bit to 1	R
17	REALS 1	Set REAL 1 bit of TSTR. 0: Ignore 1: Set REAL 1 bit to 1	R
16:3	Reserved	Writing has no effect, read as zero.	R
2	BUSYS 2	Set BUSY 2 bit of TSTR. 0: Ignore 1: Set BUSY 2 bit to 1	R
1	BUSYS 1	Set BUSY 1 bit of TSTR. 0: Ignore 1: Set BUSY 1 bit to 1	R
0	Reserved	Writing has no effect, read as zero.	R

14.3.19 Timer Status Clear Register (TSTCR)

The TSTCR is a 32-bit write-only register. It contains the timer status clear bits for each channel.

TSTCR **0x100020F8**

Bit	31:19	18	17	16:3	2	1	0
	Reserved	REALC 2	REALC 1	Reserved	BUSYC 2	BUSYC 1	Reserved
RST	0	0	0	0	0	0	0

Bits	Name	Description	RW
31:19	Reserved	Writing has no effect, read as zero.	R
18	REALC 2	Clear REAL 2 bit of TSTR. 0: Ignore	R

		1: Clear REAL 2 bit to 1	
17	REALC 1	Clear REAL 1 bit of TSTR. 0: Ignore 1: Clear REAL 1 bit to 1	R
16:3	Reserved	Writing has no effect, read as zero.	R
2	BUSYC 2	Clear BUSY 2 bit of TSTR. 0: Ignore 1: Clear BUSY 2 bit to 1	R
1	BUSYC 1	Clear BUSY 1 bit of TSTR. 0: Ignore 1: Clear BUSY 1 bit to 1	R
0	Reserved	Writing has no effect, read as zero.	R

14.3.20 Timer control mode Register (TCUMOD)

The TCUMOD is a 32-bit read write register. It contains the fifo control signal when TCU work in the TCU fifo mode , and only for the TCU0 、TCU3、TCU4、TCU5.

TCUMOD0 ,TCUMOD3, TCUMOD4,TCUMOD5 **0x10002100, 0x10002110, 0x10002120, 0x10002130**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																FIFOCYSN								FIOOCYS	FIFOSDN				FIFOSN	FIFOCL	FIFOMOD
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
31:16	Reserved	Writing has no effect, read as zero.	R
15:8	FIFOCYSN 7~0	Set the fifo cycle numbers. when the TCU work in FIFOMOD and the FIFOCYS set to 1 ,then FIFOCYSN is valid, else is Ignore.	RW
7	FIFOCYS 1	Set the TCU Work in FIFOCYS mode. Only valid when TCU FIFOMOD set. 0: disable work in FIFOCYS mode 1: enable work in FIFOCYS mode	RW
6:3	FIFOSDN 3~0	Set the fifo data numbers can read out of the fifo. when the TCU work in FIFOMOD and the FIFOSN set to 1 ,then FIFOSDN is valid, else is Ignore.	RW
2	FIFOSN 1	Set the TCU Work in FIFOSN. Only valid when TCU FIFOMOD set. 0: disable work in FIFOCYS mode 1: enable work in FIFOCYS mode	RW
1	FIFOCL 1	Clear the read write pointer set.	RW

		0: disable to clear the read write pointer 1: clear the read write pointer	
0	FIFOMOD 0	Set the TCU work in fifo mode. 0: TCU work in norm mode 1: TCU work in fifo mode	RW

14.3.21 Timer fifo write data (TFWD)

The comparison data FIFO registers TFWD is used to store the data to be compared with the content of the up-counter TCNT. This register can be directly read and written(Default: 0),but only can read the 16 bit data at once. And Only use in the fifo mode.

TFWD0 , TFWD3, TFWD4, TFWD5 — **0x10002104, 0x10002114, 0x10002124, 0x10002134**

Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
	TFWD_HIGN	TFWD_LOW
RST	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

14.3.22 Timer fifo state Register (TFIFOSR)

The TFIFOSR is a 32-bit write-only register. It contains the timer fifo status bits for TCU work in fifo mode.

TFIFOSR0 , TFIFOSR3, TFIFOSR4, TFIFOSR5 — **0x10002108, 0x10002118, 0x10002128, 0x10002138**

Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6	5 4 3 2	1	0
	Reserved	FIFO ENTRY	FIFOCYM	FIFONM
RST	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0	0	0

Bits	Name	Description	RW
31:6	Reserved	Writing has no effect, read as zero.	R
5:2	FIFO ENTRY 3~0	Use to trace the numbers of data.	R
1	FIFOCYM 1	Use to trace wether the fifo cycle is match. 0: not match 1: match	R
0	FIFONM 1	Use to trace when read, wether the fifo data is match. 0: not match 1: match	R

14.4 Operation

14.4.1 Basic Operation in TCU1 Mode

The value of TDFR should be bigger than TDHR, and the minimum settings are TDHR = 0 and TDFR = 1. In this case, the timer output clock cycle is the input clock × 1/2. If TDHR > TDFR, no comparison TFHR signal is generated.

Before the timer counter begin to count up, we need to do as follows:

If you want to use PWM you should set TCSR.PWM_EN to be 0 before you initial TCU.

1 Initial the configuration.
   a Writing TCSR.INITL to initialize PWM output level.
   b Writing TCSR.SD to setting the shutdown mode (Abrupt shutdown or Graceful shutdown).
   c Writing TCSR.PRESCALE to set TCNT count clock frequency.
   d Setting TCNT, TDHR and TDFR.

2 Enable the clock.
   a Writing TCSR.PWM_EN to set whether enable PWM or disable PWM.
   b Writing TCSR.EXT_EN, TCSR.RTC_EN or TCSR.PCK_EN to 1 to select the input clock and enable the input clock. Only one of TCSR.EXT_EN, TCSR.RTC_EN and TCSR.PCK_EN can be set to 1.

After initialize the register of timer, we should start the counter as follows:

3 Enable the counter.
   Setting the TESR.TCST bit to 1 to enable the TCNT.

NOTE: The input clock and PCLK should follow the rules advanced before.

14.4.2 Disable and Shutdown Operation in TCU1 Mode

1 Setting the TECR.TCCL bit to 1 to disable the TCNT.

14.4.3 Basic Operation in TCU2 Mode

The value of TDFR should be bigger than TDHR, and the minimum settings are TDHR = 0 and TDFR = 1. In this case, the timer output clock cycle is the input clock × 1/2. If TDHR > TDFR, no comparison TFHR signal is generated.

Initial state is that TCSR.PRESCALE=0, TCSR.PWM_EN=0 and TCENR=0.

1 Reset the TCU.
   a Writing TCSR.PCK_EN to 1 to select pclk as the input clock.
   b Set TCSR.CLRZ to 1 to clear TCNT or set TCNT to an initial value.
   c Writing TCSR.PCK_EN to 0 to close the input clock.

2 Initial the configuration.
   a Setting TDHR and TDFR.
   b Writing TCSR.INITL to initialize PWM output level (if used PWM).
   c Writing TCSR.PRESCALE to set TCNT count clock frequency.
   d Writing TCSR.EXT_EN, TCSR.RTC_EN or TCSR.PCK_EN to 1 to select the input clock and enable the input clock. Only one of TCSR.EXT_EN, TCSR.RTC_EN and TCSR.PCK_EN can be set to 1.
   e Writing TCSR.PWM_EN to set whether enable PWM or disable PWM.

After initialize the register of timer, we should start the counter as follows:

3 Setting the TESR.TCST bit to 1 to enable the TCNT.

NOTES:

1 You can clear the counter when counter is working.
   a Set TCSR.CLRZ to 1 to clear TCNT.
   b Wait till TSTR.BUSY = 0, that is the counter have been cleared.
2 You can enable PWM or disable PWM the counter when counter is working.
   a Set TCSR.PWM_EN to 1 to enable PWM.
   b Set TCSR.PWM_EN to 0 to disable PWM.

14.4.4 Disable and Shutdown Operation in TCU2 Mode

1 Writing TCSR.PWM_EN to 0 to disable PWM.
2 Setting the TECR.TCCL bit to 1 to disable the TCNT.
3 Wait till TSTR.BUSY = 0, that is the reset of counter is finished.

14.4.5 Read Counter in TCU2 Mode

If you want to read the data from register TCNT when the TCU is working, you can check TSTR.REAL whether it is good or not. It is suggested that:

1 If TSTR.REAL==1, the data read is available.
2 If TSTR.REAL==0, reread the counter till TSTR.REAL==1, the data read is available.
3 If TSTR.REAL is always 0, you can read some data, and lose some data that is quick different from the others. Then choose a data from them as the available data.

