Equuleus Series: STR813X/CNS213X, STR818X/CNS218X Network Access Processor

November 2009

Cavium Networks
805 East Middlefield Road
Mountain View, CA 94043
Phone: 650-623-7000
Fax: 650-625-9751
Email: sales@caviumnetworks.com
Web: http://www.caviumnetworks.com

Part Number Mapping Table

The following table describes the high level product and part number mapping of the Star Semiconductor products to the Cavium Networks product numbering system. The last one or two characters (denoted XX) will be the same in the Star and Cavium Networks numbering systems. Please contact your local sales representative for any required assistance with part number conversions.

Star Semiconductor to Cavium Network Part Number Mapping		
Family	**Star Part Numbers**	**Cavium Part Numbers**
Orion	STR91XX	CNS11XX
Orion with Content Processor	STR92XX	CNS12XX
Equuleus with 10/100 Phy	STR813X	CNS213X
Equuleus	STR818X	CNS218X
Orion Evaluation Board	STR91XX-EVB	CNS11XX-EVB
Equuleus Evaluation Board	STR81XX-EVB	CNS21XX-EVB

PUBLISHED BY

Cavium Networks
805 East Middlefield Road
Mountain View, CA 94043
Phone: 650-623-7000
Fax: 650-625-9751
Email: sales@caviumnetworks.com
Web: http://www.caviumnetworks.com

FEATURES

32-bit RISC Core

- ✓ High-performance ARM922-compatible RISC with 8kB I-cache, 8kB D-cache, and memory management unit (MMU) for high-level RTOS, with programmable CPU core clocks up to 200/250MHz
- ✓ Advanced system bus architecture
- ✓ Supports TRACE32, RealView and Multi-ICE JTAG debugging interfaces
- ✓ Support IRQ/FIQ interrupt modes
- ✓ Support Little-/Big-endian ordering
- ✓ Built-in intelligent power management for normal and power-saving mode of operations

Network Interfaces

- ✓ Embedded a 10/100/1000M MAC providing one RGMII/MII/Reverse MII interface for external transceiver
- ✓ Embedded 10Base-T/100Base-TX PHY with Auto MDI/MDIX
- ✓ TCP/UDP/IP header checksum offload
- ✓ 802.3x full duplex flow control and half duplex backpressure flow control
- ✓ Support Wake on LAN

External Memory Interfaces

- ✓ Support 8-bit NOR type flash memory interface, and SPI serial flash booting.
- ✓ Support 16-bit wide DDR/SDR SDRAM, and self-refresh mode when CPU in Sleep mode
- ✓ Support up to 4-banks of asynchronous 8/16-bit SRAM interface
- ✓ 8 channel configurable DMA controller for memory -to-memory, memory-to-peripheral, peripheral -to-memory and peripheral-to-peripheral transfers

Emebeed SDRAM Memory

- ✓ Only CNS2135 supports 8MB(4Mx16) SDRAM

Peripheral Interfaces

- ✓ Embedded two-port USB2.0-compliant host PHY and EHCI/OHCI controller
- ✓ Embedded one-port USB2.0-compliant device PHY and controller
- ✓ Two high-speed 16C550-compliant UART, with DMA hardware handshake (*STR8132/CNS2132 only supports one UART)
- ✓ 8/16-bit external I/O interface
- ✓ 32-bit 66/33MHz PCI v.2.2-compliant host bridge for up to 2 PCI devices
- ✓ Cardbus slot with hot insertion and remove, compliant to PCI Hot Plug Spec v1.0
- ✓ Two 32-bit programmable timers, one fast timer, one watch-dog timer, and one real time clock.
- ✓ Support up to two IDE devices, compliant to PIO, DMA(mode 0~2), and Ultra DMA(mode 0~5) modes
- ✓ Vector interrupt controller with interrupt source priority classification
- ✓ PCM supports up to 4 external voice

codec/SLIC's with Tx/Rx buffers, master/slave modes and IDL/GCI clocking modes

- ✓ Serial interfaces: Two Wire Interface (TWI), I2S (with DMA, and sampling clock at 32, 44.1, and 48kHz) and SPI (with DMA, up to 4 devices).
- ✓ Selectable external clock outputs, including 25MHz, 12MHz and other options for system use.
- ✓ Integrated PLL clock generator for all the clocks required, from 25MHz and 32.768kHz references
- ✓ Supports up to 50 GPIO's

Electrical Characteristics

- ✓ Packages:
  PQFP-128 (for STR8131/CNS2131, STR8132/CNS2132, CNS2135, STR8181 /CNS2181),
  LFBGA-269 (for STR8133/ CNS2133, STR8182/CNS2182, STR8182X/CNS2182X)
- ✓ Power Supply:
  Optimized on-chip voltage regulator controller provides 2.5V/1.8V/1.25V supplies with low-cost external PNP pass transistors, from single 3.3V supply input
- ✓ Power consumption: within 1.25 watt

Targeted Applications

- ✓ Software VoIP ATA/IP Phone, Cordless-over-IP
- ✓ Network Storage/Download Station (USB/IDE)
- ✓ MIMO Access Point
- ✓ Wireless IP Camera/Surveillance DVR
- ✓ Internet Stereo Radio Tuner
- ✓ UWB Device/Host Wire Adapter (DWA/HWA)
- ✓ Network Processor for Smart/Management Gigabit Ethernet Switch Controller
- ✓ Serial Server (RS232-to-Ethernet/802.11)

Available Data Deliverables

- ✓ Embedded Linux BSP
- ✓ Technical Documents (Data Sheets, User Guides, Application Notes, and Test Reports etc)
- ✓ Reference design schematics

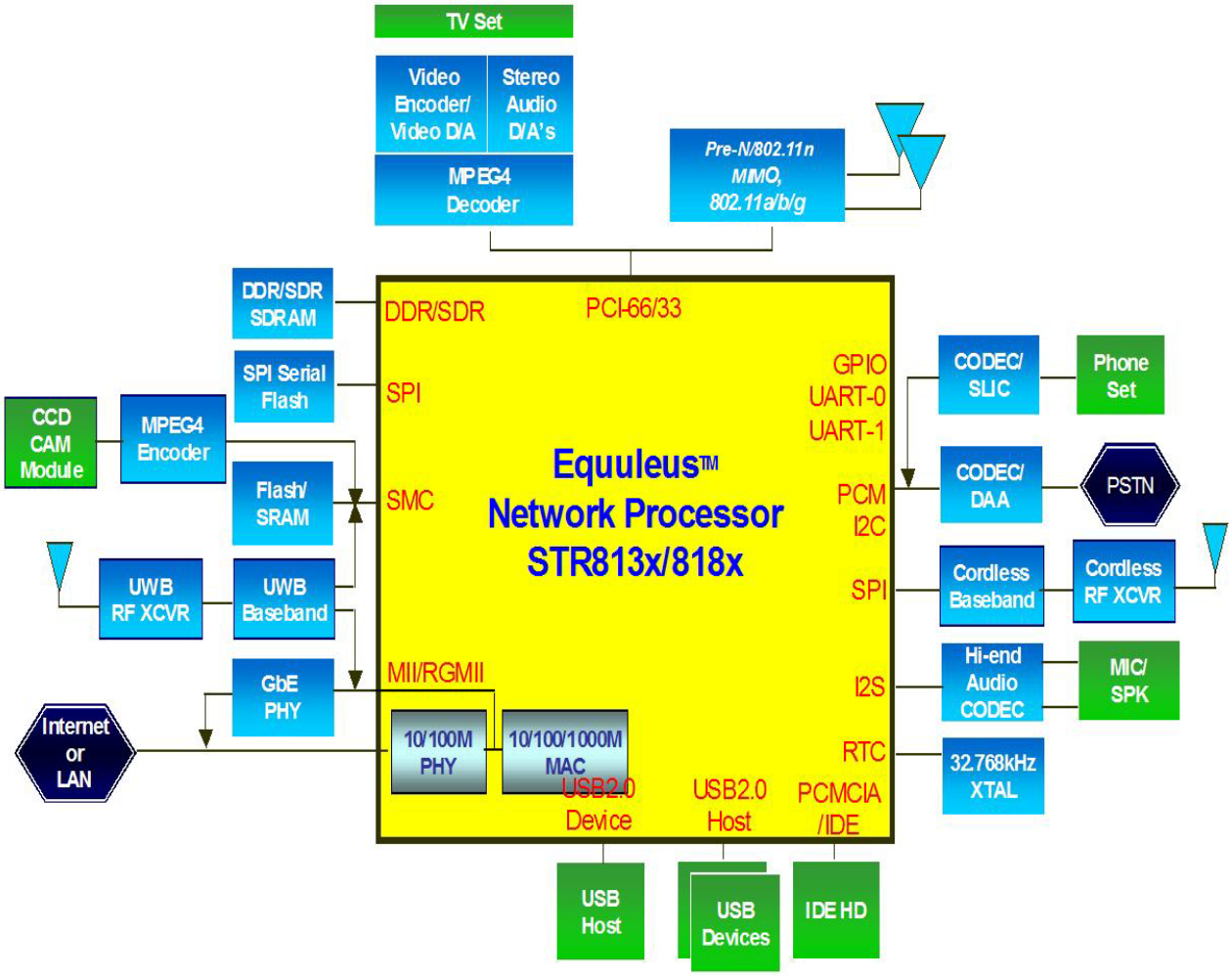


Figure 1. Overview of Application Platform (STR813X/CNS213X, STR818X/CNS218X)

Cavium Networks
805 East Middlefield Road
Mountain View, CA 94043
Phone: 650-623-7000
Fax: 650-625-9751
Email: sales@caviumnetworks.com
Web: http://www.caviumnetworks.com

Table 1. Cross Reference Matrix of P/N versus Applicable User Interfaces

P/N / Interfaces	STR8131/ CNS2131 (w/ FE-PHY)	STR8132/ CNS2132 (w/ FE-PHY)	STR8133/ CNS2133 (w/ FE-PHY)	CNS2135 (w/ FE-PHY)	STR8181/ CNS2181 (w/ MII, RGMII)	STR8182/ CNS2182 (w/ MII, RGMII)	STR8182X/ CNS2182X (w/ MII, RGMII)
Typical Applications	Software/ Hardware VoIP/CoIP/ATA(wired)	USB Download Station, Internet Radio, Smart Network Controller	Wireless-enabled VoIP, IP CAM, DVR, Download Station, Serial Server	USB Download Station, Intelligent Network Controller, Printer Server	Generic Network Processor	GbE Smart Switch Host, High Performance GbE Gateway	GbE Smart Switch Host, High Performance GbE Gateway
Package Type	PQFP-128	PQFP-128	LFBGA-269	PQFP-128	PQFP-128	LFBGA-269	LFBGA-269
Embedded SDRAM Memory				8MB (4Mx16)			
Serial Flash	✓	✓	✓	✓	✓	✓	✓
Parallel Flash	-	-	✓	-	-	✓	✓
DDR/SDR	✓	✓	✓	✓	✓	✓	✓
10/100M PHY	✓	✓	✓	✓	-	-	-
RGMII/MII/ Reverse MII	-	-	-	-	✓	✓	✓
USB2.0 Host	✓	✓	✓	✓	✓	✓	✓
USB2.0 Device	-	✓	✓	-	-	-	✓
IDE	-	-	✓	-	-	✓	✓
PCI	-	-	✓ (x2)	-	-	✓ (x2)	✓ (x2)
SPI	✓ (x3)	✓ (x2)	✓ (x4)	✓ (x2)	✓ (x4)	✓ (x4)	✓ (x4)
I2S	✓	✓	✓	✓	-	✓	✓
TWI	✓	✓	✓	-	-	✓	✓
PCM	✓	-	✓	✓	-	✓	✓
UART	✓ (x2)	✓ (x1)	✓ (x2)	✓ (x1)	✓ (x2)	✓ (x2)	✓ (x2)
GPIO	Up to 21	Up to 13	Up to 50	Up to 20	Up to 16	Up to 49	Up to 49
8/16bit Local Bus	-	-	✓ (x3)	-	-	✓ (x3)	✓ (x3)

Revision History

Revision	Date	Summary of Changes
1.8	2009-11-23 ECN-TD-09-11002	✓ Revise STR8182X/CNS2182X pin out definition ✓ Add CNS2135 product information ✓ Revise PQFP-128 package θ_{JA}
1.7	2008-11-19	✓ Terminate STR8182/CNS2182 and replace with STR8182X/CNS2182X at 1/31, 2009 ✓ Update part order number list ✓ Add new package marking infomration ✓ Add power regulator and PLL voltage specificaiton
1.6	2008-08-21	✓ Add USB 2.0 Device Support in CNS2182/STR8182 ✓ Revise CNS2182/STR8182 pin assignment ✓ Add USB 2.0 Device - DDP,DDM and RREF support for CNS2182/STR8182 ✓ Add CNS2182/STR8182 Package Outline and Dimensions information ✓ REF_32768 clock input signal level is between 0~1.8V, but not 0~3.3V.
1.5	2008-07-15 ECN-TD-08-07004	✓ Change bit definition of bit29 – 23 of GPIO_B Pin Enable Register ✓ Add maximum power consumption and typical power consumption of CPU running at 200MHz at Section 4.4.
1.4	2008-04-09	✓ Change pin name of VDD_BAT to CVDD in CNS2131/STR8131 and CNS2132/STR8132 package diagrams.
1.3	2008-03-25	✓ Change pin name of RTC_XI to REF_32768 in CNS2131/STR8131 and CNS2132/STR8132 package diagrams. ✓ Change register description of DRAM_CFG.CASL ✓ Add embedded FE PHY, LED mode selection description in Page Selection Register ✓ Add CPU Core Clock 200MHz in Part Order and Package Marking
1.2	2008-01-28 ECN-TD-08-01004	✓ Change description of bit 7 “NICCLK_Sel” of System Clock Control Register of Clock and Power Management block. ✓ Add description of GEC TX/RX Descriptors in Section 1.10.10. ✓ Update memory re-map description in Section 3.9.1 ✓ Add 5V Tolerant information in GPIO and Pin descriptions. ✓ Add embedded FE PHY register description in Section 3.24
1.1	2007-09-28	✓ Add baud-rate description at prescaler register of UART ✓ Update HCLK_Sel to RTC_Sel (bit 6 of System clock control register) ✓ Add PLLx2250 control register recommended setting ✓ Update GPIO available number vs. part number ✓ Chip of version AD support I2S full duplex transmitting/receiving. ✓ Add GDMA and HSDMA function description ✓ Add part order numbering and package marking information ✓ Update PQFP-128 package θ_{JA}(with heat spreader) ✓ Update Machine Mode ESD protection to 200V
1.0	2007-06-15	✓ Update SMC register setting and AC timing due to ECO ✓ Add SWAITn AC timing

		✓ Remove PCMCIA description in data sheet ✓ Remove RTC battery description and update RTC clock input pin description. ✓ FE PHY LED configuration register description supplement ✓ MDC_DIV default value change to 0x2 ✓ MDC/MDIO IO power of 8181/8182 is 2.5V if RGMII interface is used. ✓ Remove 8182 USB device feature support ✓ Revise power consumption table ✓ Revise AC timing tables
0.9b	2007-04-15	✓ Update Interrupt bit[22] to FNQF (From NIC Queue Full) ✓ Update CNS2133/STR8133 ball assignment ✓ Update SPI timing ✓ Specify power voltage in Table 8 (Pin Assignment)
0.9a	2007-01-26	✓ Update AC timing value.
0.9	2007-01-14	✓ Preliminary Release (including register set and AC timing)
0.8	2006-12-18 ECN-TD-06-12001	✓ Preliminary Release (not including register set)
0.2	2006-11-28	✓ Preliminary Draft for internal review

Contents

List of Figures

List of Tables

1 Functional Descriptions

1.1 Overview of SoC Architecture

Equuleus series SoC, powered with ARM922 core, provides feature-rich interface for high-performance network access for consumer applications/equipment. The SoC architecture is illustrated in Figure 2, where major peripherals are shown—DDRC/SDRC(DDR/SDR Memory Controller), Static Memory Controller(SMC), Vector Interrupt Controller(VIC), PCI Host bridge, 10/100/1000M MAC with embedded 10/100M PHY, IDE Controller, USB2.0/1.1 Host Controller and PHY(2-port), USB2.0/1.1 Device Controller and PHY(1-port), Generic DMA Controller(GDMA), High-speed DMA Controller(HSDMA), UART, Timer, Watch Dog Timer(WDT), Real-time Clock(RTC), Serial Peripheral Interface(SPI), Pulse-coded Modulation(PCM)/Time-Division Multiplexing(TDM) interfaces for external voice CODEC's, I2S interface, Two-Wire Interface(TWI), UART, GPIO, and miscellaneous logics.

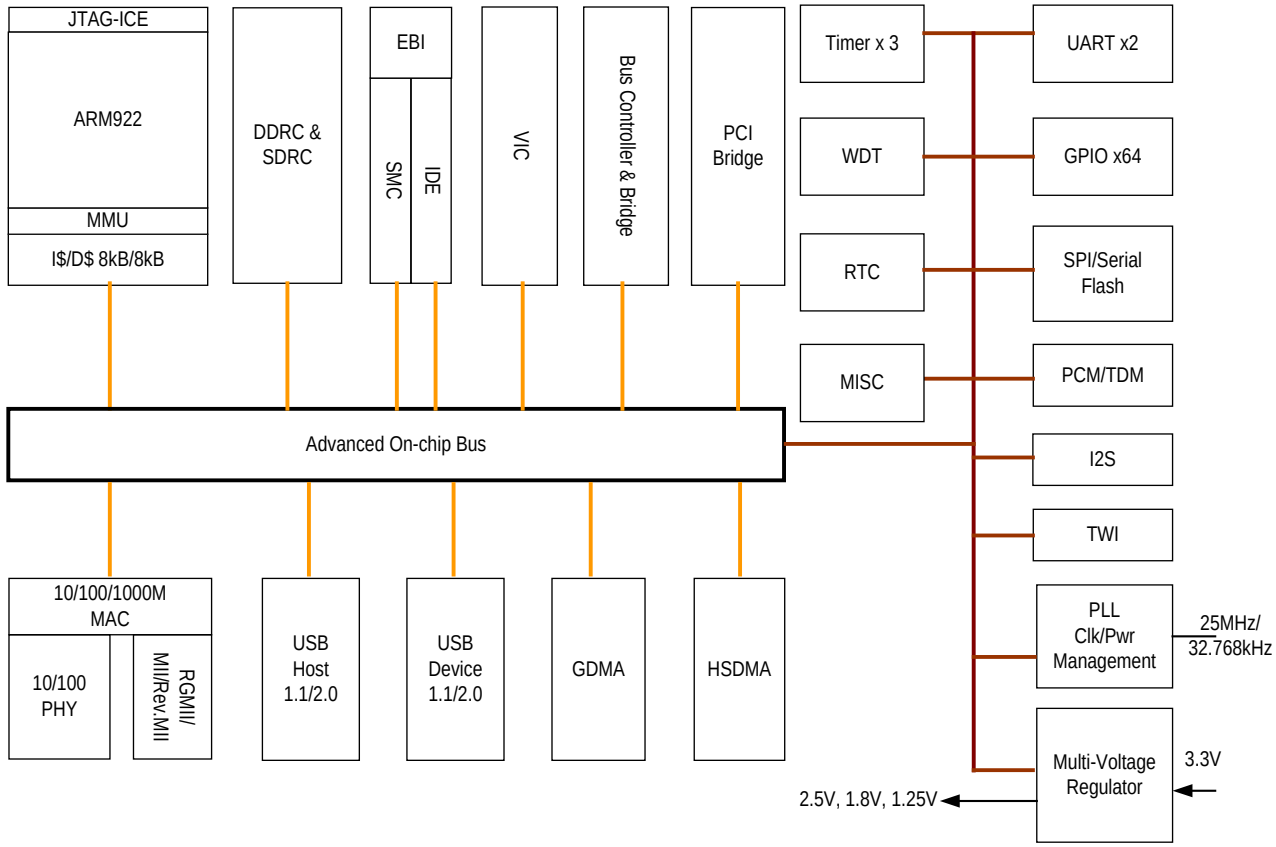


Figure 2. System Block Diagram of Equuleus SoC

1.2 32-bit RISC Core

ARM-922 RISC core, based on ARM V4 architecture and compliant to V4 instruction sets and register sets, is with Harvard architecture with SIX(6) pipelined stages—Fetch, Decode, Shift, Execution, Memory and Write. To enhance performance, the CPU core also contains a Branch Target Buffer(BTB) to reduce branch penalties. There are totally SEVEN(7) operation modes supported: Supervisor, System, FIQ, IRQ, Abort, User and Undefined Modes.

- Memory Management Unit(MMU) is supported for high-level RTOS support, with unified 4-way set associate TLB to improve overall CPU performance
- Embedded 2-way set associated I-cache and D-cache memory of 8kB each
- CPU core clock can be programmable up to 200MHz & 250MHz
- FIQ and IRQ interrupts
- Little- and Big-endian ordering
- Two ICE debugger interfaces—a simplified ARM JTAG interface and fully-compliant ARM Multi-ICE debug interfaces, compatible to *AXD*™ or *RealView*™ commercial available debuggers
- Support Burst Access at non-cacheable region and write-through regions

1.3 System Bus Architecture

A high-performance on-chip system bus is included, featured with multiple high-speed peripherals data transferring. The bus clock can be programmable with max clock frequency up to 125MHz. A dedicated low-speed bus is used for low bandwidth peripherals for lower power consumption.

1.4 DDR/SDR SDRAM Controller (DDRC/SDRC)

The STR813X/CNS213X, STR818X/CNS218X SDRAM controller provides the following features:

- Support type of SDRAM either:
  - SDR SDRAM: memory I/O are powered with 3.3V
  - DDR SDRAM: memory I/O are powered with 2.5V
- Support 16-bit memory data bus (not support 32-bit data bus)
- Tailored design for multiple channel access with high efficiency/low latency
- Maximum Addressing Space of 256M bytes
- CAS Latency
  - SDR SDRAM: 2.0 or 3.0 cycles
  - DDR SDRAM: 2.0, 2.5 or 3.0 cycles
- Supports pre-READ and post-WRITE features, with built-in buffers of two cache-line size to enhance memory bandwidth and reduce transaction latency

- Support SDRAM Self-Refresh mode when CPU entering Sleep Mode for power saving

1.5 Static Memory Controller (SMC)

The SMC supports 8/16-bit programmable external bus-width with address range up to 16M bytes. Totally 4 banks of external static memory can be accessed—one for parallel flash (Bank 0) and the other three (Bank 1~3) for asynchronous SRAM banks.

The WRITE accessing is featured with zero wait-state. For interfacing to external slow devices, wait-state control via WAITn signal can be employed to simply the interface configuration.

For Bank-1/-2/-3 memory accessing, the chip allows flexible settings of timing parameters through registers for required Setup Time, Hold Time, Read Access, Write Access and Turn-Around Time.

The SMC also supports 8/16/32-bit burst command from Generic DMA(GDMA) controller to access the external SRAM devices or DSP's. This can drastically enhance the performance for data transferring between external DSP and host network processor.

1.6 Generic DMA Engine (GDMA) and High Speed DMA Engine(HSDMA)

Equuleus has a built-in, configurable, eight-channel Generic DMA controller (GDMA) for memory-to-memory, memory-to-peripheral, peripheral-to-memory, and peripheral-to-peripheral data transferring with a shared buffer.

Furthermore, it also has a built-in, single-channel High Speed DMA controller (HSDMA) for memory-to-memory, memory-to-peripheral, and peripheral-to-memory data transferring, especially for peripheral with higher throughput requirement, such as USB device.

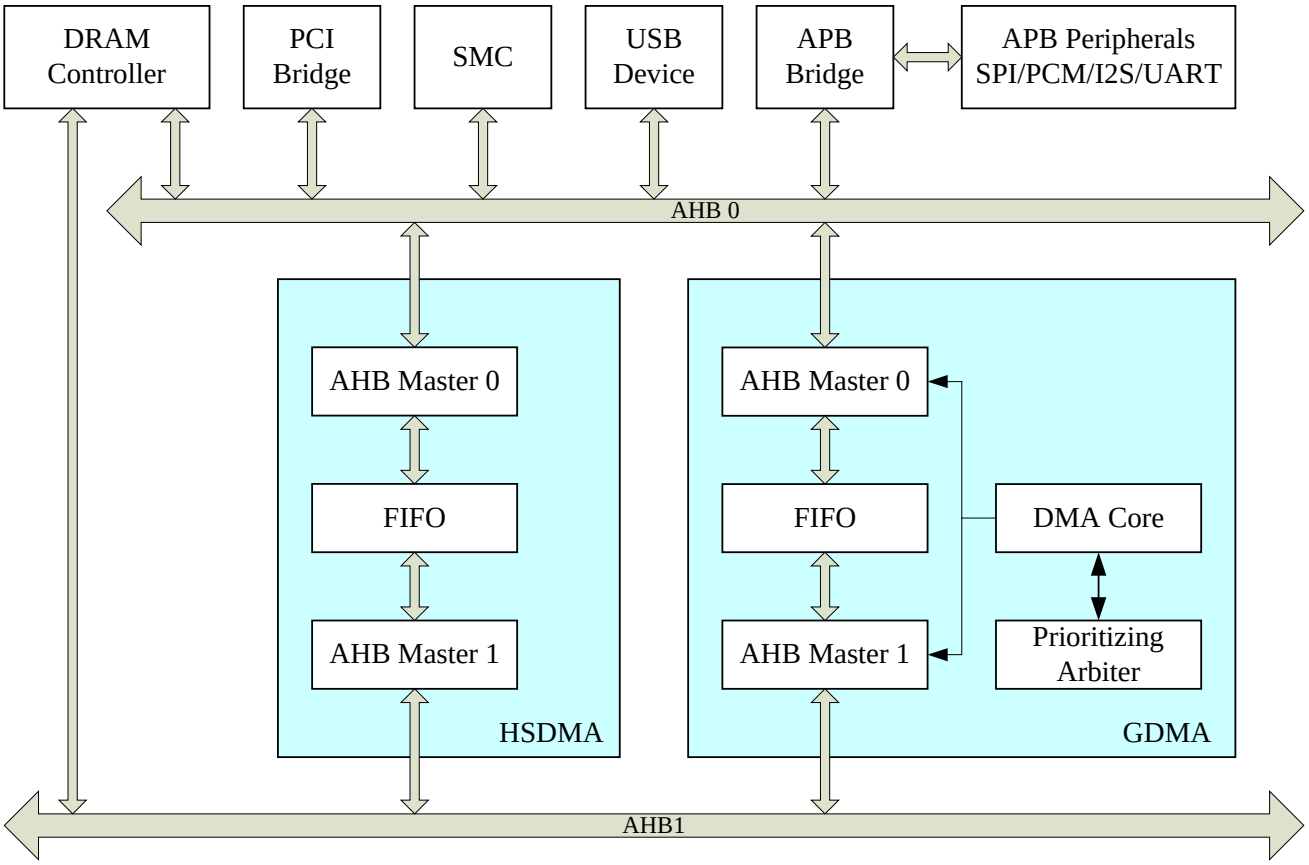


Figure 3. GDMA and HSDMA Block Diagram

1.6.1 GMDA

The GDMA engine supports four priority levels and Group Round Robin (GRR) arbitration scheme among 8 channels for 8-, 16-, and 32-bit data width transactions.

It also supports per-channel hardware handshaking for relevant peripherals to reduce the processor intervention and enhance system performance. Totally 8 pairs of handshakes are defined for PCM, SPI, I2S and UART ports, as flowing:

- PCM-TX-0 (for write data window [31:0])
- PCM-TX-1 (for write data window [63:32])
- PCM-RX-0 (for read data window [31:0])
- PCM-RX-1 (for read data window [63:32])
- SPI-TX
- SPI-RX
- I2S-TX-Left
- I2S-TX-Right
- I2S-RX-Left
- I2S-RX-Right

- UART-TX x2
- UART-RX x2

The GDMA is also featured with per-channel chain transfer (channel-0, channel-1, through to channel-7) with Linked List Descriptors resident in system memory. This can offer plenty of flexibility for data movement with least CPU loading. For each DMA access, the Link List Descriptors are fetched and the relevant control fields of descriptors are copied to the per-channel control registers. The address map for the GDMA descriptor and the structure of Linked List Descriptors are described in the following figure and tables. The base address is Cn_LLP[31:2].

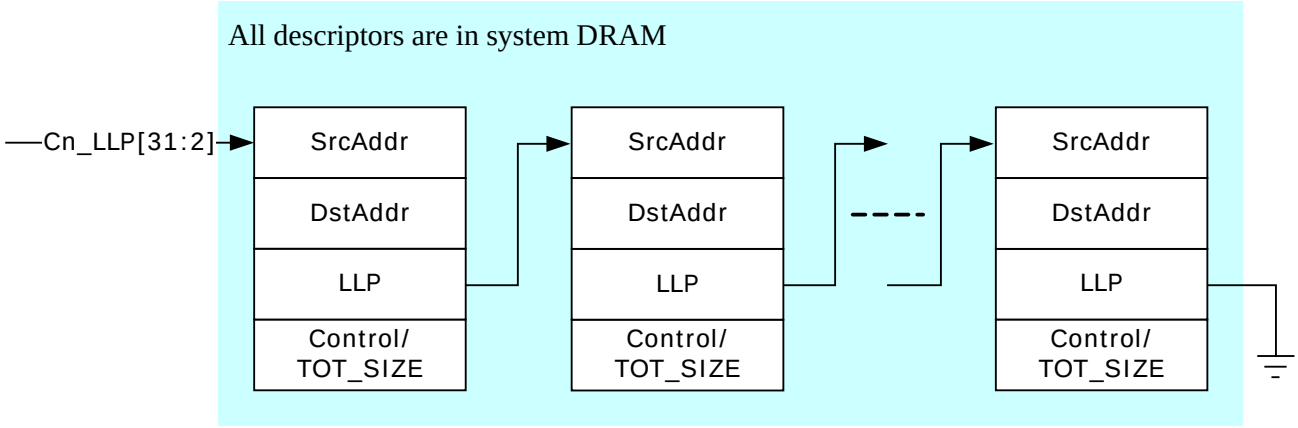


Figure 4. Structure of GDMA Link List Descriptor

Table 2. Address Map for GDMA Link List Descriptor

Name	Offset	Width	Description
SrcAddr	+0	32	Source Address
DstAddr	+4	32	Destination Address
LLP	+8	32	Linked List Pointer
Control/TOT_SIZE	+C	32	Control and Total Transfer Size

Table 3. Control field definition in GDMA Link List Descriptor

Bits	Name	Description
28	TC_MASK	Channel terminal count status mask (Same as TC_MSK in **Cn_CSR**)
27:25	SRC_WIDTH	Source transfer width (Same as SRC_WIDTH in **Cn_CSR**)
24:22	DST_WIDTH	Destination transfer width (Same as DST_WIDTH in **Cn_CSR**)
21:20	SRCAD_CTL	Source address control (Same as SRCAD_CTL in **Cn_CSR**)
19:18	DSTAD_CTL	Destination address control (Same as DSTAD_CTL in **Cn_CSR**)
17	SRC_SEL	Source selection (Same as SRC_SEL in **Cn_CSR**)
16	DST_SEL	Destination selection (Same as DST_SEL in **Cn_CSR**)
11:0	TOT_SIZE	Total transfer size (Same as TOT_SIZE in **Cn_SIZE**)

1.6.2 HSDMA

The HSDMA only supports 32-bit data width transaction. It is featured with chain transfer with Linked List Descriptors resident in registers and system memory. For each DMA access, the Linked List Descriptors are fetched. The address map for the HSDMA descriptor and the structure of Linked List Descriptors are described in the following figure. These HSDMA registers are merged in Miscellaneous block.