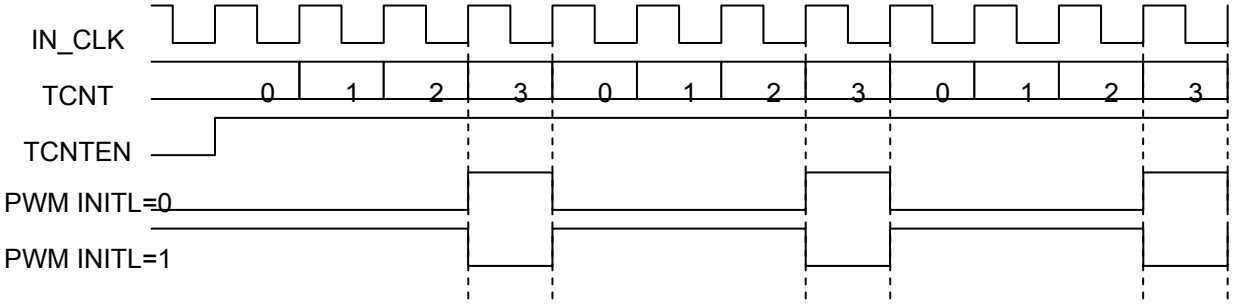
NOTES:

1 It suggested that (1), (2) is often used when the counter clock is very slow.
2 It suggested that (3) is often used when the counter clock is very fast.

14.4.6 Pulse Width Modulator (PWM)

Timer 0~7 can be used as Pulse Width Modulator (PWM). The PWM can be used to control the back light inverter or adjust bright or contrast of LCD panel.

FULL comparison match signal and HALF comparison match signal can determine an attribute of the PWM_OUT waveform. FULL comparison match signal specifies the clock cycle for the PWM module clock. HALF comparison match signal specifies the duty ratio for the PWM module clock.



14.4.7 Trackball Input Waveform Detect

Timer 0, 3~7 can be used as a waveform edge counter to count both positive edge and negative edge of an external input waveform. For example, a trackball device's input. 4 timers will need to count all four directions (up, down, left, right). You need configure relate GPIO (set relate 4 PWM IO as input) and set relate TCSR.PWM_IN_EN to 1. Both relate TDFR and TDHR need to set to 0xFFFF, unless you need a special interrupt when the counter hit TDFR or TDHR. The counter will clear to 0 when hit TDFR.

Before the timer counter begin to count up, we need to do as follows:

1 Initial the configuration.
   a Writing TCSR.SD to setting the shutdown mode (Abrupt shutdown or Graceful shutdown).
   b Writing TCSR.PRESCALE to set to 0.
   c Setting TCNT, TDHR and TDFR.
2 Enable the clock.
   a Writing TCSR.PWM_EN to disable PWM.
   b Writing TCSR.EXT_EN, TCSR.RTC_EN and TCSR.PCK_EN to 0, TCSR.PWM_IN_EN to 1 to select the input clock and enable the input clock.

After initialize the register of timer, we should start the counter as follows:

3 Enable the counter.
   Setting the TESR.TCST bit to 1 to enable the TCNT.

NOTE: The input clock and PCLK should follow the rules advanced before.

15 Operating System Timer

15.1 Overview

The OST (Operating System Timer) contains one 64-bit programmable timer. It can be used as operating system timer.

OST has the following features:

- OST includes:
  - 64-bit Counter
  - 32-bit Compare Data Register
  - Control Register
- Independent clock for each counter, selectable by software
  - PCLK, EXTAL and RTCCLK can be used as the clock for counter
  - The division ratio of the clock can be set to 1, 4, 16, 64, 256 and 1024 by software
- Match interrupt can be generated for OST using the compare data registers
  - Interrupt flag and interrupt mask is same with TCU in TCU spec

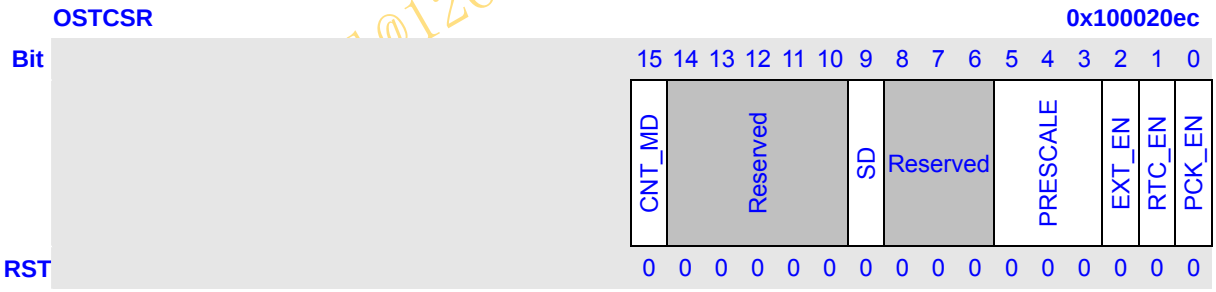
15.2 Register Description

In this section, we will describe the registers in OST. Following table lists all the registers definition. All OST register's 32bit address is physical address. And detailed function of each register will be described below.

Name	Description	RW	Reset Value	Address	Access Size
OSTDR	Operating System Timer Data Register	RW	0x????????	0x100020e0	32
OSTCNTL	Operating System Timer Counter Lower 32 Bits	RW	0x????????	0x100020e4	32
OSTCNTH	Operating System Timer Counter Higher 32 Bits	RW	0x????????	0x100020e8	32
OSTCSR	Operating System Timer Control Register	RW	0x0000	0x100020ec	16
OSTCNTH BUF	Operating System Timer Counter Higher 32 Bits Buffer	R	0x????????	0x100020fc	32

15.2.1 Operating System Control Register (OSTCSR)

The TCSR is a 16-bit read/write register. It contains the control bits for OST. It is initialized to 0x00 by any reset.



Bits	Name	Description	RW
15	CNT_MD	Counter mode choose bit. 0: When the value counter is equal to compare value, the counter will be cleared, and increase from 0. 1: When the value counter is equal to compare value, the counter will go on increasing till overflow, and then increase from 0.	
14:6	Reserved	Writing has no effect, read as zero.	R
9	SD	Shut Down (SD) the PWM output. It is only used in TCU1 mode. 0: Graceful shutdown (only used when CNT_MD = 0) 1: Abrupt shutdown	RW
5:3	PRESCALE	These bits select the TCNT count clock frequency.	RW

		Bit 2	Bit1	Bit 0	Description	
		0	0	0	Internal clock: CLK/1	
		0	0	1	Internal clock: CLK/4	
		0	1	0	Internal clock: CLK/16	
		0	1	1	Internal clock: CLK/64	
		1	0	0	Internal clock: CLK/256	
		1	0	1	Internal clock: CLK/1024	
		110~111			Reserved	
2	EXT_EN	Select EXTAL as the timer clock input. 0: Disable 1: Enable				RW
1	RTC_EN	Select RTCCLK as the timer clock input. 1: Enable 0: Disable				RW
0	PCK_EN	Select PCLK as the timer clock input. 1: Enable 0: Disable				RW

NOTE: The input clock of timer and the PCLK should keep to the rules as follows.

Input clock of timer: IN_CLK	**Clock generated from the frequency divider (PRESCALE): DIV_CLK**
PCK_EN == 0, RTC_EN == 1 and EXT_EN == 0 (IN_CLK = RTCCLK)	$f_{DIV\_CLK} < ½ f_{PCLK}$
PCK_EN == 0, RTC_EN == 0 and EXT_EN == 1 (IN_CLK = EXTAL)	$f_{DIV\_CLK} < ½ f_{PCLK}$
PCK_EN == 1, RTC_EN == 0 and EXT_EN == 0 (IN_CLK = PCLK)	ANY

15.2.2 Operating System Timer Data Register (OSTDR)

The operating system timer data register OSTDR is used to store the data to be compared with the content of the operating system timer up-counter OSTCNT. This register can be directly read and written. Please also refer to CNT_MD bit of register **OSTCSR** to understand the counter behavior. (Default: indeterminate)

OSTDR **0x100020e0**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	OSTDR																															
RST	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?

15.2.3 Operating System Timer Counter (OSTCNTH, OSTCNTL)

The operating system timer counter (OSTCNT) is a 64-bit read/write counter. The up-counter OSTCNT can be set by software and counts up using the prescaler output clock. The data can be read out at any time. The counter data can be written at any time. (Default: indeterminate)

OSTCNTH **0x100020e8**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	OSTCNTH																															
RST	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?

OSTCNTL **0x100020e4**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	OSTCNTL																															
RST	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?

15.2.4 Operating System Timer Counter high 32 bits buffer (OSTCNTHBUF)

The operating system timer counter high 32 bits buffer OSTCNTHBUF is used to store the high 32 bits of OSTCNT when its lower 32 bits are read by software. This register can be directly read. (Default: indeterminate)

OSTCNTHBUF **0x100020fc**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	OSTCNTHBUF																															
RST	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?

15.3 Operation

15.3.1 Basic Operation

Before the timer counter begin to count up, we need to do as follows:

1 Initial the configuration.
   a Writing TCSR.SD to setting the shutdown mode (Abrupt shutdown or Graceful shutdown).
   b Writing OSTCSR.PRESCALE to set OSTCNT count clock frequency.
   c Setting OSTCNTL/H and OSTDR.

2 Enable the clock.
   Writing OSTCSR.EXT_EN, OSTCSR.RTC_EN or OSTCSR.PCK_EN to 1 to select the input clock and enable the input clock. Only one of OSTCSR.EXT_EN, OSTCSR.RTC_EN and OSTCSR.PCK_EN can be set to 1.