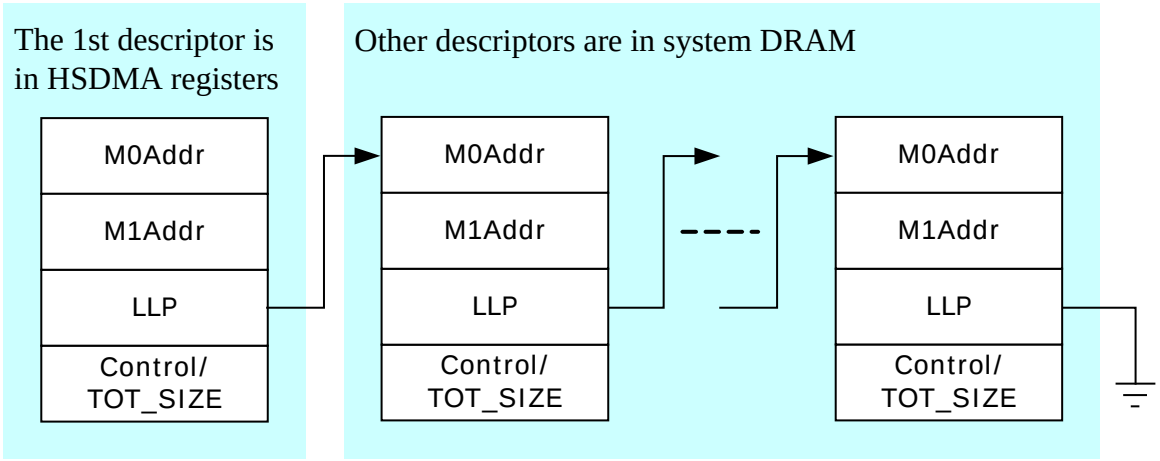


Figure 5. Structure of HSDMA Link List Descriptor

Table 4. Address Map for HSDMA Link List Descriptor

Name	Offset	Width	Description
M0Addr	+0	32	DMA Master 0 Address
M1Addr	+4	32	DMA Master 1 Address
LLP	+8	32	Linked List Pointer
Control/TOT_SIZE	+C	32	Control and Total Transfer Size

Table 5. Control field definition in HSDMA Link List Descriptor

Bits	Name	Description
29	Data_Direction	**Direction of Data Movement** 0: DMA Master 0 to DMA Master 1 1: DMA Master 1 to DMA Master 0
28	TC_MASK	**Terminal Count Interrupt Mask** 0: Pass Interrupt 1: Mask Interrupt Note: When TC_MASK of related Link List Descriptor is 1, then the TC Interrupt will be suppressed when the data movement, pointed by the descriptor, is completed.
27:24	HHS_SIZE	**Hardware Hand Shake Size Selection.** It means how many transfers in a hardware hand shake “req/ack” period. Hardware Hand Shake Size = 2^HHS_SIZE, where allowed HHS_SIZE

		= 0~10 (it means size = 1, 2, 4, …, 1K), and 11~15 are reserved.
15:0	TOT_SIZE	**Total transfer size**

1.7 USB 2.0/1.1 Host Controller with Integrated PHY's

The embedded USB host controller consists of an USB1.1 Host Controller (OHCI), an USB2.0 Host Controller (EHCI), and TWO embedded USB1.1/2.0 PHYs. The overall architecture is illustrated in following figure.

The USB1.1 Host Controller supports all of full speed (12Mbps) and low speed (1.5Mbps) devices, which are compliance with USB1.1 Specification. It embeds a 64-bytes FIFO and supports *Control Transfer, Bulk Transfer, Interrupt Transfer*, and *Isochronous Transfer* and can connect up to 127 devices at the same time.

The USB2.0 Host Controller supports high speed (480Mbps) devices, which are compliance with USB 2.0 Specification. TWO 1K-bytes FIFO's are embedded, one for TX and one for RX respectively, and supports *Control Transfer, Bulk Transfer, Interrupt Transfer*, and *Isochronous Transfer* and can connect up to 127 devices at the same time.

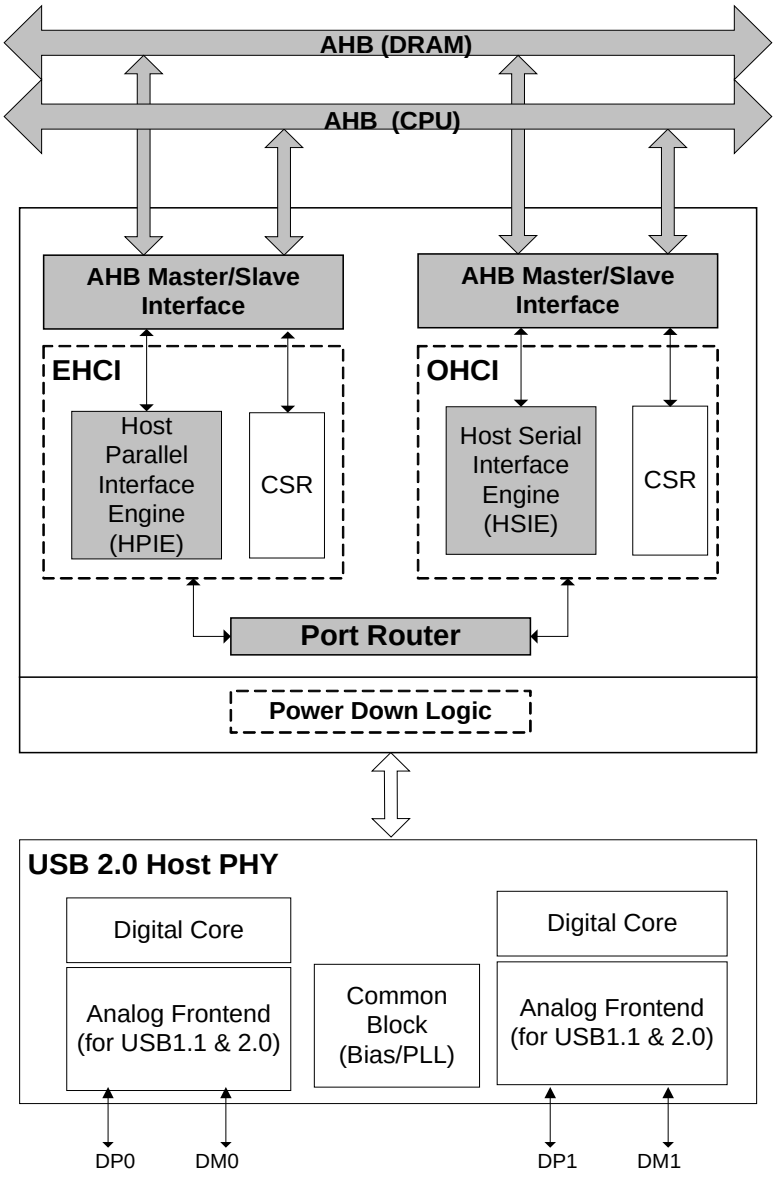


Figure 6. USB2.0/1.1 Host Controller with Integrated PHY's Block Diagram

1.7.1 AHB Interface and DMA

There are a dedicated AHB Master Interface and AHB Slave Interface for each EHCI controller and OHCI controller. An AHB Master Interface and related DMA controller co-work to do DMA of frame data and frame descriptor between external DRAM and the USB Host Controller. And AHB Slave Interface is for the processor to configure the USB Host Controller

1.7.2 Host Parallel Interface Engine (HPIE) or Host Serial Interface Engine (HSIE)

The embedded USB host controller contains engines of HPIE and HSIE for USB2.0 and 1.1 respectively.

The HPIE (HSIE) is responsible for managing all transactions to the USB. It controls the

bus protocol, packet generation/extraction, data parallel-to-serial conversion, CRC coding, bit stuffing, and NRZI encoding.

1.7.3 Port Router

The USB Host controller comprises one high-speed host controller, which implements the EHCI programming interface and one OHCI host controller. This configuration is used to deliver the required full USB 2.0-defined port capability; e.g. Low-, Full-, and High-speed capability for each port.

There exists one transceiver per physical port and each host controller module has its own port status and control registers. The EHCI controller and OHCI controller have individual port status and control registers for each port. Either EHCI or OHCI host controller can control each physical transceiver. Routing logic lies between the transceiver and the port status and control registers. The port routing logic is controlled from signals originating in the EHCI host controller. The EHCI host controller has a *global* routing policy control field and per-port *ownership* control fields. The *Configured Flag* (*CF*) bit (defined in OP register of EHCI) is the global routing policy control. At power-on or reset, the default routing policy is to the OHCI controllers. In general, when the EHCI owns the ports, the OHCI host controller's port registers do not see a connect indication from the transceiver. Similarly, when the OHCI host controller owns a port, the EHCI controller's port registers do not see a connect indication from the transceiver.

1.8 USB2.0 Device Controller with Integrated PHY

STR813X/CNS213X, STR818X/CNS218X includes a USB2.0-compliant Device Controller with integrated USB2.0 PHY for USB slave applications, with controller architecture shown in following figure.

The following features are supported:

- USB2.0 transceiver complies to 1.5Mbps(LS), 12Mbps(FS) and 480Mbps(HS) data rates.
- DP/DM of transceiver analog ports support 5V tolerance.
- Support 8 instances of 512-byte embedded SRAM for BULK/ISO transfer type endpoints.
- Support 2 instances of 64-byte embedded SRAM for Interrupt transfer type endpoints.
- Support a dedicated control transfer type endpoint, and 8 extra endpoints, allowing users to configure them as BULK-IN, BULK-OUT, ISO-IN, ISO-OUT, or Interrupt endpoints.
- Support a dedicated High Speed DMA between USB device controller and system DRAM to provide full-line speed throughput between USB device controller and DRAM.
- Support a dedicated DMA engine between USB device controller and IDE Host controller to provide wire speed through between USB device controller and external hard disk.

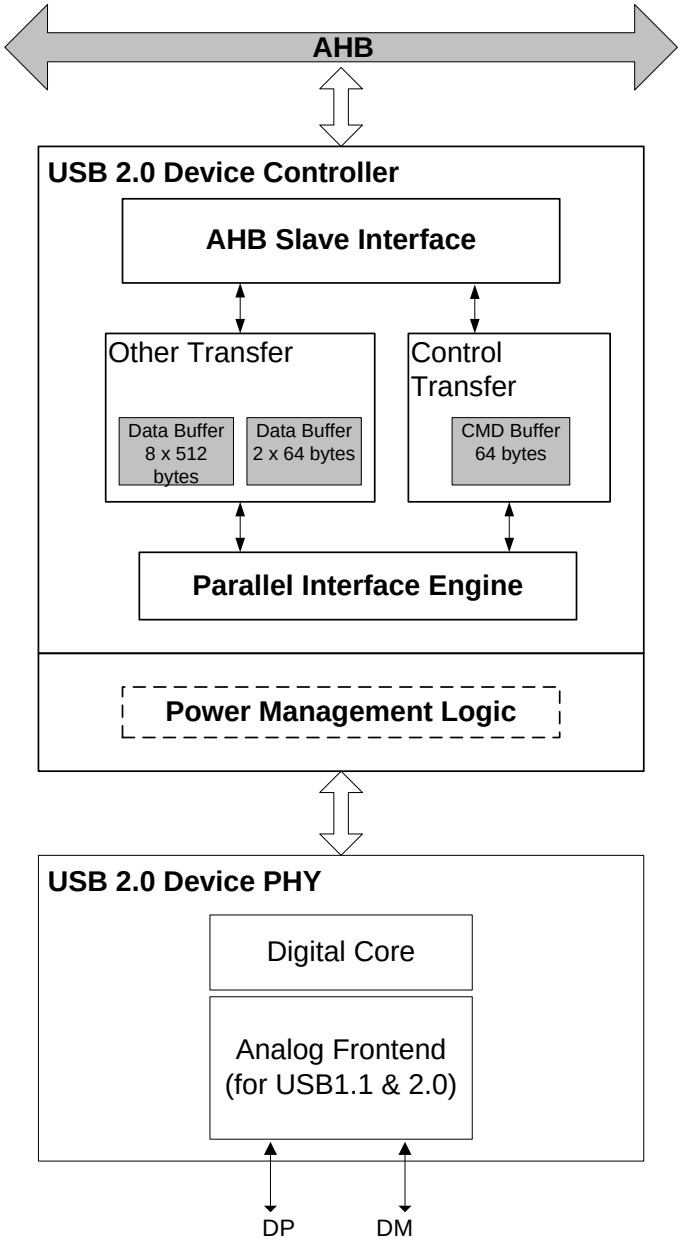


Figure 7. Block Diagram of USB2.0 Device Controller

1.9 IDE Host Controller

ATA(IDE) Host Controller in STR813X/CNS213X, STR818X/CNS218X supports the following features:

- Support up to TWO(x2) IDE Hard Disk & ATAPI devices
- Support an embedded fast path between IDE Host Controller and USB 2.0 device Controller for fast data copies between IDE device and USB Host
- Support PIO (Programmed I/O) mode of 0, 1, 2, 3 and 4, where processor is in control of the data transfer.

- Support Multiword DMA mode of 0, 1 and 2, with data transferring rates up to 16M bytes/sec
- Support Ultra DMA mode of 0, 1, 2, 3, 4, and 5, with data transferring rates up to 100M bytes/sec
- Support Ultra DMA burst CRC calculation by hardware
- DMA engine supports sophisticate cache-line alignment for efficient DRAM access
- An embedded 8x32-bit buffer is employed for optimal transferring handling.

The IDE host controller in STR813X/CNS213X, STR818X/CNS218X can be programmed to operate with ATA protocols using respective defined timing, as specified in ANSI ATA-6 specification.

1.10 Gigabit Ethernet Controller with Embedded 10/100M PHY(GEC)

1.10.1 Overview

The embedded GbE Controller incorporates the following features:

- Support a 10/100/1000M MAC port exposed with external programmable RGMII/MII/Reverse MII interface, or with embedded 10/100M PHY
- Support maximum packet length up to 1536 bytes
- Support 802.3x full duplex flow control and half duplex backpressure flow control
- Auto strip extra one more nibble at the end of RX packet.
- TX/RX FIFO's
  - TX_FIFO = 2KB to accommodate at least a long packet to implement minimum IPG and packet data are purged immediately after the first 64 bytes are transmitted successfully.
  - RX_FIFO = 4KB to guarantee no packet drop, given flow control is enabled.
- Support sophisticate scatter-gather descriptor ring
- When run out of RX/TX descriptors, the RX/TX DMA engines will activate an internal timer, and periodically check the availability of new descriptors, but interrupt only at 1st time.
- Support MII internal and external loop back test modes
- Support packet and byte counting for transmitted, received or dropped packets
- Support 512-bit ARL hash table
- Support TX/RX TCP/UDP/IP checksum offload
- Support CRC strip/generation by hardwire
- Support VLAN Tag strip/attach by hardwire
- Support Wake–on-LAN by scanning incoming Magic Packets.

STR813X/CNS213X, STR818X/CNS218X integrates a Giga Ethernet controller with one

10/100/1000 Mbps RGMII/MII/Reverse MII port and an embedded 10/100 Ethernet PHY. It can supports 10/100M application with embedded PHY, or exposed RGMII interface to an external G-PHY. It is compliant to specifications of IEEE 802.1Q Virtual LANs (VLAN), IEEE 802.3u 100Base-T, IEEE 802.3ab 1000 Mbps (Gigabit Ethernet) and IEEE 802.3x Flow Control. The GEC is VLAN-aware and able to identify L3 and L4 packets and with L3/L4 check-sum off-load.

Two SRAM macros are embedded in GEC: one is 1024x33-bit for RX FIFO and the other is 512x33 bit for TX FIFO. Packets are received by a reliable cut-and-through scheme. And when packets are transmitted by the controller, it is based on a reliable store-and-forward scheme.

1.10.2 10/100M Ethernet PHY

The embedded Fast Ethernet PHY supports standard PHY registers for GEC to control it through Management Interface (MDC/MDIO). These registers are described in Section 3.24. And the followings are the PHY supported features:

- Compliant to IEEE 802.3u clause 28 ; 1.8V operation with 3.3V IO signal tolerance
- Support 10/100Mbps operation modes
- Support full- and half-duplex modes
- Support Auto-negotiation
- Support power-down mode
- Support 10M power-saving mode
- Support DC baseline wandering compensation
- Adaptive equalization for various CAT-5 cable length compensation(up to 120 meters)
- Network LED status
- VCT (Virtual Cable Test) function support

1.10.3 Packet Format

The GEC can support Enthernet/802.3 format packet with packet length up to 1536 bytes. The acceptable packet length is controllable. However, those oversize packets can also be forwarded to host memory by setting a register bit.

Each packet transmitted from the MAC port must meet the requirement packet length of 64-byte. And for VLAN-tagged packets, 68-byte is the minimum packet length.

1.10.4 MAC Address Filtering

The incoming packet will be deposited in the RX FIFO, and concurrently the GEC performs DA MAC address filtering according to the configuration.
It supports reserved multicast address filtering to filter special reserved multicast address packets. It also supports promiscuous mode to bypass MAC DA matching check and receive all of packets. Furthermore, it also supports to receive My_MAC and Broadcast-only packets or packets that hit MAC hash table. The MAC hash table is resident in a 512-bit memory. Two kinds of hash algorithms, Direct or 32 bit CRC hash,

are supported. The MAC hash table is software accessible and can auto-learn SA of packets transmitted from CPU.

1.10.5 VLAN

The GEC supports 802.1q tag-based VLAN ingress check, and it can support up to 4 VLANs, set in registers, where these VLAN IDs can be any of the 4K VLAN space. Internally, the controller uses 4 bits of “My VLAN ID Control Register” to enable VLAN ingress check for each pre-defined VLAN ID. When at least one of the pre-defined VLAN ID is enabled, RX MAC will compare the pre-defined VLAN ID with the tagged VID of the received packet. If one of them is matched, the packet will be received, but it will be dropped if unmatched, and the relevant MIB counter will be increased by 1 accordingly.

When a received packet is VLAN-tagged, the tag can be stripped from the packet or retained with the packet. No matter VLAN tag is stripped or not, the VLAN tag information will be stamped at RX DMA Descriptor.

When a received packet is un-tagged VLAN, the packet is always forwarded to system memory. Each egress packets can be tagged or un-tagged, based on the setting of the assignment of TX DMA descriptor. If tagged, the VLAN ID (12-bits) of TX DMA descriptor will be inserted to the packet as VID of VLAN tag. If tagging/un-tagging will modify the transmitted packet, CRC will be re-generated.

1.10.6 Inter-Switch-Tag (IST)

The GEC also supports a proprietary Inter-Switch-Tagging (IST) to share necessary information with external switch controller chips. After enabling IST at this port, the MAC treat each received packet as VLAN-tagged, no matter of the value at EPID field. In normal cases, the EPID field is with the value of 0x8100 to indicate that this packet is VLAN-tagged. But at IST mode, user can replace this field with other pre-defined value and route the packet to CPU.

1.10.7 Flow Control

The 10/100/1000 MAC port in GEC fully supports IEEE 802.3x flow control at full-duplex mode. The 802.3x flow control function ON/OFF is dependent on auto-negotiation result or force-mode configuration if auto-negotiation function is OFF.

At half-duplex mode of operations, back-pressure flow control is supported. The back-pressure flow control supports 2 kinds of operation mode:

- **Force Collision Mode.** At this mode, all of incoming packets will be jammed until back pressure condition released.
- **Pass-one-every-N backpressure collision policy**. When consecutive number of JAM packet reaches the jam_no threshold, the following incoming packet will be received, not be back-pressured.

Asserting or de-asserting of flow control is also dependent on a global RX buffer threshold, which can be set through "Flow Control Configuration Register". Once above threshold is reached, flow control asserts, and then 802.3x flow control packet is sent at full-duplex mode or back-pressure at half-duplex mode.

In order to smoothing packet flow throughput, a hysteresis scheme at global threshold is also supported. When occupied RX packet buffer reach "Flow Control Assert Threshold", global threshold flag asserts. And, only when occupied RX packet buffer is less than "Flow Control De-Assert Threshold ", the global threshold flag will de-assert.

1.10.8 MIB counter

The GEC maintains a local information database for network management use, which comprises a set of the counters. They can provide the statistics about frame counters, byte counters, error log counters, etc. The statistics data can be separated into two categories: one is for RX path and the other for TX path. All packets are not double counted in these counters.

1.10.9 Power Management

A remote Wake-on-LAN (WOL, with respect to AMD Magic Packet) is supported to provide power management for most efficient power efficiency. When Wake-on-LAN is enabled, TX MAC will be powered down and RX MAC will only scan Magic Packet and not forward any packet to system memory. After detecting the Magic Packet, GEC asserts WOL interrupt to CPU and wake-up CPU accordingly.

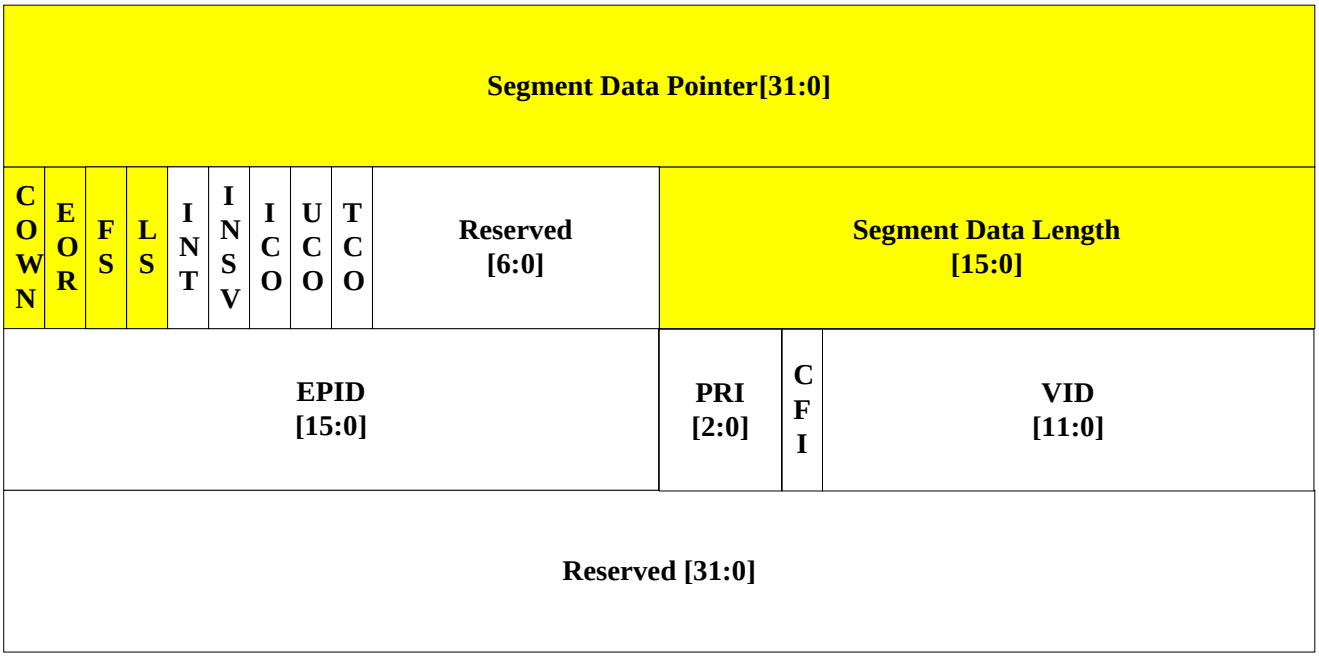
In addition to Wake-on-LAN feature, the whole GEC can be powered-down as well. During power-down mode, the Wake-on-LAN does not function and will not receive any more packets, and will flush all of the queued TX packets. Then, the clock of GEC will be gated off for power saving.

1.10.10 DMA of GEC

The DMA controller forwards packets between host memory and embedded packet memory within GEC. It implements sophisticated descriptor ring architecture, and support multiple segments for a TX/RX packet to comply modern zero-copy socket driver architecture. It also supports READ-Alignment and WRITE-Alignment for both transmit path and receive path respectively. The READ-Alignment feature enhances the DMA performance in cache-line oriented accessing, by terminating transmit DMA cycles on a cache line boundary and start the next transaction on a cache-line aligned address.

The WRITE-Alignment feature allows a packet to be stored at 2-bytes address offset from a cache line boundary at host memory. This feature meets 2-byte offset requirement in protocol stack for high-level commercial RTOS (like VxWorks) or Linux OS, and achieves zero-copy from Ethernet driver to TCP/IP protocol stack software to enhance the packet transferring efficiency.

The detailed TX/RX descriptor formats are illustrated below.



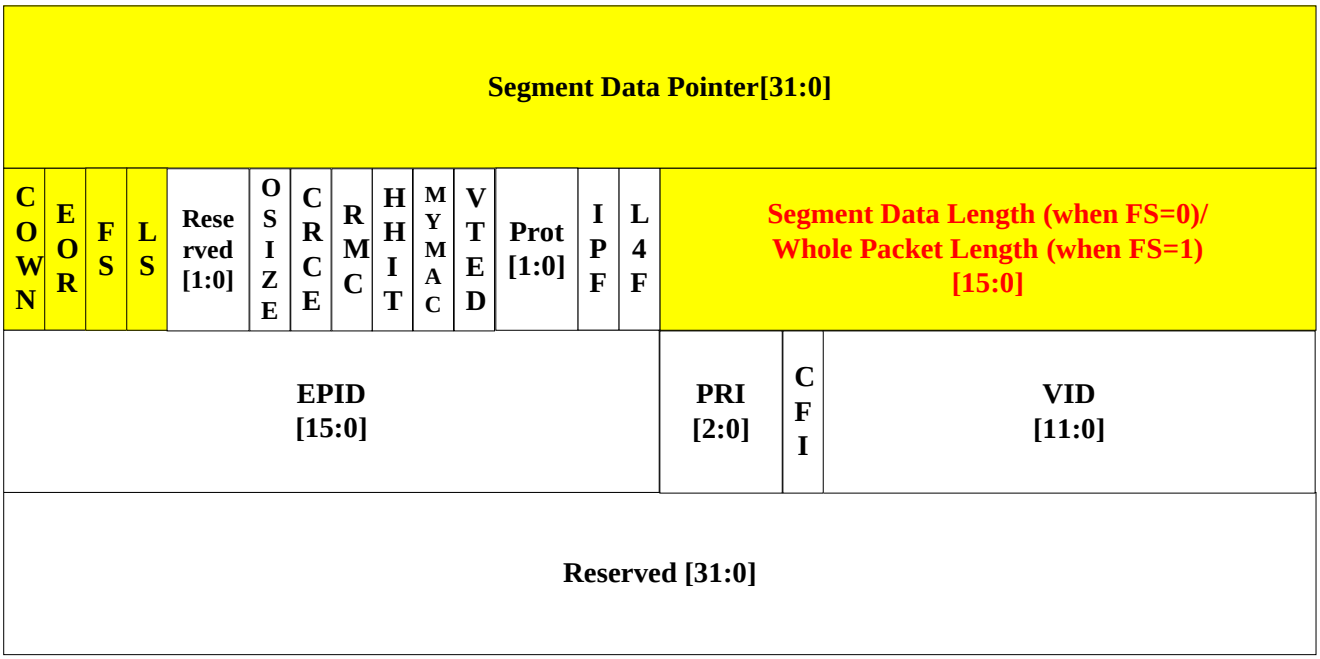
*The highlighted (shaded) fields are per Descriptor sensitive; the other fields are per Packet sensitive (meaningful only at the descriptor with FS=1)

Figure 8. TX Descriptor Format

Table 6. TX Descriptor Field Description

Offset#	Bit#	Symbol	Descriptions
0x0	31:0	SDP	Segment data pointer: point to the starting address of this transmitted data segment. The pointer is allowed to be only byte alignment.
0x4	31	COWN	CPU Ownership: This bit, when set, indicates that the descriptor owned by CPU. When cleared, it indicates that the descriptor own by the DMA. The DMA sets this bit when the relative segment data is transmitted and return it to the CPU.
0x4	30	EOR	End of descriptor ring: This bit, when set, indicates that this is the last descriptor in the descriptor ring. When DMA's internal transmit pointer reaches here, the pointer will return to the first descriptor (TX_DES_BASE, reg. 0x110) of the descriptor ring
0x4	29	FS	First Segment descriptor: This bit, when set, indicates that this is the first descriptor of a TX packet, and that this descriptor is pointing to the first segment of the packet.
0x4	28	LS	Last Segment descriptor: This bit, when set, indicates that this is the last descriptor of a TX packet, and that this descriptor is pointing to the last segment of the packet
0x4	27	INT	Interrupt: When set, DMA will generate an interrupt (txtc_int) after sending out this packet (not this segment only).
0x4	26	INSV	Insert VLAN Tag in the following word.
0x4	25	ICO	Enable IP checksum generation offload
0x4	24	UCO	Enable UDP checksum generation offload
0x4	23	TCO	Enable TCP checksum generation offload
0x4	22:16		Reserved
0x4	15:0	SDL	Segment Data length: indicate the length of this transmitted segment in bytes
0x8	31:16	EPID	VLAN Tag EPID

0x8	15:13	PRI	VLAN Tag Priority
0x8	12	CFI	VLAN Tag CFI (Canonical Format Indicator)
0x8	11:0	VID	VLAN Tag VID
0xC	31:0		Reserved



*The highlighted (shaded) fields are per Descriptor sensitive; the other fields are per Packet sensitive (meaningful only at the descriptor with FS=1)

Figure 9. RX Descriptor Format

Table 7. RX Descriptor Field Description

Offset#	Bit#	Symbol	Descriptions
0x0	31:0	SDP	Segment data pointer: point to the starting address of this received data segment. The pointer must be 4-word cache line alignment or offset 2 bytes from the cache line boundary.
0x4	31	COWN	CPU Ownership: This bit, when set, indicates that the descriptor owned by the CPU. When cleared, it indicates that the descriptor own by the DMA. The DMA sets this bit when the relative segment data is received.
0x4	30	EOR	End of descriptor ring: This bit, when set, indicates that this is the last descriptor in the descriptor ring. When DMA's internal receive pointer reaches here, the pointer will return to the first descriptor (RX_Des_BASE) of the descriptor ring
0x4	29	FS	First Segment descriptor: This bit, when set, indicates that this is the first descriptor of a RX packet, and that this descriptor is pointing to the first segment of the packet. CPU should reset this bit when it allocates this descriptor.
0x4	28	LS	Last Segment descriptor: This bit, when set, indicates that this is the last descriptor of a RX packet, and that this descriptor is pointing to the last segment of the packet CPU should reset this bit when it allocates this descriptor

0x4	27:26		Reserved
0x4	25	OSIZE	The received packet is oversize.
0x4	24	CRCE	The RX packet is CRC Error Note: Only when Acpt_CRC_Err = 1 of MAC Configuration Register, checksum error packet will be received to CPU.
0x4	23	RMC	The RX packet DA is Reserved Multicast Address
0x4	22	HHIT	The RX packet DA hits hash table
0x4	21	MYMAC	The RX packet DA is My_MAC
0x4	20	VTED	VLAN Tagged in the following word.
0x4	19:18	Prot	Protocol: 2'b00: (IPV4H5 & Fragment) or (IPV4H5NF & not TCP & not UDP) (can do IP checksum) 2'b01: IPV4H5NF & UDP (can do IP/UDP checksum) 2'b10: IPV4H5NF & TCP (can do IP/TCP checksum) 2'b11: Others (no any checksum offload is done)
0x4	17	IPF	IP checksum check fail. This bit is meaningful only when Prot != 2'b11. Note: Only when Acpt_CKS_Err = 1 of MAC Configuration Register, checksum error packet will be received to CPU.
0x4	16	L4F	Layer-4 checksum fail (TCP or UDP over IP). This bit is meaningful only when Prot=2'b01(UDP) or 2'b10(TCP) Note: Only when Acpt_CKS_Err = 1 of MAC Configuration Register, checksum error packet will be received to CPU.
0x4	15:0	SDL/WPL	Segment Data Length/Whole Packet Length: indicates the length of this received segment in bytes when FS=0, or the length of this received packet when FS=1. CPU should set SDL to the allocated segment buffer length (in bytes) when it allocates the descriptor. DMA will modify this field to the actual data length it fills for the non-first segment (FS=0) or the whole packet length for the first segment (FS=1).
0x8	31:16	EPID	VLAN Tag EPID
0x8	15:13	PRI	VLAN Tag Priority
0x8	12	CFI	VLAN Tag CFI (Canonical Format Indicator)
0x8	11:0	VID	VLAN Tag VID
0xC	31:0		Reserved

1.11 PCI Host Bridge

The PCI host bridge design supports PCI Specification Revision of v2.2, and support features of PCI Power Management Spec v1.1. The bus clock can be running at clock frequencies up to 66MHz. Simultaneously up to TWO(2) external PCI Master devices can be supported.