After initialize the register of timer, we should start the counter as follows:

3 Enable the counter.
   Setting the TESR.OSTCST bit to 1 to enable the OSTCNT.

NOTE: The input clock and PCLK should follow the rules advanced before.

15.3.2 Disable and Shutdown Operation

1 Setting the TECR.OSTCCL bit to 1 to disable the OSTCNT.

16 Watchdog Timer

16.1 Overview

The watchdog timer is used to resume the processor whenever it is disturbed by malfunctions such as noise and system errors. The watchdog timer can generate the reset signal.

Features:

- Generates WDT reset
- A 16-bit Data register and a 16-bit counter
- Counter clock uses the input clock selected by software
  - PCLK, EXTAL and RTCCLK can be used as the clock for counter
  - The division ratio of the clock can be set to 1, 4, 16, 64, 256 and 1024 by software

16.2 Register Description

In this section, we will describe the registers in WDT. Following table lists all the registers definition. All WDT register's 32bit address is physical address. And detailed function of each register will be described below.

Name	Description	RW	Reset Value	Address	Access Size
TDR	Watchdog Timer Data Register	RW	0x????	0x10002000	16
TCER	Watchdog Counter Enable Register	RW	0x00	0x10002004	8
TCNT	Watchdog Timer Counter	RW	0x????	0x10002008	16
TCSR	Watchdog Timer Control Register	RW	0x0000	0x1000200C	16

16.2.1 Watchdog Control Register (TCSR)

The TCSR is a 16-bit read/write register. It contains the control bits for WDT. It is initialized to 0x00 by any reset.

TCSR 0x1000200C

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved										PRESCALE			EXT_EN	RTC_EN	PCK_EN
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
15:6	Reserved	Writing has no effect, read as zero.	R
5:3	PRESCALE	These bits select the TCNT count clock frequency.	RW
2	EXT_EN	Select EXTAL as the timer clock input. 1: Enable 0: Disable	RW
1	RTC_EN	Select RTCCLK as the timer clock input. 1: Enable	RW

Bit 2	Bit1	Bit 0	Description
0	0	0	Internal clock: CLK/1
0	0	1	Internal clock: CLK/4
0	1	0	Internal clock: CLK/16
0	1	1	Internal clock: CLK/64
1	0	0	Internal clock: CLK/256
1	0	1	Internal clock: CLK/1024
110~111			Reserved

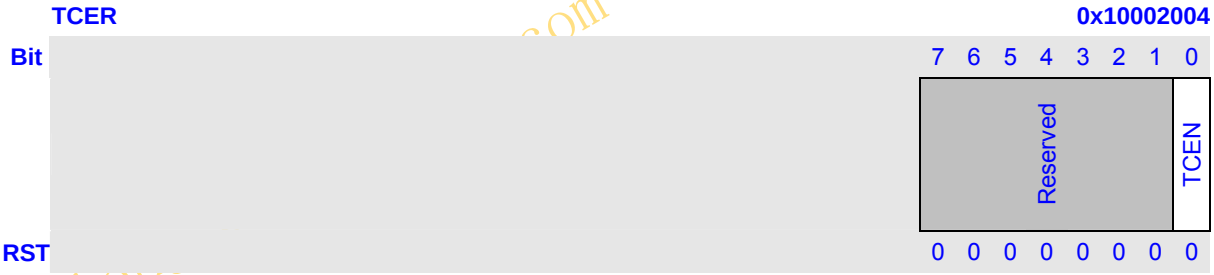
		0: Disable	
0	PCK_EN	Select PCLK as the timer clock input. 1: Enable 0: Disable	RW

NOTE: The input clock of timer and the PCLK should keep to the rules as follows:

Input clock of timer: IN_CLK	**Clock generated from the frequency divider (PRESCALE): DIV_CLK**
PCK_EN == 0, RTC_EN == 1 and EXT_EN == 0 (IN_CLK = RTCCLK)	$f_{DIV\_CLK} < ½ f_{PCLK}$
PCK_EN == 0, RTC_EN == 0 and EXT_EN == 1 (IN_CLK = EXTAL)	$f_{DIV\_CLK} < ½ f_{PCLK}$
PCK_EN == 1, RTC_EN == 0 and EXT_EN == 0 (IN_CLK = PCLK)	ANY

16.2.2 Watchdog Enable Register (TCER)

The TCER is an 8-bit read/write register. It contains the counter enable control bits for watchdog. It is initialized to 0x00 by any reset.



Bits	Name	Description	RW
7:1	Reserved	Writing has no effect, read as zero.	R
0	TCEN	Counter enable control. 0: Timer stop 1: Timer running	RW

16.2.3 Watchdog Timer Data Register (TDR)

The watchdog timer data register TDR is used to store the data to be compared with the content of the watchdog timer up-counter TCNT. This register can be directly read and written. (Default: indeterminate)

TDR																0x10002000
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TDR															
RST	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?

16.2.4 Watchdog Timer Counter (TCNT)

The watchdog timer counter (TCNT) is a 16-bit read/write counter. The up-counter TCNT can be reset to 0 by software and counts up using the prescaler output clock. When TCNT count up to equal to TDR, the comparison match signal will be generated and a WDT reset is generated. The data can be read out at any time. The counter data can be written at any time. (Default: indeterminate)

TCNT																0x10002008
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TCNT															
RST	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?

16.3 Watchdog Timer Function

The following describes steps of using WDT:

1 Setting the PRESCALE of input clock in register TCSR.
2 Set register TDR and TCNT.
3 Select the input clock and enable the input clock in register TCSR.

After initialize the register of timer, we should start the counter as follows:

4 Set TCEN bit in TCER to 1. The counter TCNT begins to count.
5 If TCNT = TDR, a WDT reset will be generated.

NOTES:

1 The input clock and PCLK should follow the rules advanced before.
2 The clock of WDT can be stopped by setting register TSR, and register TSR can only be set by register TSSR or TSCR. The content of register TSR, TSSR and TSCR can be found in TCU spec.

17 XBurst Boot ROM Specification

The JZ4770 contains an internal 8KB boot ROM. The CPU boots from the boot ROM after reset.

17.1 Boot Select

The boot sequence of the JZ4770 is controlled by boot_sel [2:0]. The configuration is shown as follow:

Table 17-1 Boot Configuration of JZ4770

boot_sel[2:0]	Boot method
111	NAND boot @ CS1
100	SD boot @ MSC0 (use GPIO Port A)
000	eMMC boot @ MSC0 (use GPIO Port A)
101	SPI boot @ SPI0/CE0
011	NOR boot @ CS4 (just for FPGA testing)
110	USB boot @ USB 2.0 device, EXTCLK=12MHz
001	USB boot @ USB 2.0 device, EXTCLK=26MHz
010	USB boot @ USB 2.0 device, EXTCLK=19.2MHz

17.2 Boot Procedure

After reset, the boot program on the internal boot ROM executes as follows:

1 Disable all interrupts and read boot_sel[2:0] to determine the boot method.

2 If it is boot from NAND flash, 4 flags at the beginnig of NAND are read to know the NAND information including bus width(8 or 16 bits), page cycle(2 or 3 cycles) and its page size(512B, 2KB, 4KB or 8KB). Then 8KB code are read out from NAND to cache, if the 8KB reading failed, the next 8KB backup in NAND will be read. Then branch to cache at 12 bytes offset.

3 There 8KB backup reading failed, the 8KB backup at 64th, 128th, 192th , …, and finally 1280th page will be tried in consecutive order.

4 If it is boot from MMC/SD card at MSC0, its function pins MSC0_D0, MSC0_CLK, MSC0_CMD are initialized, the boot program loads the 8KB code from MMC/SD card to cache and jump to it. Only one data bus which is MSC0_D0 is used. The clock EXTCLK/128 is used initially. When reading data, the clock EXTCLK/4 is used.

5 If it is boot from eMMC boot partition1 at MSC0, its function pins MSC0_D0, MSC0_CLK, MSC0_CMD are initialized, the boot program loads the 16KB code from eMMC boot partition1 to cache and jump to it. Only one data bus which is MSC0_D0 is used. The clock EXTCLK/4 is used.

6 If it is boot from USB, a block of code will be received through USB cable connected with host PC and be stored in cache. Then branch to this area in cache.

NOTE: The JZ4770's cache is 16KB, its address is from 0x80000000 to 0x80004000.

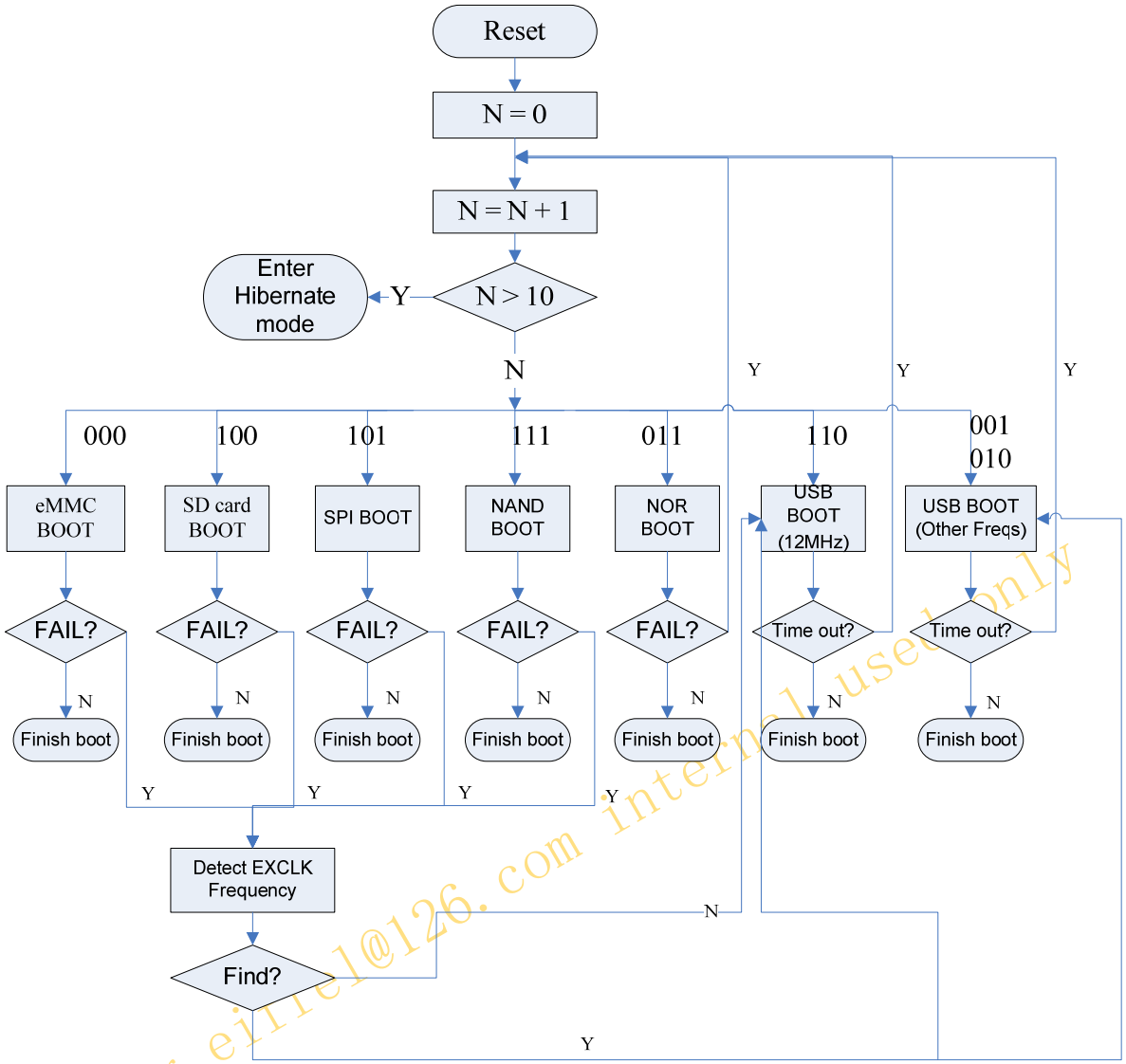


Figure 17-1 Boot sequence diagram of JZ4770

17.3 NAND Boot Specification

If CPU boots from NAND flash (CS1), the boot ROM will read 4 flags from NAND flash to know the NAND information including bus width(8 or 16 bits), page cycle(2 or 3 cycles) and its page size(512, 2KB, 4KB or 8KB bytes).

The content and definition of the 4 flags are shown as follow:

Table 17-2 The definition of 4 flags in NAND flash

Name	Location (in byte)	length (in byte)	Value	Description
buswidth_flag	0-63	64	0x55 or 0xaa	Bus width. 0x55: 8bit bus width 0xaa: 16bit bus width
rowcycle_flag	64-95	32	0x55 or 0xaa	The number of bytes of row cycles. 0x55: 2-byte row cycles 0xaa: 3-byte row cycles
pagesize_flag1	96-127	32	0x55 or 0xaa	pagesize_flag1 pagesize_flag0 pagesize(byte). 0x55 0x55 512 0x55 0xaa 2048 0xaa 0x55 4096 0xaa 0xaa 8192
pagesize_flag0	128-159	32	0x55 or 0xaa	

The buswidth_flag containing 64 bytes locates at the beginning of NAND, if the bus width of NAND is 8 bit, the buswidth_flag should be filled with 0x55 for all 64 bytes, or else it should be filled with 0xaa. The rowcycle_flag containing 32 bytes locates behind the buswidth_flag, if the number of bytes of row cycles is 2, the flag should be filled with 0x55 for all 32 bytes, or else it should be filled with 0xaa. The pagesize_flag1 and pagesize_flag0 each containing 32 bytes locate behind the rowcycle_flag, which value should be filled is determined by the page size of NAND. Please refer to table 2. Totally, 160 bytes are allocated for the 4 flags.

At first, the first 256 bytes (which is a PN* unit) in NAND containing 4 flags will be read out to a buffer assuming the bus width of NAND is 8 bit. The buswidth_flag will be get from the buffer to detect the page size (whether 512B or not) and bus width of nand. If there is no 0x55 or 0xaa in buswidth_flag, 64th, 128th, 192th, ..., 1280th page will be tried in sequence. If failed at 1280th page, bootrom will jump to usb_boot. If bus width is 16 bit, 256 bytes will be read again to the buffer mentioned above. If buswidth_flag is valid, the rowcycle_flag will be obtained from the buffer to know the number of row cycles. At last, pagesize_flag1 and pagesize_flag0 will be obtained from the buffer to know precise page size.

8KB codes in NAND will be loaded up to dcache and transferred to icache and branch to icache at 160 bytes offset. Hardware PN and 24-bit BCH ECC will be used for every 256 bytes during reading. The

ECC(39 bytes per 256 bytes data) stores in the data area of a NAND page behind the page storing code data. If no ECC error is detected or ECC error is correctable(number of error bits <= 24), NAND boots successfully. If uncorrectable error occurred, next 8KB backup at 64 pages behind will be tried. 64th, 128th, 192th, ..., 1280th page will be tried in sequence. If failed at 1280th page, bootrom will jump to usb_boot.

The distribution and structure of the boot code in NAND is shown as Figure 17-2.
The procedure of the JZ4770 NAND boot is shown as Figure 17-3.

NOTE: PN is short for pseudorandom noise which is used for supporting TLC (three-level cell) NAND.