The PCI bus I/O can be configured to comply with Cardbus Specification as well, by register setting, for Cardbus devices. No glue logics, like PCI/Cardbus bridge, is required in these cases. For Cardbus application, the bus slot can support hot insertion and remove, which is compliant to PCI Hot Plug Spec (Revision 1.0).

The host bridge is designed for interfacing the Host CPU with PCI bus and forward data access from both the upstream and downstream directions. The AHB bus and the PCI bus can operate at two different clock domains, and it has built-in multiple data buffers to achieve high-speed data posting, prevent bus deadlock.

The host bridge core allows the CPU to initialize the entire system during power-up reset using standard PCI protocol. Both type-zero and type-one configuration transactions are supported. The CPU requests configuration access on the PCI bus by writing to or reading from the CONFIG_ADDR (0xA400_0000) and CONFIG_DATA (0xA000_0000) registers.

The host bridge initiates memory or IO read and write cycles on the PCI bus upon AHB bus requests. It contains 4 write buffers, two in the AHB bus clock domain and two in the PCI clock domain, to post-write data. Data can be written from the AHB bus, at the same time, write operation is running on the PCI bus.

Reading by the AHB bus is handled as delayed-read. The AHB slave retries the CPU, while it is reading data from the PCI bus. Instead of inserting wait state while waiting for return data, the AHB slave uses AHB bus “retry” to free up the AHB bus for other accesses. Once read data is ready from the PCI bus, data return to the CPU with zero wait state in subsequent read. The primary benefit of the delayed-read method is to prevent deadlock between the PCI and AHB buses.
When accessed by an external PCI bus master, the host bridge functions as a PCI target. The PCI target contains two write-buffers and a read-buffer to handle write posting and transferring data across the two clock domains. The read/write request received from the PCI bus is forwarded upstream to the AHB bus through the built-in AHB bus master.

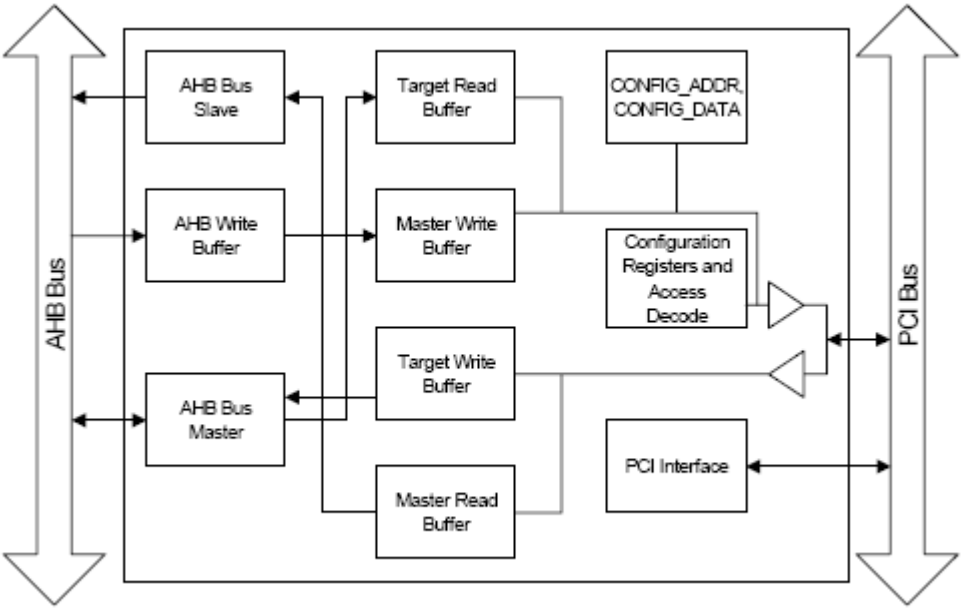


Figure 10. PCI Host Bridge

1.11.1 PCI Configuration Cycle

The PCI Host Bridge provides an access window (CONFIG_ADDR and CONFIG_DATA registers) for CPU to configure external PCI devices and the bridge itself. The configuration mechanism implemented by the host bridge is the PC-compatible standard mechanism, defined by the PCI specification as Configuration Mechanism #1. Both Type 0 and Type 1 configuration cycles are supported.

Type 0 and Type 1 configuration cycle implies that the host bridge is capable of configure

PCI agents in the same bus segment as well as PCI agents in the other side of a PCI-to-PCI bridge. In other words, the host bridge supports multiple segments PCI bus.

To initiate a PCI configuration access, the CPU is first required to write the address into the CONFIG_ADDR register. The bridge will translate the CONFIG_ADDR information into a configuration register address based on whether it is a Type 0 or Type 1 configuration access. The format of the CONFIG_ADDR register and the translation scheme of type 0 and type 1 are showed at the following 2 figures.

The CPU then read or write to the CONFIG_DATA register to initiate a configuration read or write access. The read/write access to the CONFIG_DATA triggers the host bridge to initiate PCI configuration access to the PCI bus. If it is a read, configuration data read from PCI bus is deposited to the CONFIG_DATA register and returned to the AHB bus. If it is a write access, data written into the CONFIG_DATA register is written to the PCI bus as configuration write data.

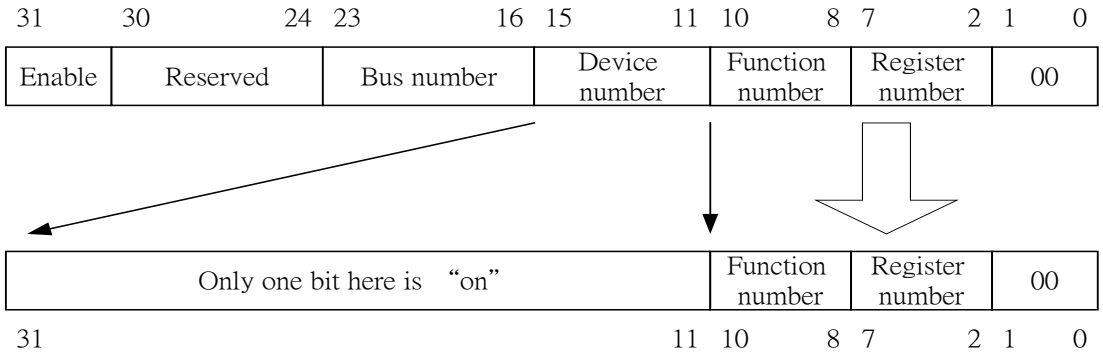


Figure 11. Type 0 Translation

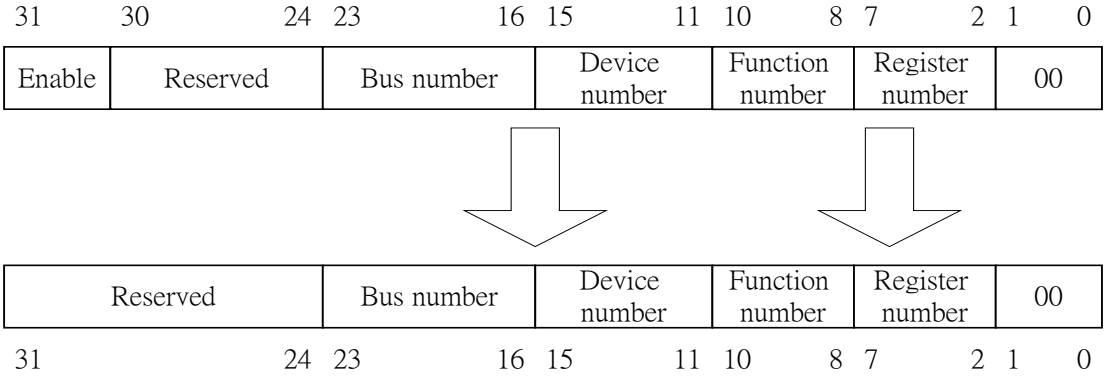


Figure 12. Type 1 Translation

1.12 High-speed UART Controller

Two ports of 16C550 UART controllers are embedded. It has two 8-bit wide FIFOs with depth of 16 for TX and RX, and supports a hardware handshake with Generic DMA to

reduce the processor interrupt interactions and enhance system performance. Its baud rate is programmable and can be up to 1.5Mbps.

1.13 General Purpose Inputs/Outputs (GPIO's)

The GPIO block provides totally up to programmable 64 I/O ports with each port being able to be independently programmed. The GPIO block is divided into GPIOA[31:0] and GPIOB[31:0]. Note that these pins are shared with other functional pins, as shown in Table 8 and Table 9 below. A mask-able interrupt is generated for each pin for a pre-defined event selectable from high- or low-level, rising- or falling-edge. They can be used as external interrupt pins as well.

1.13.1 GPIO Group-A

Table 8. GPIO Group-A Pin Assignment and Shared Pins Description

Note that the GPIOA pins enable/disable are controlled by a register of MISC.

GPIOA	Type, Internal PU or PD	Shared Pin Name	Shared Pin Description PU: Pull-Up PD: Pull-Down
GPIOA0	I/O, PU/PD	EXT_INT29	After power-on-reset, this pin is in input state for GPIO. EXT_INT29 can be enabled by setting relevant control bit. By default, internal pull-down.
GPIOA1	I/O, PU/PD	EXT_INT30	After power-on-reset, this pin is in input state for GPIO. EXT_INT30 can be enabled by setting relevant control bit. By default, internal pull-down.
GPIOA2	I/O, PU/PD	UR_ACT0	After power-on-reset, this pin is in input state for GPIO. UR_ACT0 can be enabled by setting relevant control bit. By default, internal pull-down.
GPIOA3	I/O, PU/PD	I2SDR, UR_ACT1	After power-on-reset, this pin is in input state for GPIO or I2S Data Input (I2SDR). UR_ACT1 can be enabled by setting relevant control bit. By default, internal pull-down.
GPIOA4~7	I/O, PU/PD		This pin is a dedicated GPIO pin. By default, internal pull-down.
GPIOA8~12	I/O, PU/PD		This pin is a dedicated GPIO pin. By default, internal pull-up.
GPIOA13	I/O, PU	SDA	After power-on-reset, this pin is in input state for GPIO. If TWI device is used and SDA can be enabled by setting relevant control bit.
GPIOA14	I/O, PU	SCL	After power-on-reset, this pin is in input state for GPIO. If TWI device is used and SCL can be enabled by setting relevant control bit.
GPIOA15	I/O, PD	I2SSD	After power-on-reset, this pin is in input state for GPIO. If I2S device is used and I2SSD can be enabled by setting relevant control bit.

GPIOA16	I/O, PD	I2SWS	After power-on-reset, this pin is in input state for GPIO. If I2S device is used and I2SWS can be enabled by setting relevant control bit.
GPIOA17	I/O, PD	I2SCLK	After power-on-reset, this pin is in input state for GPIO. If I2S device is used and I2SCLK can be enabled by setting relevant control bit.
GPIOA18	I/O, PD	PCMDR	After power-on-reset, this pin is in input state for GPIO. If PCM device is used and PCMDR can be enabled by setting relevant control bit.
GPIOA19	I/O, PD	PCMDT	After power-on-reset, this pin is in input state for GPIO. If PCM device is used and PCMDT can be enabled by setting relevant control bit.
GPIOA20	I/O, PD	PCMFS	After power on reset, this pin is GPIO in input state. If PCM device exists in system, set related control bit to enable PCMFS function.
GPIOA21	I/O, PD	PCMCLK	After power-on-reset, this pin is in input state for GPIO. If PCM device is used and PCMCLK can be enabled by setting relevant control bit.
GPIOA22	I/O, PU	LED0	After power-on-reset, this pin is in input state for GPIO. If Fast Ethernet PHY device is used and LED0 can be enabled by setting relevant control bit.
GPIOA23	I/O, PU	LED1	After power-on-reset, this pin is in input state for GPIO. If Fast Ethernet PHY device is used and LED1 can be enabled by setting relevant control bit.
GPIOA24	I/O, PU	LED2	After power-on-reset, this pin is in input state for GPIO. If Fast Ethernet PHY device is used and LED2 can be enabled by setting relevant control bit.
GPIOA25	I/O, PU	WDTRSTn	After power-on-reset, this pin is in input state for GPIO. If system reset is needed when Watch Dog Timer Time Out, set related control bit to enable WDTRSTn function.
GPIOA26	I/O, PD	SPIDR	After power on reset, if the system uses SPI flash booting, this pin is SPIDR always; otherwise, this pin is GPIO in input state. If SPI device exists in system, set related control bit to enable SPIDR function.
GPIOA27	I/O, PD	SPICLK	After power on reset, if the system uses SPI flash booting, this pin is SPICLK always; otherwise, this pin is GPIO in input state. If SPI device(s) exists in system, set related control bit to enable SPICLK function.
GPIOA28	I/O, PU	SPICSn[0]	After power on reset, if the system uses SPI flash booting, this pin is SPICSn[0] always; otherwise, this pin is GPIO in input state. If SPI device # 0 exists in system, set related control bit to enable SPICS0n function.
GPIOA29	I/O, PU	SPICSn[1]	After power on reset, this pin is in input state for GPIO. If SPI device # 1 is used in system, set related control bit to enable SPICS1n function.
GPIOA30	I/O, PU	SPICSn[2]	After power on reset, this pin is in input state for GPIO. If SPI device # 2 is used in system, set related control bit to enable SPICS2n function.
GPIOA31	I/O, PU	SPICSn[3]	After power on reset, this pin is in input state for GPIO. If SPI device # 3 is used in system, set related control bit to enable SPICS3n function.

1.13.2 GPIO Group B

Note that the GPIOB pins enable/disable are controlled by a register of MISC.