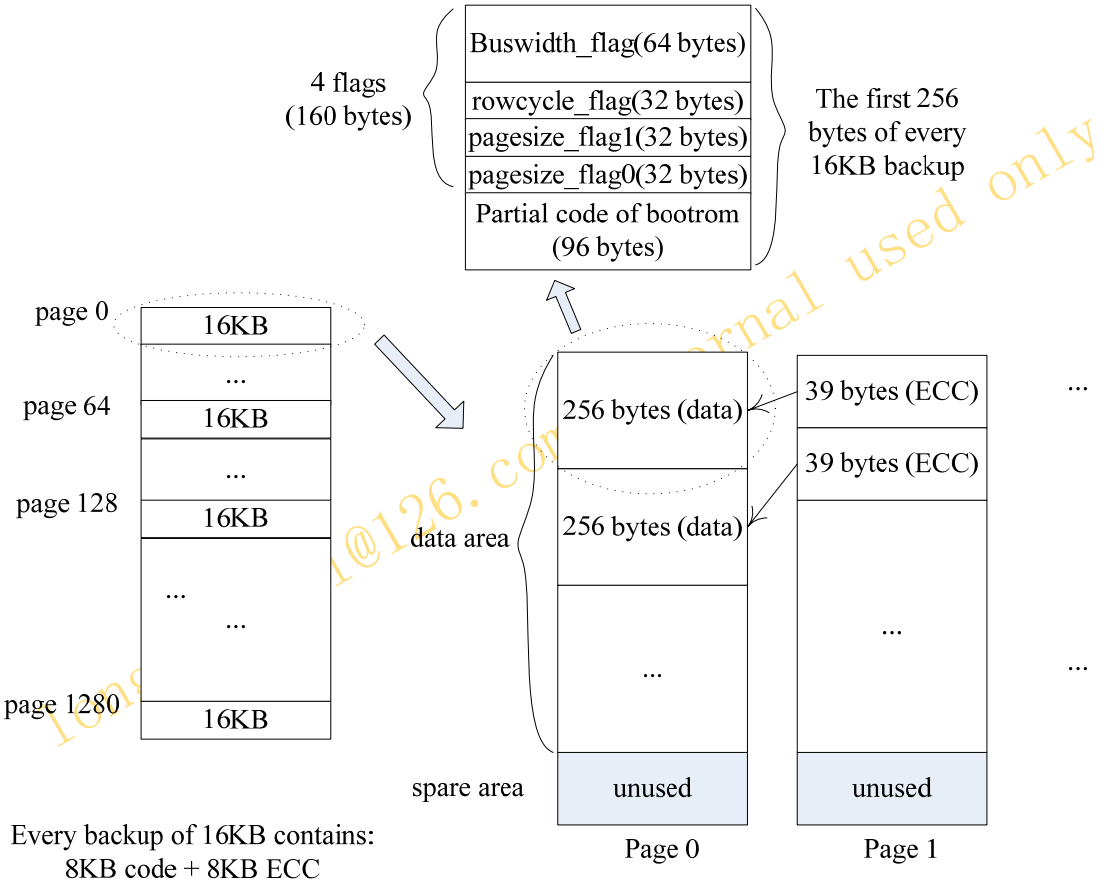


Figure 17-2 the distribution and structure of the boot code in NAND

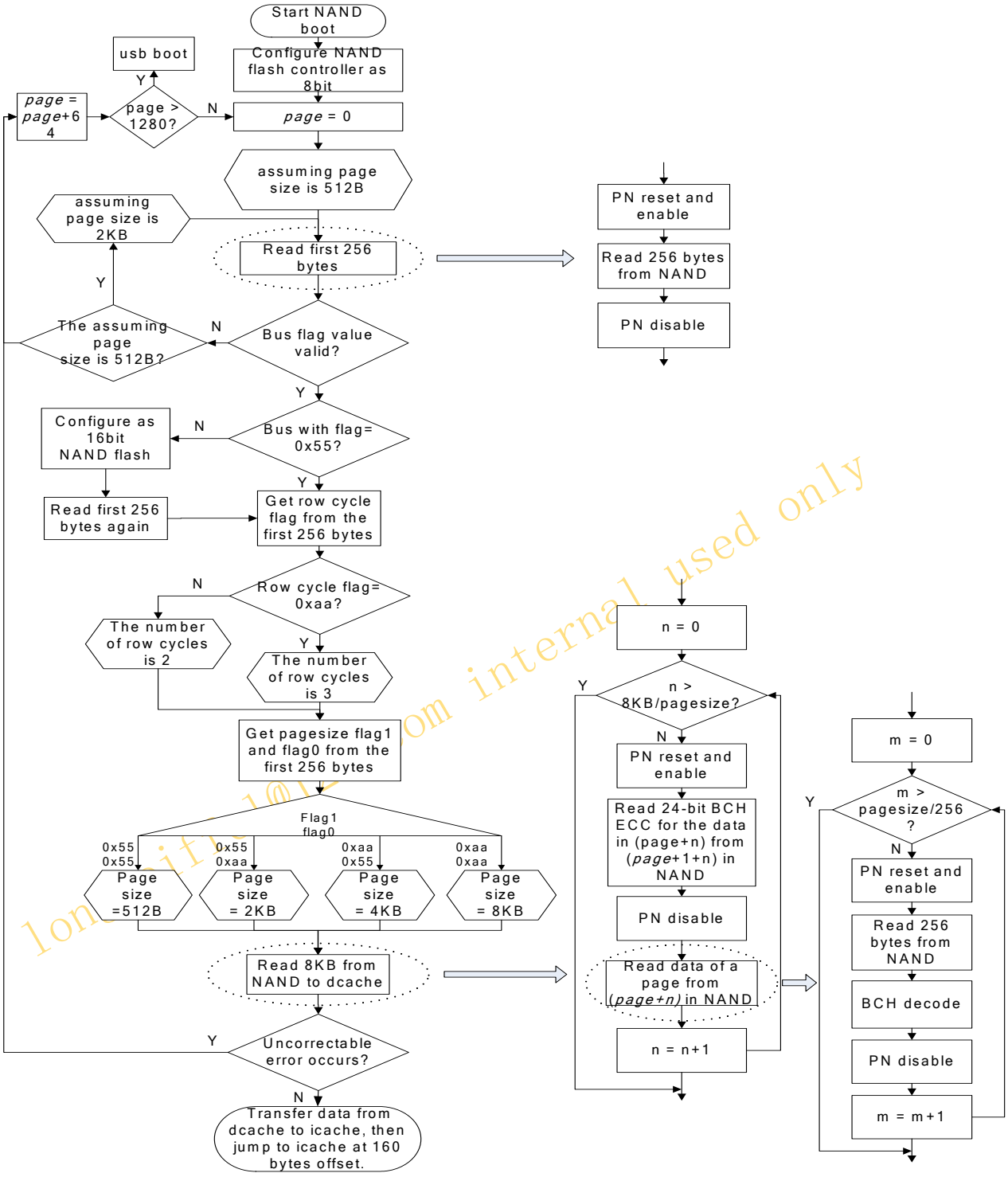


Figure 17-3 JZ4770 NAND Boot Procedure

17.4 USB Boot Specification

When boot_sel[2:0] is selected as USB boot, the internal boot ROM downloads user program from the USB port to internal SRAM and branches to the internal SRAM to execute the program.

JZ4770 supports the external main crystal whose frequency is 12MHz, 19.2MHz, 26MHz.

The boot program supports both high-speed (480MHz) and full-speed (12MHz) transfer modes. The boot program uses the following two transfer types.

Table 17-3 Transfer Types Used by the Boot Program

Transfer Type	Description
Control Transfer	Used for transmitting standard requests and vendor requests.
Bulk Transfer	Used for responding to vendor requests and transmitting a user program.

The following figure shows an overview of the USB communication flow.

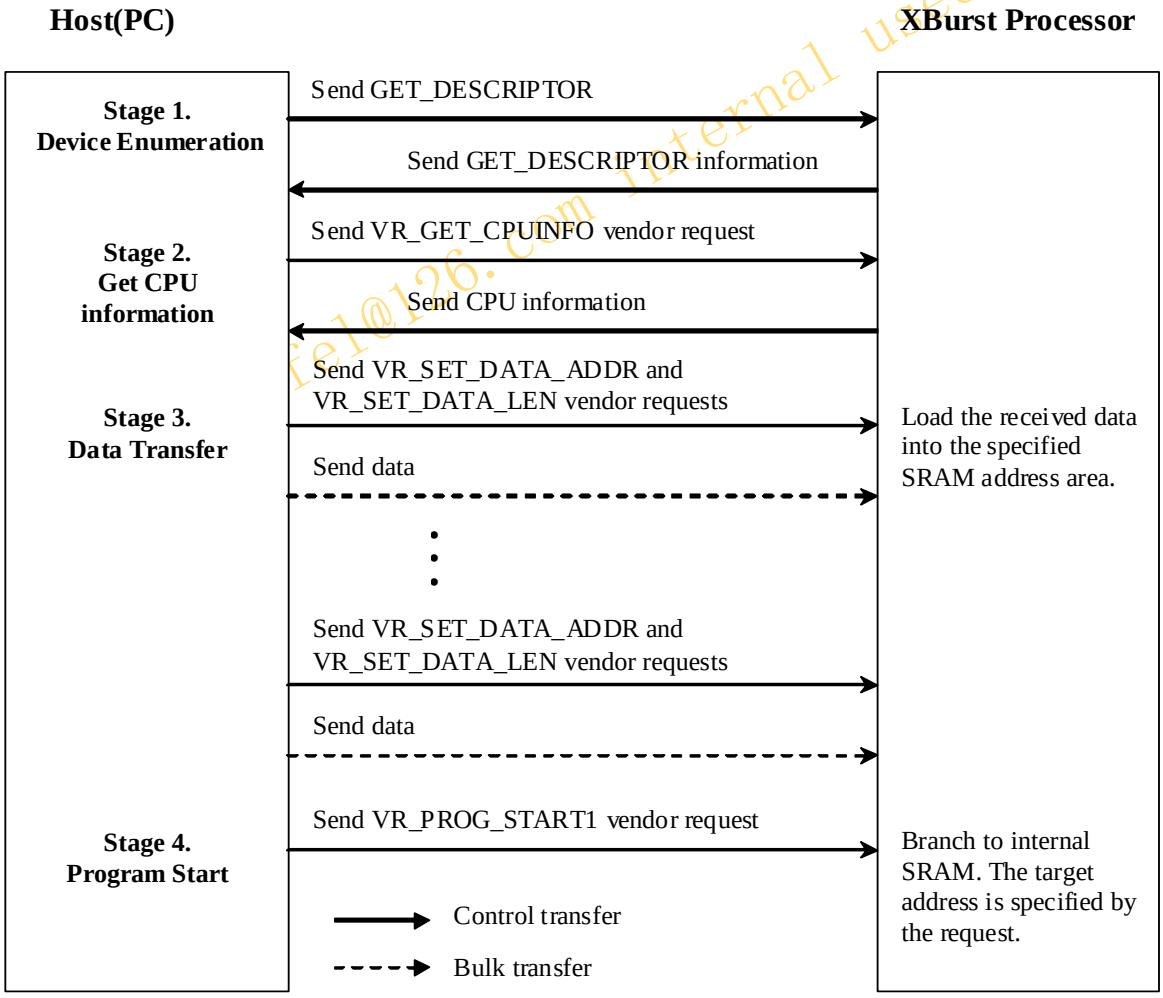


Figure 17-4 USB Communication Flow

The vendor ID and product ID for the USB boot device are 0xa108 and 0x4770 respectively. The Configuration for USB is for Control Endpoint 0 with Max Packet Size equals 64 bytes, Bulk IN at Endpoint 1 with Max Packet Size equals 512 bytes in high-speed and 64 bytes in full-speed, Bulk OUT at Endpoint 1 with Max Packet Size equals 512 bytes in high-speed and 64 bytes in full-speed.

The USB boot program provides six vendor requests through control endpoint for user to download/upload data to/from device, and to branch to a target address to execute user program. The six vendor requests are VR_GET_CPU_INFO (0x00), VR_SET_DATA_ADDRESS (0x01), VR_SET_DATA_LENGTH (0x02), VR_FLUSH_CACHES (0x03), VR_PROGRAM_START1 (0x04) and VR_PROGRAM_START2 (0x05). User program is transferred through Bulk IN or Bulk OUT endpoint.

When JZ4770 is reset with boot_sel[2:0] equals 110b, 001b or 010b, the internal boot ROM will switch to USB boot mode and wait for USB requests from host. After connecting the USB device port to host, host will recognize the connection of a USB device, and start device enumeration. After finishing the device enumeration, user can send VR_GET_CPU_INFO (0x00) to query the device CPU information. If user wants to download/upload a program to/from device, two vendor requests VR_SET_DATA_ADDRESS (0x01) and VR_SET_DATA_LENGTH (0x02) should be sent first to tell the device the address and length in byte of the subsequent transferring data. Then data can be transferred through bulk-out/bulk-in endpoint. After this first stage program has been transferred to device, user can send vendor request VR_PROGRAM_START1 (0x04) to let the CPU to execute the program. This first stage program must not greater than 16KB and is normally used to init GPIO and SDRAM of the target board. At the end of the first stage program, it can return back to the internal boot ROM by jumping to ra ($31) register. Thus user can download a new program to the SDRAM of the target board like the first stage, and send vendor request VR_FLUSH_CACHES (0x03) and VR_PROGRAM_START2 (0x05) to let the CPU to execute the new program. Next figure is the typical procedure of USB boot.

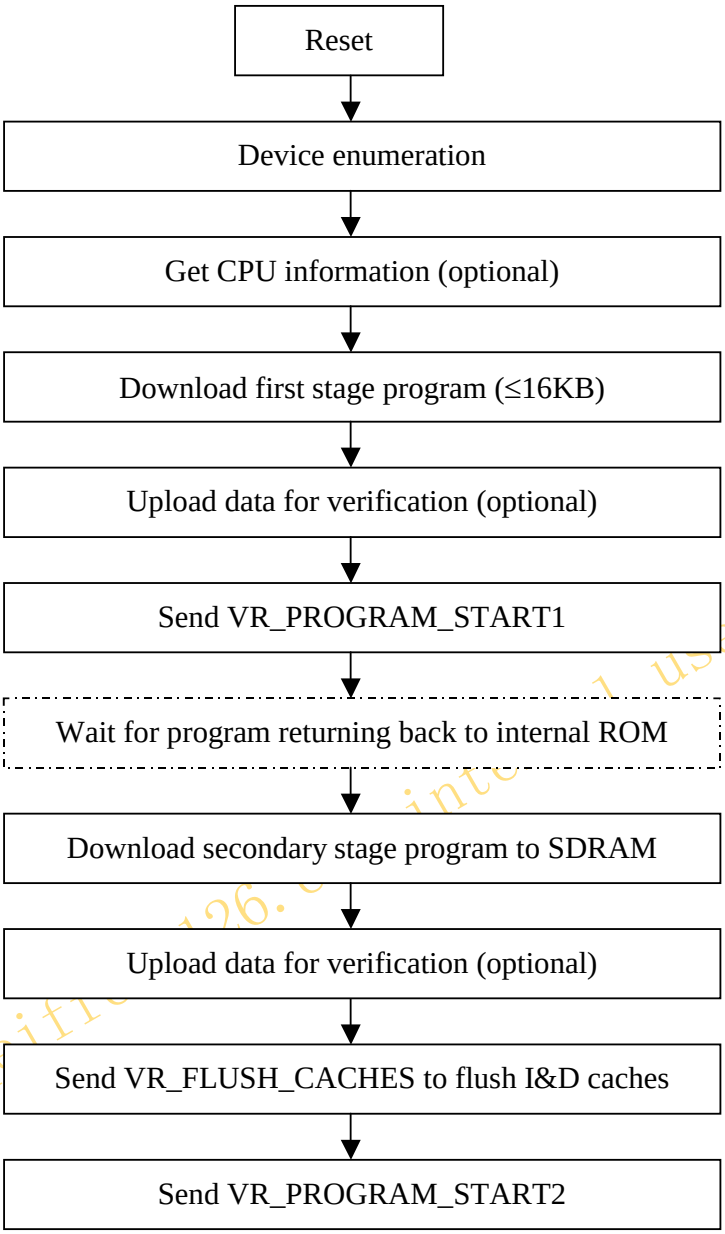


Figure 17-5 Typical Procedure of USB Boot

Following tables list all the vendor requests that USB boot program supports:

Table 17-4 Vendor Request 0 Setup Command Data Structure

Offset	Field	Size	Value	Description
0	bmRequestType	1	40H	D7 0: Host to Device. D6-D5 2: Vendor. D4-D0 0: Device.
1	bRequest	1	00H	VR_GET_CPU_INFO: get CPU information.
2	wValue	2	0000H	Not in used.
4	wIndex	2	0000H	Not in used.
6	wLength	2	0008H	8 bytes.

Table 17-5 Vendor Request 1 Setup Command Data Structure

Offset	Field	Size	Value	Description
0	bmRequestType	1	40H	D7 0: Host to Device. D6-D5 2: Vendor. D4-D0 0: Device.
1	bRequest	1	01H	VR_SET_DATA_ADDRESS: set address for next bulk-in/bulk-out transfer.
2	wValue	2	xxxxH	MSB (bit[31:16]) of the data address.
4	wIndex	2	xxxxH	LSB (bit[15:0]) of the data address.
6	wLength	2	0000H	Not in used.

Table 17-6 Vendor Request 2 Setup Command Data Structure

Offset	Field	Size	Value	Description
0	bmRequestType	1	40H	D7 0: Host to Device. D6-D5 2: Vendor. D4-D0 0: Device.
1	bRequest	1	02H	VR_SET_DATA_LENGTH: set length in byte for next bulk-in/bulk-out transfer.
2	wValue	2	xxxxH	MSB (bit[31:16]) of the data length.
4	wIndex	2	xxxxH	LSB (bit[15:0]) of the data length.
6	wLength	2	0000H	Not in used.

Table 17-7 Vendor Request 3 Setup Command Data Structure

Offset	Field	Size	Value	Description
0	bmRequestType	1	40H	D7 0: Host to Device. D6-D5 2: Vendor. D4-D0 0: Device.
1	bRequest	1	03H	VR_FLUSH_CACHES: flush I-Cache and D-Cache.
2	wValue	2	0000H	Not in used.
4	wIndex	2	0000H	Not in used.
6	wLength	2	0000H	Not in used.

Table 17-8 Vendor Request 4 Setup Command Data Structure

Offset	Field	Size	Value	Description
0	bmRequestType	1	40H	D7 0: Host to Device. D6-D5 2: Vendor. D4-D0 0: Device.
1	bRequest	1	04H	VR_PROGRAM_START1: transfer data from D-Cache to I-Cache and branch to address in I-Cache. **NOTE:** After downloading program from host to device for the first time, you can only use this request to start the program. Since the USB boot program will download data to D-Cache after reset. This request will transfer data from D-Cache to I-Cache and execute the program in I-Cache.
2	wValue	2	xxxxH	MSB (bit[31:16]) of the program entry point.
4	wIndex	2	xxxxH	LSB (bit[15:0]) of the program entry point.
6	wLength	2	0000H	Not in used.

Table 17-9 Vendor Request 5 Setup Command Data Structure

Offset	Field	Size	Value	Description
0	bmRequestType	1	40H	D7 0: Host to Device. D6-D5 2: Vendor. D4-D0 0: Device.
1	bRequest	1	05H	VR_PROGRAM_START2: branch to target address directly.
2	wValue	2	xxxxH	MSB (bit[31:16]) of the program entry point.
4	WIndex	2	xxxxH	LSB (bit[15:0]) of the program entry point.