Table 9. GPIO Group-B Pin Assignment and Shared Pins Description

GPIOB	Type, Internal PU or PD	Shared Pin Name	Shared Pin Description PU: Pull-Up PD: Pull-Down
GPIOB0	I/O, PU, 5V Tolerant	MDC	After power on reset, this pin is in input state for GPIO. If the PHY management interface is used, set related control register bit to enable MDC function.
GPIOB1	I/O, PU, 5V Tolerant	MDIO	After power on reset, this pin is in input state for GPIO. If the PHY management interface is used, set related control register bit to enable MDIO function.
GPIOB2	I/O, PD	COL	After power on reset, this pin is in input state for GPIO. If the COL pin of MII or reverse MII interface will be used, set related control register bit to enable COL function.
GPIOB3	I/O, PU, 5V Tolerant	IORDY	After power on reset, this pin is in input state for GPIO. If IDE device is used in system, set related control register bit to enable IORDY function.
GPIOB4	I/O, PD, 5V Tolerant	DMARQ	After power on reset, this pin is in input state for GPIO. If IDE device is used in system, set related control register bit to enable DMARQ function.
GPIOB5	I/O, PU, 5V Tolerant	DMACKn	After power on reset, this pin is in input state for GPIO. If IDE device is used in system, set related control register bit to enable DMACKn function.
GPIOB6	I/O, PD, 5V Tolerant	INTRQ	After power on reset, this pin is in input state for GPIO. If IDE device is used in system, set related control register bit to enable INTRQ function.
GPIOB7	I/O, PU, 5V Tolerant	IDECS0n	After power on reset, this pin is in input state for GPIO. If IDE device is used in system, set related control register bit to enable IDECS0n function.
GPIOB8	I/O, PU, 5V Tolerant	IDECS1n	After power on reset, this pin is in input state for GPIO. If IDE device is used in system, set related control register bit to enable IDECS1n function.
GPIOB9	I/O, PU, 5V Tolerant	DIORn	After power on reset, this pin is in input state for GPIO. If IDE device is used in system, set related control register bit to enable DIORn function.
GPIOB10	I/O, PU, 5V Tolerant	DIOWn	After power on reset, this pin is in input state for GPIO. If IDE device is used in system, set related control register bit to enable DIOWn function.
GPIOB11	I/O, PU, 5V Tolerant	SCE1n	After power on reset, this pin is in input state for GPIO. If SRAM device # 1 is used in system, set related control bit to enable SCE1n function.
GPIOB12	I/O, PU, 5V Tolerant	SCE2n	After power on reset, this pin is in input state for GPIO. If SRAM device # 2 is used in system, set related control

			bit to enable SCE1n function.
GPIOB13	I/O, PU, 5V Tolerant	SCE3n	After power on reset, this pin is in input state for GPIO. If SRAM device # 3 is used in system, set related control bit to enable SCE1n function.
GPIOB14	I/O, PU, 5V Tolerant	SWAIT1n	After power on reset, this pin is in input state for GPIO. If SRAM device # 1 needs WAIT hand shake function, set related control bit to enable SWAIT1n function.
GPIOB15	I/O, PU, 5V Tolerant	SWAIT2n	After power on reset, this pin is in input state for GPIO. If SRAM device # 2 needs WAIT hand shake function, set related control bit to enable SWAIT2n function.
GPIOB16	I/O, PU, 5V Tolerant	SWAIT3n	After power on reset, this pin is in input state for GPIO. If SRAM device # 3 needs WAIT hand shake function, set related control bit to enable SWAIT3n function.
GPIOB17	I/O, PU, 5V Tolerant	-	Reserved
GPIOB18	I/O, PU, 5V Tolerant	-	Reserved
GPIOB19	I/O, PU, 5V Tolerant	-	Reserved
GPIOB20	I/O, PU, 5V Tolerant	-	Reserved
GPIOB21	I/O, PU	UR_TXD[1]	After power on reset, this pin is in input state for GPIO. If UART device #1 is used in system, set related control register bit to enable UR_TXD[1] function.
GPIOB22	I/O, PU	UR_RXD[1]	After power on reset, this pin is in input state for GPIO. If UART device #1 is used in system, set related control register bit to enable UR_RXD[1] function.
GPIOB23	I/O, PU, 5V Tolerant	TRDYn	After power on reset, this pin is in input state for GPIO. If PCI device is used in system, set related control register bit to enable TRDYn function.
GPIOB24	I/O, PU, 5V Tolerant	DEVSELn	After power on reset, this pin is in input state for GPIO. If PCI device is used in system, set related control register bit to enable DEVSELn function.
GPIOB25	I/O, PU, 5V Tolerant	STOPn	After power on reset, this pin is in input state for GPIO. If PCI device is used in system, set related control register bit to enable STOPn function.
GPIOB26	I/O, PU, 5V Tolerant	PERRn	After power on reset, this pin is in input state for GPIO. If PCI device is used in system, set related control register bit to enable PERRn function.
GPIOB27	I/O, PU, 5V Tolerant	SERRn	After power on reset, this pin is in input state for GPIO. If PCI device is used in system, set related control register bit to enable SERRn function.
GPIOB28	I/O, PU, 5V Tolerant	REQ0n	After power on reset, this pin is in input state for GPIO. If PCI device#0 is used in system, set related control register bit to enable REQ0n function.

GPIOB29	I/O, PU, 5V Tolerant	GNT0n	After power on reset, this pin is in input state for GPIO. If PCI device#0 is used in system, set related control register bit to enable GNT0n function.
GPIOB30	I/O, PU, 5V Tolerant	REQ1n	After power on reset, this pin is in input state for GPIO. If PCI device#1 is used in system, set related control register bit to enable REQ1n function.
GPIOB31	I/O, PU, 5V Tolerant	GNT1n	After power on reset, this pin is in input state for GPIO. If PCI device#1 is used in system, set related control register bit to enable GNT1n function.

1.14 Timer

Two 32 bit general purpose programmable timers are implemented. They can count up or down, and be clocking at 1kHz or fine APB bus clock (which can be programmable at 50MHz, 47.5MHz, etc. dependent on AHB clock frequency). If high-resolution timer is needed, an additional free running timer with resolution of 100kHz is also supported.

1.15 Watch-Dog Timer (WDT)

A 32 bit down counter, clocking at 10Hz, is employed for WDT. The output signals at time out can be one or combinations of system reset, and system interrupt. A WDT time out external reset pin is supported.

1.16 Real-time Counter (RTC)

The clock frequency of RTC is 1Hz. RTC provides separate *second, minute, hour,* and *day* counters to off load firmware complexity and reduce power consumption. It supports per second, per minute, and per hour auto alarm and, of course, any real time alarm. The RTC can be driven by a 25MHz reference clock.

1.17 Vector Interrupt Controller (VIC)

Low latency interrupt for serving multi-tasking software ISR is very beneficial in system performance. Vector Interrupt Controller is employed in the chip allowing directly executing routines determining the source of the interrupts. VIC also reduces the interrupt latency.

Software can assign some non-used hardware interrupt sources as software interrupt sources (for software interrupt generation). When entering into software ISR, the ISR should clear the interrupt source through Software Interrupt Clear Register. Between IRQ and FIQ request logic, it is designed with an asynchronous path allowing interrupt asserted when the clock is disabled.

Reading from the vector interrupt address register VectAddr, provides the address of the ISR, and updates the interrupt priority hardware that masks out the current and any lower priority interrupt requests. However, the values in the IRQ Status Register and Interrupt Source Register are not affected. Writing to the VectAddr register indicates to the interrupt priority hardware that the current interrupt is served, allowing lower

priority interrupts to go active.

To make sure that the Vector Address Register (VectAddr) can be read in a single instruction, the VIC base address is placed at 0xFFFF-F000, the upper 4K of memory. Placing the VIC anywhere else in memory increase interrupt latency, because the ARM processor is unable to access the VectAddr Register using a single instruction.

If no interrupt is currently active, the VectAddr Register holds the previous active interrupt address.

The VIC supports round-robin interrupt dispatcher scheme in a priority group to prevent starvation. There are 32 vectored IRQ interrupts, each can be programmed as level trigger or edge trigger. And 8 priority groups are defined with timing sensitive interrupts placed at higher priority groups. Software-generated interrupt is also supported.

Table 10. Peripheral Interrupt Source Mapping

Interrupt Register(0x00)	Interrupt Sources	Triggering Scheme
Bit[0]	Timer#1	Rising-edge Trigger
Bit[1]	Timer#2	Rising-edge Trigger
Bit[2]	CPU Frequency Scaling Interrupt	Falling-edge Trigger
Bit[3]	Watch Dog Timer	Rising-edge Trigger
Bit[4]	GPIO Controller	Programmable Trigger
Bit[5]	PCI External Interrupt 0	Low-level/Programmable Trigger
Bit[6]	PCI External Interrupt 1	Low-level/Programmable Trigger
Bit[7]	PCI Broken Interrupt	High-level Trigger
Bit[8]	AHB-to-PCI Bridge	High-level Trigger
Bit[9]	UART0	High-level Trigger
Bit[10]	UART1	High-level Trigger
Bit[11]	Generic DMA Terminal Counter	High-level Trigger
Bit[12]	Generic DMA Error	High-level Trigger
Bit[13]	Reserved	Programmable Trigger
Bit[14]	RTC	High-level Trigger
Bit[15]	PCM Controller	Low-level Trigger
Bit[16]	USB 1.1/2.0 Device Controller	Low-level Trigger
Bit[17]	IDE Device	High-level Trigger
Bit[18]	NIC MIB Counter	High-level Trigger
Bit[19]	NIC DMA TSTC *(To-NIC-Tx-Complete)*	Rising-edge Trigger
Bit[20]	NIC DMA FSRC *(Fm-NIC-Rx-Complete)*	Rising-edge Trigger
Bit[21]	NIC DMA TSQE *(To-NIC-Queue-Empty)*	Rising-edge Trigger
Bit[22]	NIC DMA FSQF *(Fm-NIC-Queue-Full)*	Rising-edge Trigger
Bit[23]	USB 1.1 host controller	Low-level Trigger
Bit[24]	USB 2.0 host controller	Low-level Trigger
Bit[25]	I2S Controller	Low-level Trigger
Bit[26]	SPI Controller	Low-level Trigger
Bit[27]	TWI Controller	Low-level Trigger
Bit[28]	USB Device VBUS interrupt	Rising and falling edge Trigger
Bit[29]	External Interrupt 29 (share pin with	Programmable Trigger

	GPIOA[0])	
Bit[30]	External Interrupt 30 (share pin with GPIOA[1])	Programmable Trigger
Bit[31]	HSDMA Terminal Counter and Error Interrupt	Rising-edge Trigger

1.17.1 External Interrupts

There are several external interrupt pins as defined, but they are also shared with other pins as addressed in Table 11.

Table 11. Shared External Interrupt

External Interrupt	Type	Shared Pin Name	Description of Shared Pin
EXT_INT5	I, PU	INT0n	If PCI device #0 is not used in application, this pin can be re-assigned for generic external interrupt use.
EXT_INT6	I, PU	INT1n	If PCI device #1 is not used in application, this pin can be re-assigned for generic external interrupt use.
EXT_INT13	I, PU	-	Reserved
EXT_INT28	I/O, PD	VBUS	If USB2.0 Device interface is not used in application, this pin can be re-assigned for generic external interrupt use.
EXT_INT29	I/O	GPIOA0	After power-on-reset, this pin is in input state for GPIOA0 by default. User can re-assign this as an external interrupt pin, by setting relevant control register bit to enable the external interrupt function (and disable the GPIOA0 function) accordingly.
EXT_INT30	I/O	GPIOA1	After power-on-reset, this pin is in input state for GPIOA1 by default. User can re-assign this as an external interrupt pin, by setting relevant control register bit to enable the external interrupt function (and disable the GPIOA1 function) accordingly.

1.18 I^2S Interface

The I^2S controller is used to transport the audio data to/from external peripherals, like high-end audio codec chips, for network transmission or streaming/accessing. The sampling clock frequencies (WS rate) of the I^2S can be programmable to 32kHz, 44.1kHz or 48kHz.

Please note, chip version AD supports full duplex operation (version AC only supports half-duplex operation). By configuration, pin “I2SSD” can be input or output data pin. And pin “GPIOA[3]/I2SDR/UR_ACT1” can be I2S data input pin. That means I2S data input can be from I2SSD pin or I2SDR pin. If user needs full duplex function, I2SSD must be configured to be output (I2S_CFG.I2SSD_DIR=0), and pin “GPIOA[3]/I2SDR/UR_ACT1” must be configured to be I2SDR, I2S data input pin (I2S_CFG.I2S_IN_SEL=1 and MISC.GPIOA_EN.UR_ACT1=0 and GPIOA.PinDir.PinDir[3]=0). Although, I2SSD can be input/output, recommend to fix it at output, and always keep “GPIOA[3]/I2SDR/UR_ACT1” to be I2SDR, I2S data input.

The data frame transferred in I^2S bus shall be in 2’s complement format with MSB

transmitted first. The data width for transferring can be programmable as 16- or 32-bit wide.

- Support hardware handshake DMA's, one for TX and another for RX.
- Six(6) Modes of Operation
  - Transmitter in Master mode,
  - Receiver in Master mode,
  - Transmiter and Receiver in Master mode,
  - Transmitter in Slave mode,
  - Receiver in Slave mode,
  - Transmitter and Receiver in Slave mode;
- THREE(3) Data Formats
  - I^2S,
  - Right Justified (RJF),
  - Left Justified (LJF);
- TWO(2) Serial Clock Transfer Modes
  - 256-S Transfer Mode,
  - Continuous Transfer Mode;
- Left- and Right-channel Transmission Indication, by signal of Word Select(WS)
  - WS=0; left channel being transmitted for I^2S,
  - WS=1; right channel being transmitted for I^2S.

1.19 Two-Wire Serial Interface (TWI)

The TWI is used as serial 2-wire bus control interface for external devices. Only master mode is supported. It supports only one master in the bus. It complies with TWI Bus Specification version 2.0—normal bus data rate (100kHz) and fast bus data rate (400KHz). It also supports clock frequency up to 12.5MHz. Both normal (7-bit) and extended (10-bit) addressing mode are supported. Additionally, it can automatically check bus error (check if SCL or SDA stuck at zero). Data frame is transferred on TWI with the MSB transferred first.

Totally 4 transfer commands can be used: only-read-cmd, only-write-cmd, write-read-cmd and read-write-cmd. The maximum serial data transferring rate across TWI can be up to 12.5Mbps.

1.20 Pulse-Coded Modulation (PCM) Interface

The PCM can be used for interfacing with external PCM devices, like linear PCM codec chip with 8kHz sampling, for voice-over-IP or IP CAM applications.

Supporting features are highlighted as below:

- Support up to 4 external PCM devices shared the PCM-bus via Time-Division Multiplexing (TDM)
- Support Master and Slave mode operations
- Support 8- and 16-bit data per channel
- Supports short and long FRAME SYNC (sampling frequency: 8kHz)
- The PCM interface can be configured to the following modes of operation:
  - Tx/Rx-full duplex
  - TX-only
  - Rx-only.
- Master clock rate of PCM (PCM_CLK RATE) can be selected from 4.096MHz/N (where N=1, 2, 4, 8, 16, or 32)
- Supports IDL and GCI modes of operation
  - IDL: data bit rate=PCMCLK,
  - GCI: data bit rate=PCMCLK/2.

The following Table shows the relationship among Channel Number, Master Clock Rate, IDL/GCI mode, and Data Width

Table 12. PCM Configuration Guide Line

Master Clock	IDL Mode	GCI Mode
4.096MHz	Up to 4 channels with max data width 16-bit each;	Up to 4 channels with max data width 16-bit each;
2.048MHz	Up to 4 channels with max data width 16-bit each;	Up to 4 channels with max data width 16-bit each;
1.024MHz	Up to 4 channels with max data width 16-bit each;	Up to 2 channels with max data width 16-bit each;
512kHz	Up to 2 channels with max data width 16-bit each;	Only 1 channel with max data width 16-bit;
256kHz	Up to 1 channels with max data width 16-bit each;	Only 1 channel with data width 8-bit;
128kHz	Up to 4 channels with max data width 8-bit each;	Not Available

1.21 Serial Peripheral Interface (SPI)

The Serial Peripheral Interface (SPI) is a 4-wire bus: Tx Data(SPI_DT), Rx Data (SPI_DR), Clock(SPI_CLK), and Select (SPI_nCSx). It provides a synchronous, full-duplex, and serial communication between master and slave, peripheral or devices. For VoIP application, SPI is usually used to configure and control the external CODEC & SLIC chip.

Supporting features are highlighted as below:

- Support Master and Slave mode of operations
- Support up to 4 chip-select pins for up to 4 SPI-bus devices(slaves)
- Support SPI serial Flash memory booting
- Support Generic SPI and Microprocessor Interface(MPI), which can be selectable
- Support MPI of popular CODEC/SLIC chips for VoIP application
- Support hardware handshake DMA's—one for Tx, the other for Rx
- Support 8 data transfer rates: baud-rate is equal to FPCLK/N, where N=1, 2, 4, 8, 16, 32, 64, and 128 configurable
- Supports SPI Clock(SPICLK) Polarity Change(HIGH/LOW)
- Support Tx/Rx FIFO/buffer and with under-/over-run handling
- Supports Tx/Rx-Full Duplex, Tx-only , or Rx-only modes of operation for SPI-master and SPI-slave

1.22 Clock Generator and Power Management

A sophisticated Clock Generator is designed with clock domains as showed in following figure. External 25MHz and 32.768kHz reference inputs are used to generate all of the clocks required for individual functional blocks, such as RISC CPU, AHB peripherals, APB peripherals, GbE Controller(NIC), PCI Bridge, UART, USB host, USB device, … etc. An intelligent Power Management is also implemented. Major embedded functional blocks can be powered down by programmable gated-clock. Moreover, it supports programmable operation clock frequency for CPU, AHB bus, APB bus, and PCI bus to optimize Power/Performance ratio.