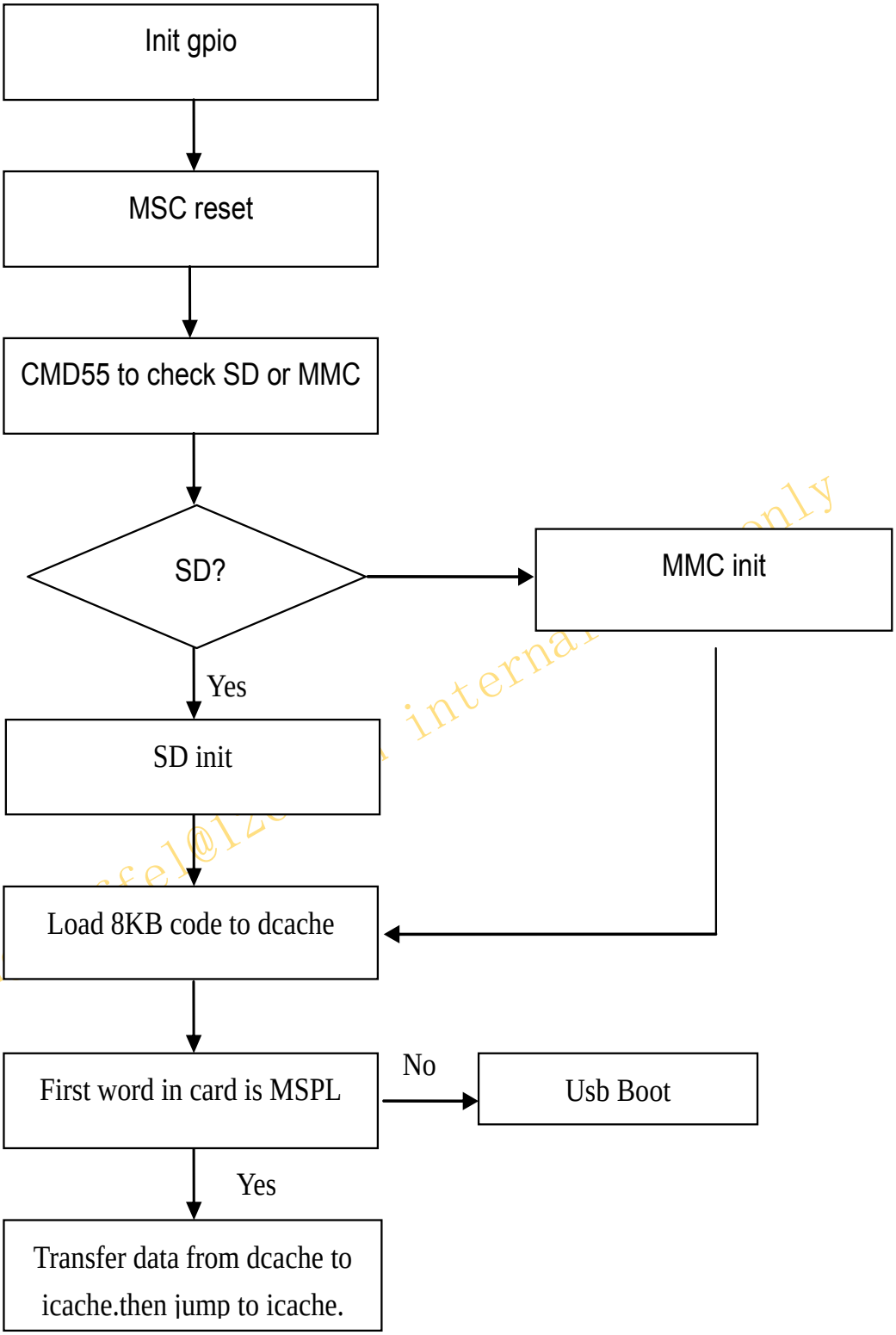
6	WLength	2	0000H	Not in used.

17.5 MMC/SD Boot Specification

All cards can boot from MMC/SD Boot from MSC0, the boot program will load 8KB code starting at sector 0 from MMC/SD card to cache. First the boot program initializes MSC0_D0, MSC0_CLK, MSC0_CMD as function pins. Only one data pin MSC0_D0 is used. Then the boot program sends CMD55 to test if it's SD or MMC card and initializes the card. At last it loads 8KB code from the card to cache and branches to execute the code in cache.

When initializing the card, the clock of EXTCLK/128 is used. And when reading data, the clock of EXTCLK/4 is used.

The procedure of the JZ4770 MMC/SD boot is shown as follow:



17.6 eMMC Boot Specification

If eMMC is MultiMediaCard System Specification Ver. 4.4 compatible, you can use eMMC boot method. you should write boot code to boot partition1 , then the boot program will load 16KB code from eMMC boot partition1 area to cache. First the boot program initializes MSC0_D0, MSC0_CLK, MSC0_CMD as function pins. Only one data pin MSC0_D0 is used. Then the boot program sends CMD0 to set eMMC in boot mode. At last it loads 16KB code from the card to cache and branches to execute the code in cache. and the clock of EXTCLK/4 is used.

The procedure of the JZ4770 eMMC boot is shown as follow:

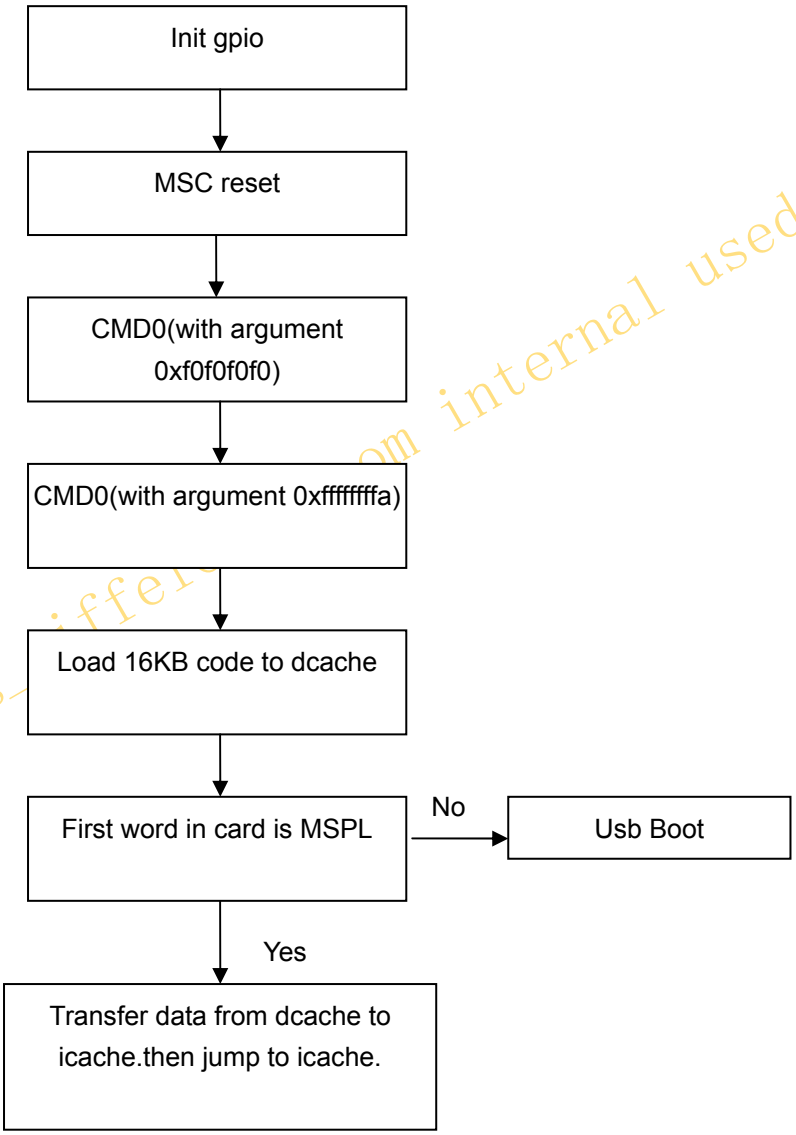


Figure 17-6 JZ4770 eMMC Boot Procedure

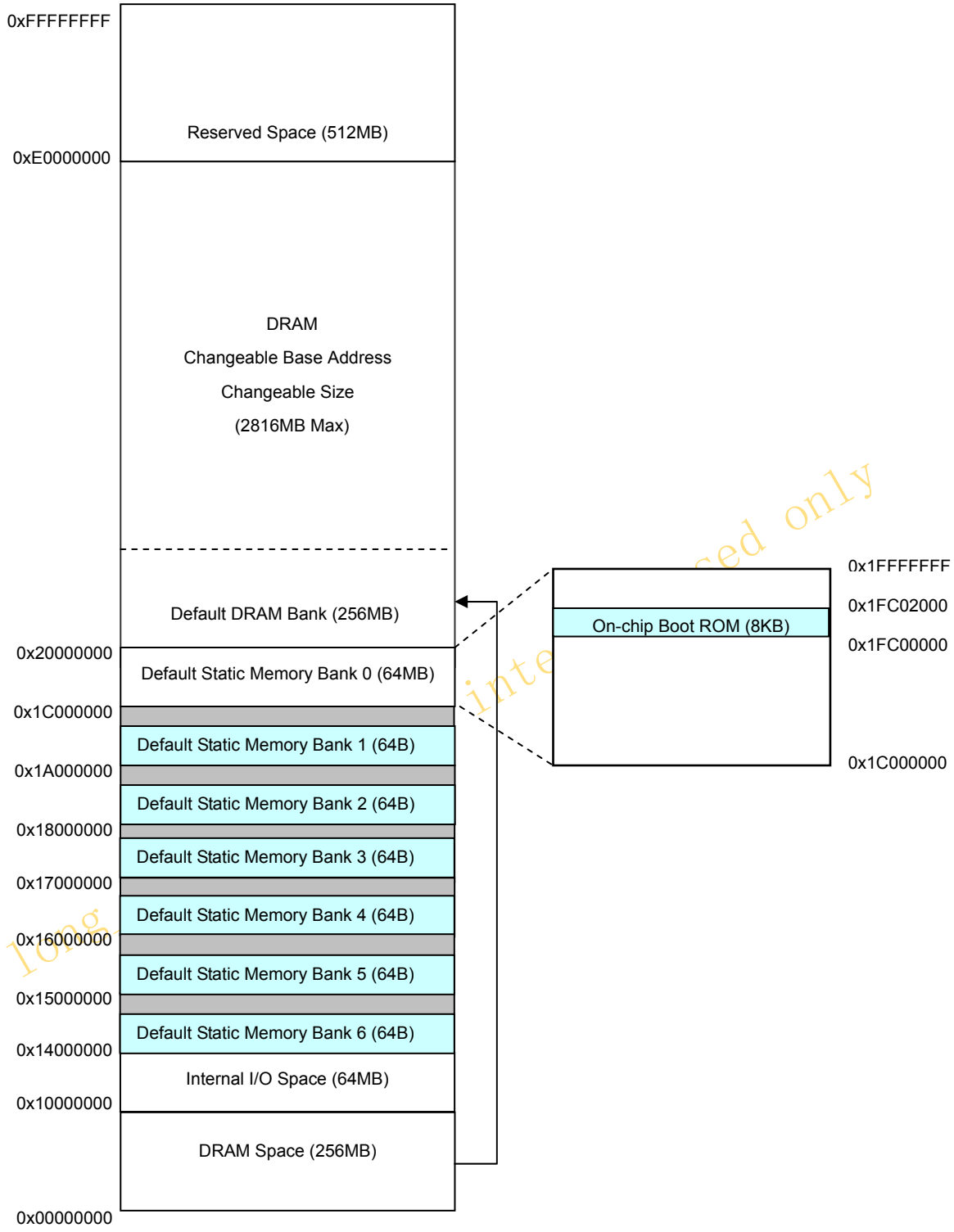
18 Memory Map and Registers

18.1 Physical Address Space Allocation

This chapter describes the physical address map, memory-mapped regions for every block in the JZ4770 processor. Both logical space and physical space of the JZ4770 are 32 bits wide. The 4Gbyte physical space is divided into several partitions for external memory, PCMCIA and internal I/O devices. Table 18-1 shows the basic physical memory map:

Table 18-1 JZ4770 Processor Physical Memory Map

Start Address	End Address	Size (MB)	Function
0x00000000	0x0FFFFFFF	256	DRAM Memory
0x10000000	0x10FFFFFF	16	I/O Devices on APB Bus
0x11000000	0x12FFFFFF	32	Reserved
0x13000000	0x13FFFFFF	16	I/O Devices on AHB Bus
0x14000000	0x1400003F	64B	Static Memory, CS6#
0x14000040	0x14FFFFFF		Reserved
0x15000000	0x1500003F	64B	Static Memory, CS5#
0x15000040	0x15FFFFFF		Reserved
0x16000000	0x1600003F	64B	Static Memory, CS4#
0x16000040	0x16FFFFFF		Reserved
0x17000000	0x1700003F	64B	Static Memory, CS3#
0x17000040	0x17FFFFFF		Reserved
0x18000000	0x1800003F	64B	Static Memory, CS2#
0x18000040	0x19FFFFFF		Reserved
0x1A000000	0x1A00003F	64B	Static Memory, CS1#
0x1A000040	0x1BFFFFFF		Reserved
0x1C000000	0x1FBFFFFF	60	Reserved
0x1FC00000	0x1FC01FFF	0.008	On-chip Boot ROM (8kB)
0x1FC02000	0x1FFFFFFF	3.992	Reserved
0x20000000	0xDFFFFFFF	3072	DRAM Memory
0xE0000000	0xFFFFFFFF	512	Reserved



NOTES:

1 Data width of static memory banks can be configured to 8, 16 or 32 bits by software.

2 The 8KB address space from H'1FC00000 to H'1FC01FFF in bank 0 is mapped to on-chip boot ROM. The other memory spaces in bank 0 are not used.

3 To support large DRAM space, DDRC re-maps the physical address H'00000000-H07FFFFFF to H'20000000-H'27FFFFFF. Software must configure the DRAM base address by the re-mapped address.

The JZ4770 processor AHB bus devices are mapped at the addresses based at 0x13000000, and each device is allocated for 64KB space. Table 18-2 lists the complete addresses:

Table 18-2 AHB0 Bus Devices Physical Memory Map

Module	Start Address	End Address	Size (KB)	Description
HARB	0x13000000	0x1300FFFF	64	AHB Bus Arbiter
DDRC	0x13020000	0x1302FFFF	64	External DDR Controller
LCDC TVE	0x13050000	0x1305FFFF	64	LCD Controller TV Encoder
CIM	0x13060000	0x1306FFFF	64	Camera Interface Module
AOSD Compress	0x13070000	0x1307FFFF	64	Alpha-OSD Controller Compress Controller
IPU	0x13080000	0x1308FFFF	64	Image Process Unit

Table 18-3 AHB1 Bus Devices Physical Memory Map

Module	Start Address	End Address	Size (KB)	Description
DMAGP0	0x13210000	0x1321FFFF	64	2D-DMA Controller 0
DMAGP1	0x13220000	0x1322FFFF	64	2D-DMA Controller 1
DMAGP2	0x13230000	0x1323FFFF	64	2D-DMA Controller 2
MCE	0x13250000	0x1325FFFF	64	Motion Compensation/Estimation
DEBLK1	0x13270000	0x1327FFFF	64	De-Block 1
DEBLK2	0x132D0000	0x132DFFFF	64	De-Block 2
VMAU	0x13280000	0x1328FFFF	64	Video Matrix Arithmetic Unit
SDE	0x13290000	0x1329FFFF	64	Stream DEC/ENC
AUX	0x132A0000	0x132AFFFF	64	Auxiliary cpu core
TCSM0	0x132B0000	0x132BFFFF	64	Tightly coupled sram 0
TCSM1	0x132C0000	0x132CFFFF	64	Tightly coupled sram 1
SRAM	0x132F0000	0x132FFFFF	64	General purpose sram

Table 18-4 AHB2 Bus Devices Physical Memory Map

Module	Start Address	End Address	Size (KB)	Description
HARB	0x13400000	0x1340FFFF	64	AHB Bus Arbiter
NEMC	0x13410000	0x1341FFFF	64	External Normal Memory / Boot ROM / OTP Controller
DMAC	0x13420000	0x1342FFFF	64	DMA Controller
UHC	0x13430000	0x1343FFFF	64	USB 1.1 Host Controller
OTG	0x13440000	0x1344FFFF	64	OTG 2.0 Controller
BDMAC	0x13450000	0x1345FFFF	64	BCH/NAND DMA Controller
GPS	0x13480000	0x1348FFFF	64	GPS Baseband
ETHC	0x134B0000	0x134BFFFF	64	ETH Controller
BCH	0x134D0000	0x134DFFFF	64	BCH Controller
TSSI	0x134E0000	0x134EFFFF	64	TS Slave Interface

The JZ4770 processor APB bus devices are based at 0x10000000, and each device is allocated for 4KB space. Table 18-5 lists the complete addresses:

Table 18-5 APB Bus Devices Physical Memory Map

Module	Start Address	End Address	Size (KB)	Description
CPM	0x10000000	0x10000FFF	4	Clocks and Power Manager
INTC	0x10001000	0x10001FFF	4	Interrupt Controller
TCU OST WDT	0x10002000	0x10002FFF	4	Timer/Counter Unit Operating System Timer Watchdog Timer
RTC	0x10003000	0x10003FFF	4	Real-Time Clock
GPIO	0x10010000	0x10010FFF	4	General-Purpose I/O
AIC CODEC	0x10020000	0x10020FFF	4	AC97/I2S/SPDIF Controller Embedded CODEC
MSC0	0x10021000	0x10021FFF	4	MMC/SD 0 Controller
MSC1	0x10022000	0x10022FFF	4	MMC/SD 1 Controller
MSC2	0x10023000	0x10022FFF	4	MMC/SD 2 Controller

UART0	0x10030000	0x10030FFF	4	UART 0
UART1	0x10031000	0x10031FFF	4	UART 1
UART2	0x10032000	0x10032FFF	4	UART 2
UART3	0x10033000	0x10033FFF	4	UART 3
SCC	0x10040000	0x10040FFF	4	Smart Card Controller
SSI0	0x10043000	0x10043FFF	4	Synchronous Serial Interface 0
SSI1	0x10044000	0x10044FFF	4	Synchronous Serial Interface 1
I2C0	0x10050000	0x10050FFF	4	I2C 0 Bus Interface
I2C1	0x10051000	0x10051FFF	4	I2C 1 Bus Interface
I2C2	0x10055000	0x10055FFF	4	I2C 2 Bus Interface
KBC	0x10060000	0x10060FFF	4	KBC Bus Interface
SADC	0x10070000	0x10070FFF	4	SAR A/D Controller
PCM0	0x10071000	0x10071FFF	4	PCM 0 Interface
OWI	0x10072000	0x10072FFF	4	One-Wire Bus Interface
PCM1	0x10074000	0x10074FFF	4	PCM 1 Interface