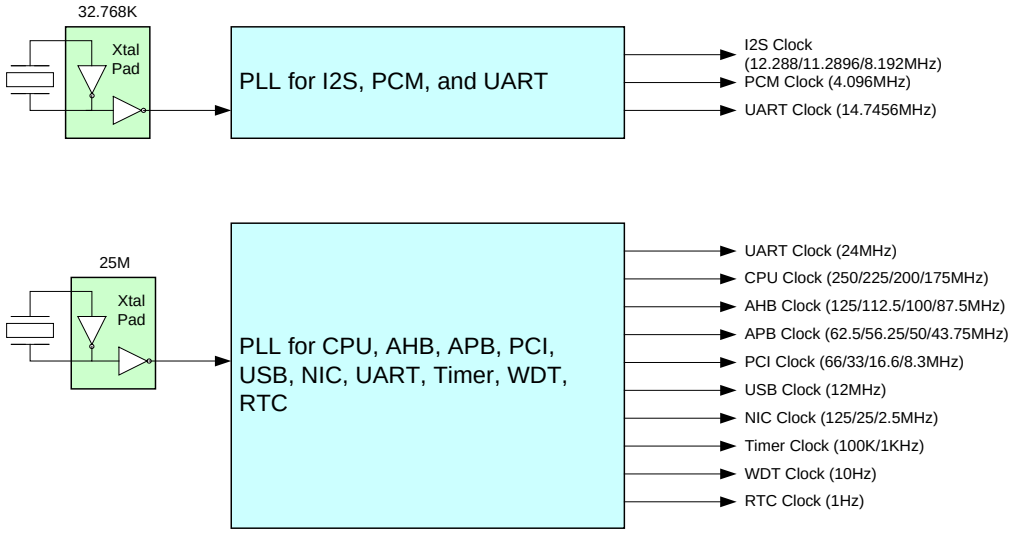


Figure 13. Overall Clock Domains

1.23 Linear Regulators

Three on-chip voltage regulator controllers with off-chip low-cost PNP BJT's are included to reduce system BOM—for 3.3-to-2.5V, 3.3-to-1.8V and 3.3V-to-1.25V step-down linear regulators. The following figure shows the typical application circuit for the Regulators. It should be noted that the regulated 1.8V, 2.5V, and 1.25V can supply on-chip peripherals as well as off-chip key components in system (for example, external gigabit Ethernet PHY, DDR). Further more, the 3.3-to-1.25V regulator can track 2.5V power supply, and to be just half of 2.5V supply voltage. It can be used as reference voltage of DDR, but it doesn't support current sink capability. When in SDR mode, or when 1.25V supply is not required in system, it can be powered down.

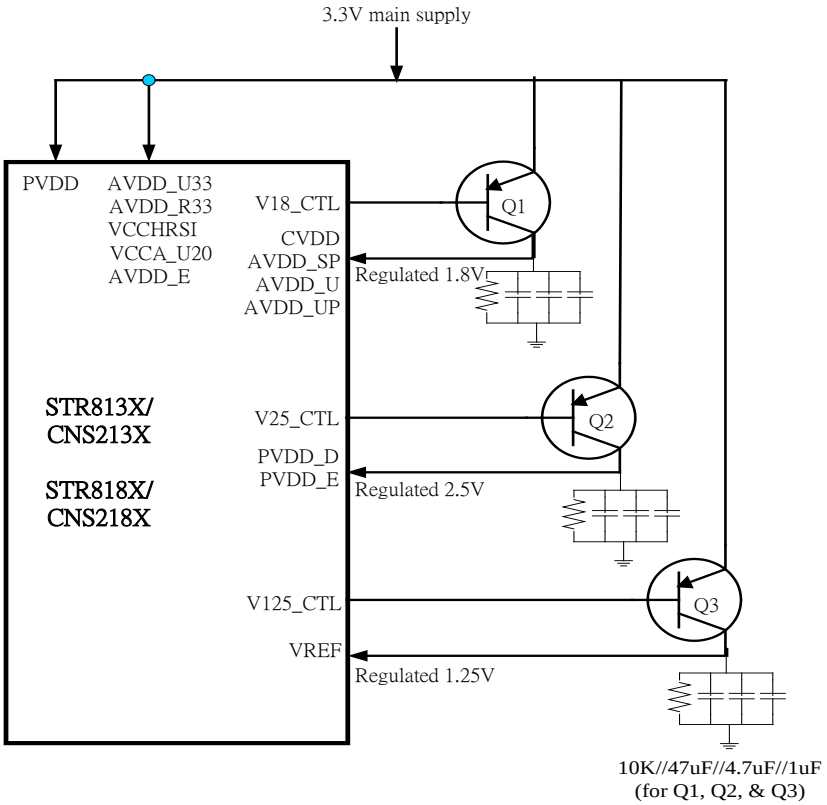


Figure 14. Typical Application Circuits for Regulator

2 Pin Assignment and Descriptions

Cross-reference matrix addressing the pin assignment for different part numbers are summarized in Table 14.

Table 13. Pin Type Notations

I/O Type	Function
I	Digital Input Pin
O	Digital Output Pin
I/O	Digital Bi-directional Pin
PU	Pin with Pull Up Resistor
PD	Pin with Pull Down Resistor
5VT	5V Tolerant
A	Analog Pin
AO	Analog Output Pin
AI	Analog Input Pin
P	Power Supply
G	Power Ground

Table 14. Pin Assignment

Part Number	STR 8131/ CNS 2131	STR 8132/ CNS 2132	STR 8133/ CNS 2133	CNS 2135	STR 8181/ CNS 2181	STR 8182/ CNS 2182	STR 8182X /CNS 2182X	I/O Type	Pin Descriptions	Alternative Pins
PKG Type / Pin Name	LQFP 128	LQFP 128	LFBGA 269	LQFP 128	LQFP 128	LFBGA 269	LFBGA 269	*(..) means DEFAULT	PU: Pull-up PD: Pull-Down	
DDR/SDR Memory Interface										
CK, CKn	V	V	V	V	V	V	V	O	**DDR Clock Output**. CK and CKn are differential clock outputs. SDR: CKn is not used.	
CKE/ SPIBoot	V	V	V	V	V	V	V	I/O	**Clock Enable** This pin is also used as power on reset latch of SPI Flash Boot Enable, configuring by external	Power on reset latch of Boot selection

									pull-up or -down resistor. ✓ Pull-down, boot from parallel flash memory. ✓ Pull-up, boot from SPI serial flash memory.	
DCSn	V	V	V	-	V	V	V	O	**Chip Select**, LOW active.	
RASn	V	V	V	-	V	V	V	O	**Row Address Select** One of command signals.	
CASn	V	V	V	-	V	V	V	O	**Column Address Select** One of command signals.	
DWEn	V	V	V	-	V	V	V	O	**Write Enable** One of command signals.	
DM[1:0]	V	V	V	-	V	V	V	O	**Data Mask**	
BA[1:0]	V	V	V	-	V	V	V	O	**Bank Address**	
DA[12:0]	V	V	V	-	V	V	V	O	**Row or Column Address Bus**	
DDQ[15:0]	V	V	V	-	V	V	V	I/O	**Data Bus**	
VREF	V	V	V	-	V	V	V	AI	**DDR I/O Reference Voltage** (1.25V). Not used for SDR	
DQS[1:0]	V	V	V	-	V	V	V	I/O	**DDR Data Strobe** Input for read data, output for write data. Not used for SDR	
PCI Interface										
AD[31:0]	-	-	V	-	-	V	V	I/O, 5VT	**PCI Address and Data bus**	

CBEn[3:0]	-	-	V	-	-	V	V	I/O, 5VT	**Bus command and Byte Enables**, LOW active.	
PAR	-	-	V	-	-	V	V	I/O, 5VT	**Even Parity** across AD[31:0] and C_BE_n[3:0]	
FRAMEn	-	-	V	-	-	V	V	I/O, PU, 5VT	**Cycle Frame** is driven by the current master to indicate the beginning and duration of the access; LOW active.	
IRDYn	-	-	V	-	-	V	V	I/O, PU, 5VT	**Initiator Ready** indicates the initiating agent's ability to complete the current data phase of the transaction; LOW active.	
TRDYn	-	-	V	-	-	V	V	I/O, PU, 5VT	**Target Ready** indicates the target agent's ability to complete the current data phase of the transaction; LOW active.	GPIOB[23]
STOPn	-	-	V	-	-	V	V	I/O, PU, 5VT	**Stop** indicates the current target is requesting the master to stop the current transaction; LOW active.	GPIOB[25]
DEVSELn	-	-	V	-	-	V	V	I/O, PU, 5VT	**Device Select**, when actively driven, indicates the driving device has decoded its address as	GPIOB[24]

									the target of the current access; LOW active.	
PERRn	-	-	V	-	-	V	V	I/O, PU, 5VT	**Parity Error** indicates data parity errors during all PCI transactions except at Special Cycle; LOW active.	GPIOB[26]
SERRn	-	-	V	-	-	V	V	I/O, PU, 5VT	**System Error** indicates address errors, data parity errors on the Special Cycle command, or any other system error where the result will be catastrophic; LOW active.	GPIOB[27]
REQ0n REQ1n	-	-	V	-	-	V	V	I/O, PU, 5VT	**Request** indicates to the arbiter that agent 0, 1 desires use of the bus; low active. *Note: REQ1n shared with GPIO pin*	GPIOB[28] GPIOB[30]
GNT0n GNT1n	-	-	V	-	-	V	V	I/O, PU, 5VT	**Grant** indicates to agent 0, 1 that access to the bus has been granted, LOW active. *Note: GNT1n shared with GPIO*	GPIOB[29] GPIOB[31]
INT0n INT1n	-	-	V	-	-	V	V	I, PU	**PCI Interrupt 0, 1** for 2 PCI agents to request an interrupt to the host; LOW active.	EXT_INT5 EXT_INT6

									Note: These interrupt pins can be used as generic external interrupt pins, if the related PCI device is not connected.	
PCIRSTn	-	-	V	-	-	V	V	O	**PCI Reset**, LOW active.	
PCICLK	-	-	V	-	-	V	V	O	**PCI Clock**	
Static Parallel Memory Interface								**5V Tolerant**		
SCE0n SCE1n SCE2n SCE3n	-	-	V	-	-	V	V	I/O, PU, 5VT	**Static Memory Chip Enable of Bank 0~3**, low active *Note: SCE1n, SCE2n and SCE3n share with GPIO pins.*	- GPIOB[11] GPIOB[12] GPIOB[13]
SWEn	-	-	V	-	-	V	V	O	**Static Memory Write Enable, low active**	
SOEn	-	-	V	-	-	V	V	O	**Static Memory Output Enable**, LOW active	
SWAIT1n SWAIT2n SWAIT3n	-	-	V (1:2)	-	-	V (1:2)	V (1:2)	I/O, PU, 5VT	**Static Memory Wait Hand Shake Signal for SMC Bank 1, 2, and 3**, LOW active *Note: These pins share with GPIO pins.*	GPIOB[14] GPIOB[15] GPIOB[16]
SA[15:0]	-	-	V	-	-	V	V	O	**Static Memory Address Bus [15:0]** For 8-bit SRAM device, the address bus width is 24, SA[23:0]. For 16-bit	

									SRAM device, the address bus width is 16, SA[15:0].	
SA[23:16] / SDQ[15:8]	-	-	V	-	-	V	V	I/O, PD, 5VT	**Static Memory Address Bus [23:16] or Data Bus [15:8]** For 8-bit SRAM device, the address bus width is 24, SA[23:0]. For 16-bit SRAM device, the address bus width is 16, SA[15:0]. Namely, SA[23:16] and SDQ[15:8] share the same 8 pins. The data bus width can be programmable through SMC control registers.	
SDQ[7:0]	-	-	V	-	-	V	V	I/O, PD, 5VT	**Static Memory Data Bus [7:0]** For 8-bit SRAM device, the address bus width is 24. For 16-bit SRAM device, the address bus width is 16	
IDE Interface								**5V Tolerant**		
IDECS0n	-	-	V	-	-	V	V	I/O, PU, 5VT	**IDE Device Chip Select for Command Register Block** For ATA command register block	GPIOB[7]

IDECS1n	-	-	V	-	-	V	V	I/O, PU, 5VT	**IDE Device Chip Select for Control Register Block** For ATA control register block	GPIOB[8]
IDEA[2:0]	-	-	V	-	-	V	V	O	**IDE Device Address**	SA[2:0]
IDED[15:0]	-	-	V	-	-	V	V	I/O, PD, 5VT	**IDE Device Data**	SDQ[15:0]
DMARQ	-	-	V	-	-	V	V	I/O, PD, 5VT	**IDE Device DMA Request** It is asserted by IDE device to request a data transfer.	GPIOB[4]
DMACKn	-	-	V	-	-	V	V	I/O, PU, 5VT	**IDE Device DMA Acknowledge** It indicates to IDE DMA slave devices that a given data transfer cycle (assertion of DIORn or DIOWn) is a DMA data transfer cycle.	GPIOB[5]
DIORn / DWSTB / RDMARDYn	-	-	V	-	-	V	V	I/O, PU, 5VT	**Disk I/O Read (PIO and Non-Ultra DMA)** Data is latched by the Host on the de-assertion edge of DIORn **Disk Write Strobe (Ultra DMA Write to Disk)** This is the data write strobe for writes to disk. When writing to disk, Host drives valid data on the rising and	GPIOB[9]

									falling edges of DWSTB. **Disk DMA Ready (Ultra DMA Reads from Disk)** This is the DMA ready for reads from disk. When reading from Disk, Host de-asserts RDMARDYn to pause burst data transfers.	
DIOWn / DSTOP	-	-	V	-	-	V	V	I/O, PU, 5VT	**Disk I/O Write (PIO and Non-Ultra DMA)** Data is latched by the device on the de-assertion edge of DIOWn **Disk Stop (Ultra DMA)** Host asserts this signal to terminate a burst.	GPIOB[10]
IORDY / DRSTB / WDMARDYn	-	-	V	-	-	V	V	I/O, PU, 5VT	**I/O Channel Ready (PIO)** This signal will keep the strobe active (DIORn or DIOWn) longer than the minimum width. It adds wait-states to PIO transfers. **Disk Read Strobe (Ultra DMA Read from Disk)** When reading from disk, Host latches data on rising and falling edges of this	GPIOB[3]

									signal from the disk. **Disk DMA Ready (Ultra DMA Writes to Disk)** When writing to Disk, Disk de-asserts WDMARDYn to pause burst data transfers.	
INTRQ	-	-	V	-	-	V	V	I/O, PD, 5VT	**IDE Device Interrupt**	GPIOB[6]
RGMII/MII/Reverse MII Interfaces										
TXCLK	-	-	-	-	V	V	V	I/O	**Transmit Clock.** ● MII: transmit clock, 25/2.5 MHz, input ● Reverse MII: receive clock, 25/2.5 MHz, output ● RGMII: transmit clock, 125/25/2.5 MHz, output	
TXD[3:0]	-	-	-	-	V	V	V	O	**Transmit Data.** ✓ MII: transmit data, output ✓ Reverse MII: receive data, output ✓ RGMII: transmit data [3:0] at TXCLK rising	

									edge and [7:4] at TXCLK falling edge when 1000Mbps ; and transmit data [3:0] and [7:4] only at TXCLK rising edge when 10/100Mbps, output.	
TXEN	-	-	-	-	V	V	V	O	**Transmit Enable** ✓ MII: TXEN, output ✓ Reverse MII: RXDV, output ✓ RGMII: TX_CTL, output	
RXCLK	-	-	-	-	V	V	V	I/O	**Receive Clock.** ✓ MII: receive clock, 25/2.5 MHz, input ✓ Reverse MII: transmit clock, 25/2.5 MHz, output ✓ RGMII: receive clock, 125/25/2.5 MHz, input	
RXD[3:0]	-	-	-	-	V	V	V	I	**Receive Data.** ✓ MII:	

									✓ receive data, input ✓ Reverse MII: transmit data, input ✓ RGMII: receive data [3:0] at RXCLK rising edge and [7:4] at RXCLK falling edge when 1000Mbps; and receive data [3:0] and [7:4] only at rising edge when 10/100Mbps, input.	
RXDV	-	-	-	-	V	V	V	I	**Receive Data Valid** ✓ MII: RXDV, input ✓ Reverse MII: TXEN, input ✓ RGMII: RX_CTL, input	
COL	-	-	-	-	V	V	V	I/O, PD	**Collision** ✓ MII: Collision, input ✓ Reverse MII: Collision, output ✓ RGMII: useless, configure this pin	GPIOB[2]

									as GPIO Note: The IO power supply is from PVDD_E. It can be 2.5V if RGMII interface is used.	
MDC	-	-	-	-	V	V	V	I/O, PU, 5VT	**Management Data Clock**. Only support MII & RGMII It is shared with GPIO pin. Note: The IO power supply is from PVDD_E. It can be 2.5V if RGMII interface is used.	GPIOB[0]
MDIO	-	-	-	-	V	V	V	I/O, PU, 5VT	**Management Data Input/Output**. Only support MII & RGMII. It is shared with a GPIO pin. Note: The IO power supply is from PVDD_E. It can be 2.5V if RGMII interface is used.	GPIOB[1]
10/100 PHY										
RX+ RX-	V	V	V	V	-	-	-	A	**Receive Differential Pair**	
TX+ TX-	V	V	V	V	-	-	-	A	**Transmit Differential Pair**	
REF_RES	V	V	V	V	-	-	-	A	**Reference Resistor**	
LED0	V	V	V	V	V	-	-	I/O, PU	**LED 0** Its output signal can be programmed. For example,	GPIOA[22]

									Link, Activity, Speed, Duplex, Collision. It is shared with a GPIO pin.	
LED1	V	V	V	V	V	-	-	I/O, PU	**LED 1** Its output signal can be programmed. For example, Link, Activity, Speed, Duplex, Collision. It is shared with a GPIO pin.	GPIOA[23]
LED2	V	V	V	V	V	-	-	I/O, PU	**LED 2** Its output signal can be programmed. For example, Link, Activity, Speed, Duplex, Collision. It is shared with a GPIO pin.	GPIOA[24]
USB Host Interfaces										
P0_DP, P0_DM	V	V	V	V	V	V	V	A	USB 1.1/2.0 Host PHY Port 0 Differential Signals	
P1_DP, P1_DM	V	V	V	V	V	V	V	A	USB 1.1/2.0 Host PHY Port 1 Differential Signals	
REXT	V	V	V	V	V	V	V	A	External reference resistance for signal swing	
USB Device Interfaces										
VBUS	-	V	V	-	-	V	V	I, PD	USB device attachment indicator, connect to USB bus power. Note: This pin	EXT_INT28

									can be used as the generic external interrupt pin EXT_INT28, if USB device interface is not used.	
DDP, DDM	-	V	V	-	-	-	V	A	USB 1.1/2.0 Device PHY Differential Signals	
RREF	-	V	V	-	-	-	V	A	External reference resistance for signal swing (12kohm+-1%) to analog GND	
JTAG Interfaces										
ICK/TCK	V	V	V	V	V	V	V	I/O	JTAG Test Clock	
IMS/TMS	V	V	V	V	V	V	V	I/O	JTAG Test Mode Select	
IDIO/TDO	V	V	V	V	V	V	V	I/O	ARM-like ICE Data In/Out or JTAG Test Data Out	
EXTGOICE/TDI	V	V	V	V	V	V	V	I/O	ARM-like ICE control signal or JTAG Test Data In	
TRSTn	V	V	V	V	V	V	V	I/O	JTAG Reset	
UART Interfaces										
UR_TXD0	V	V	V	V	V	V	V	I/O, PU	UART0 TX Data	
UR_RXD0	V	V	V	V	V	V	V	I/O, PU	UART0 RX Data	
UR_ACT0	V	V	V	V	V	V	V	I/O,(PD) (4mA)	UART0 TX Active Indicator 0: Idle 1: Active	GPIOA[2]
UR_TXD1	V	-	V	-	V	V	V	I/O, PU	UART1 TX Data	GPIOB[21]
UR_RXD1	V	-	V	-	V	V	V	I/O, PU	UART1 RX Data	GPIOB[22]
UR_ACT1	V	V	V	V	V	V	V	I/O,(PD) (4mA)	UART1 TX Active Indicator 0: Idle 1: Active	GPIOA[3]

PCM Interfaces										
PCMCLK	V	-	V	V	-	V	V	I/O, PD	PCM Clock Master: output/ Slave: input	GPIOA[21]
PCMFS	V	-	V	V	-	V	V	I/O, PD	Frame Sync signal Master: output/ Slave: input	GPIOA[20]
PCMDT	V	-	V	V	-	V	V	I/O, PD	Transmit Data signal	GPIOA[19]
PCMDR	V	-	V	V	-	V	V	I/O, PD	Receive Data signal	GPIOA[18]
SPI Interface										
SPICS0n SPICS1n SPICS2n SPICS3n	V (0:2)	V (0:1)	V	V (0:1)	V	V	V	I/O, PU I/O, PU I/O, PU I/O, PU	**SPI Chip Select** Master: 4 chip select pins are output pins Slave: SPICS0n is input pin, others are don't-care	GPIOA[28] GPIOA[29] GPIOA[30] GPIOA[31]
SPICLK	V	V	V	V	V	V	V	I/O, PD	**SPI clock**	GPIOA[27]
SPIDT	V	V	V	V	V	V	V	I/O	**SPI data out** This pin is also used as power on reset latch of Endian Select, configuring by external pull up or pull down resistor. Pull-down: CPU runs at little-endian mode. Pull-up: CPU runs at big-endian mode.	Power on reset latch of Endian selection
SPIDR	V	V	V	V	V	V	V	I/O, PD	**SPI data in**	GPIOA[26]
TWI Interfaces										
SCL	V	V	V	-	-	V	V	I/OD, PU	**TWI Clock**	GPIOA[14]
SDA	V	V	V	-	-	V	V	I/OD, PU	**TWI Data**	GPIOA[13]
I2S Interfaces										

I2SCLK	V	V	V	V	-	V	V	I/O, PD	**I2S bus Serial Clock** Master: output Slave: input	GPIOA[17]
I2SWS	V	V	V	V	-	V	V	I/O, PD	**I2S bus Word Select** Master: output Slave: input	GPIOA[16]
I2SSD	V	V	V	V	-	V	V	I/O, PD	**I2S bus Serial Data** It can be serial data input or output.	GPIOA[15]
I2SDR	V	V	V	V	-	V	V	I, PD	**I2S bus Serial Data Input**	GPIOA[3]
GPIO Interfaces	**21 pins**	**13 pins**	**50 pins**	**20 pins**	**16 pins**	**49 pins**	**49 pins**			
GPIOA[1:0]	V	V	V	V	V	V	V	I/O, (PD) (4mA) Schmitt Trigger	GPIOA[1:0] These pins are shared with external interrupt pins: EXT_INT29, and EXT_IN30. Note: When the following TESTMODE_EN pin is asserted, GPIOA[2:0] are defined as TESTMODE[2:0] Note: GPIOA[1:0] internal pull up/down resistor are programmable. User can base on application to change these GPIO pads to be with pull up resistor or pull down resistor. Default, theses pads are with pull down resistor.	

									And GPIOA[1:0] output drive strength are also programmable. They can be 4mA or 8mA. Default, they are 4mA.	
GPIOA[2]	V	V	V	V	V	V	V	I/O, (PD) (4mA) Schmitt Trigger	GPIOA[2] This pin is shared with UART0 active pin, UR_ACT0 *Note: When the following TESTMODE_EN pin is asserted, GPIOA[2:0] are defined as TESTMODE[2:0]* Note: GPIOA[2] internal pull up/down resistor are programmable. User can base on application to change these GPIO pads to be with pull up resistor or pull down resistor. Default, theses pads are with pull down resistor. And GPIOA[2] output drive strength are also programmable. They can be 4mA or 8mA. Default, they are 4mA.	
GPIOA[3]	V	V	V	V	V	V	V	I/O, (PD)	GPIOA[3] This is shared	

								(4mA)	with UART1 active pin, UR_ACT1. Note: When the following TESTMODE_EN pin is asserted, GPIOA[3] is defined as BYPASS. Note: GPIOA[3] internal pull up/down resistor are programmable. User can base on application to change these GPIO pads to be with pull up resistor or pull down resistor. Default, theses pads are with pull down resistor. And GPIOA[3] output drive strength are also programmable. They can be 4mA or 8mA. Default, they are 4mA.	
GPIOA[7:4]	-	-	V	-	-	6:4	6:4	I/O, (PD) (4mA)	GPIOA[7:4] These are dedicated GPIO pins Note: GPIOA[7:4] internal pull up/down resistor are programmable. User can base on application to change these	

									GPIO pads to be with pull up resistor or pull down resistor. Default, theses pads are with pull down resistor. And GPIOA[7:4] output drive strength are also programmable. They can be 4mA or 8mA. Default, they are 4mA.	
GPIOA[12:8]	-	-	-	-	-	-	-	I/O, (PU) (4mA)	GPIOA[12:8] These are dedicated GPIO pins Note: GPIOA[12:8] internal pull up/down resistor are programmable. User can base on application to change these GPIO pads to be with pull up resistor or pull down resistor. Default, theses pads are with pull up resistor. And GPIOA[12:8] output drive strength are also programmable. They can be 4mA or 8mA. Default, they are 4mA.	
GPIOA[31:13]	24:13 25 30:29	17:13 24:22 29	31:13	15:24 26:29	25:22 31:29	21:13 31:25	21:13 31:25	I/O PU or PD	GPIOA[31:13] They are shared with other	

									functional pins.	
GPIOB[31:0]	22:21	-	12:3 15,14 31:21	4:3	2:0 22:21	12:0 15:14 31:21	12:0 15:14 31:21	I/O PU or PD	GPIOB[31:0] They are shared with other functional pins.	
External Interrupt										
EXT_INT5	-	-	V	-	-	V	V	I, PU	Since this PCI interrupt 0 pin connect into internal VIC directly, if PCI device #0 doesn't exist, this pin can be used as a generic external interrupt pin.	INT0n
EXT_INT6	-	-	V	-	-	V	V	I, PU	Since this PCI interrupt 1 pin connect into internal VIC directly, if PCI device #1 doesn't exist, this pin can be used as a generic external interrupt pin.	INT1n
EXT_INT13	-	-	-	-	-	-	-	I, PU	This pin can be used as a generic external interrupt pin.	
EXT_INT28	-	V	V	-	-	V	V	I/O, PD	Since this USB Device VBUS Attach interrupt pin connect into internal VIC directly, if USB Device interface is not used, this pin can be used as a generic external interrupt pin.	VBUS
EXT_INT29	V	V	V	V	V	V	V	I/O, (PD)	After power on reset, this pin	GPIOA[0]

									is GPIOA0 in input state. If it must be treated as an external interrupt pin, set related control register bit to enable the external interrupt function and disable the GPIOA0 function.	
EXT_INT30	V	V	V	V	V	V	V	I/O, (PD)	After power on reset, this pin is GPIOA1 in input state. If it must be treated as an external interrupt pin, set related control register bit to enable the external interrupt function and disable the GPIOA1 function.	GPIOA[1]
Miscellaneous Interfaces										
RESETn	V	V	V	V	V	V	V	I	**System Reset**, LOW active.	
WDTRSTn	V	-	V	-	V	V	V	O	**WDT Reset**, LOW active.	GPIOA[25]
SXIN	V	V	V	V	V	V	V	AI	**Crystal In** for system reference clock (25MHz).	
SXOUT	V	V	V	V	V	V	V	AO	**Crystal Out** for system reference clock.	
REF_32768	V	V	V	V	V	V	V	AI	**32.768KHz Reference Clock input from Oscillator.**	

									Please note, the input voltage level is between 0~1.8V, but not 0~3.3V.	
CLKOUT	V	V	V	V	V	V	V	I/O	*Clock output* Its frequency can be 25M, 12M, 4.096MHz, ... This pin is also used as power on reset latch of ICE Select, configuring by external pull up or pull down resistor. ✓ Pull-down, CPU runs at ARM-like ICE mode. ✓ Pull-up, CPU runs at Multi-ICE mode.	Power on reset latch of ICE selection.
V25_CTL	V	V	V	V	V	V	V	AO	3.3V-to-2.5V Regulator control pin for external PNP BJT	
V18_CTL	V	V	V	V	V	V	V	AO	3.3V-to-1.8V Regulator control pin for external PNP BJT	
V125_CTL	V	V	V	-	V	V	V	AO	3.3V-to-1.25V Regulator control pin for external PNP BJT	
TESTMODE_EN	V	V	V	V	V	V	V	I, PD	Test Mode Enable Pin, for test purpose only; keep it in low.	
Power and Ground										
VFS	V	V	V	V	V	V	V	P	eFuse Power Supply for programming	

									(3.8V)	
CGND+PGND	-	-	28	-	-	31	30	G	Digital Core Ground and Pad Ground	
CGND	4	4	-	7	4	-	-	G	Digital Core Ground	
CVDD	7	7	7	4	7	7	7	P	Core Power Supply (1.8V)	
PGND	5	5	-	7	6	-	-	G	Pad Ground	
PVDD	3	3	8	5	3	7	8	P	Pad Power Supply (3.3V)	
PVDD_D	2	2	2	2	2	2	2	P	DDR/SDR I/O Pad Power Supply (2.5V or 3.3V)	
PVDD_E	-	-	-	-	1	1	1	P	MII/RGMII Power Supply (3.3V or 2.5V)	
AVDD_E	2	2	3	2	-	-	-	P	Fast Ethernet PHY Power Supply (3.3V)	
AGND_E	2	2	3	2	-	-	-	G	Fast Ethernet PHY Ground	
AVDD_SP	1	1	1	1	1	1	1	P	System PLL Power Supply (1.8V)	
AGND_SP	1	1	1	1	1	1	1	G	System PLL Ground	
AVDD_U	-	-	1	-	-	1	1	P	USB2.0 Host PHY Analog Power Supply (1.8V)	
AGND_U	1	1	1	1	1	1	1	G	USB2.0 Host PHY Analog Ground	
AVDD_UP	1	1	1	1	1	1	1	P	USB Host PHY PLL Analog Power Supply (1.8V)	
AGND_UP	1	1	1	1	1	1	1	G	USB Host PHY PLL Analog Ground	
AVDD_U33	1	1	1	1	1	1	1	P	USB1.1 Host PHY Analog Power Supply (3.3V)	
AVDD_R33	1	1	1	1	1	1	1	P	Analog Power (3.3V) of Regulator 3.3V-to-2.5V, and 3.3V-to-1.8V, and	

									3.3V-to-1.25V	
AGND_R	-	-	1	V	-	1	1	G	Ground of Regulator 3.3V-to-2.5V, and 3.3V-to-1.8V, and 3.3V-to-1.25V	
VCCHRST	-	1	1	-	-	-	-	P	USB Device PHY analog power supply (3.3V)	
GNDHRST	-	1	1	-	-	-	-	G	USB Device PHY Analog ground supply	
VCCA_U20	-	1	1	-	-	-	-	P	USB Device PHY analog power supply (3.3V)	
GNDA_U20	-	1	1	-	-	-	-	G	USB Device PHY Analog ground supply	

Table 15. Reset-Latch Configuration Pins

Configuration	Pin Name	Settings *(PU=External Pull-Up, PD=External Pull-Down)*
Multi-ICE Selection	CLKOUT	PU: ARM Multi-ICE PD: ARM-like ICE.
Endian Selection	SPIDT	PU: Big-Endian PD: Little-Endian
SPI Boot Enable	CKE	PU: SPI Serial Flash Boot enable PD: Parallel Flash Boot enable

2.1 Package Pin-out

2.1.1 STR8131/CNS2131 (PQFP-128)

The package pin out for STR8131/CNS2131 is shown in the following figure.

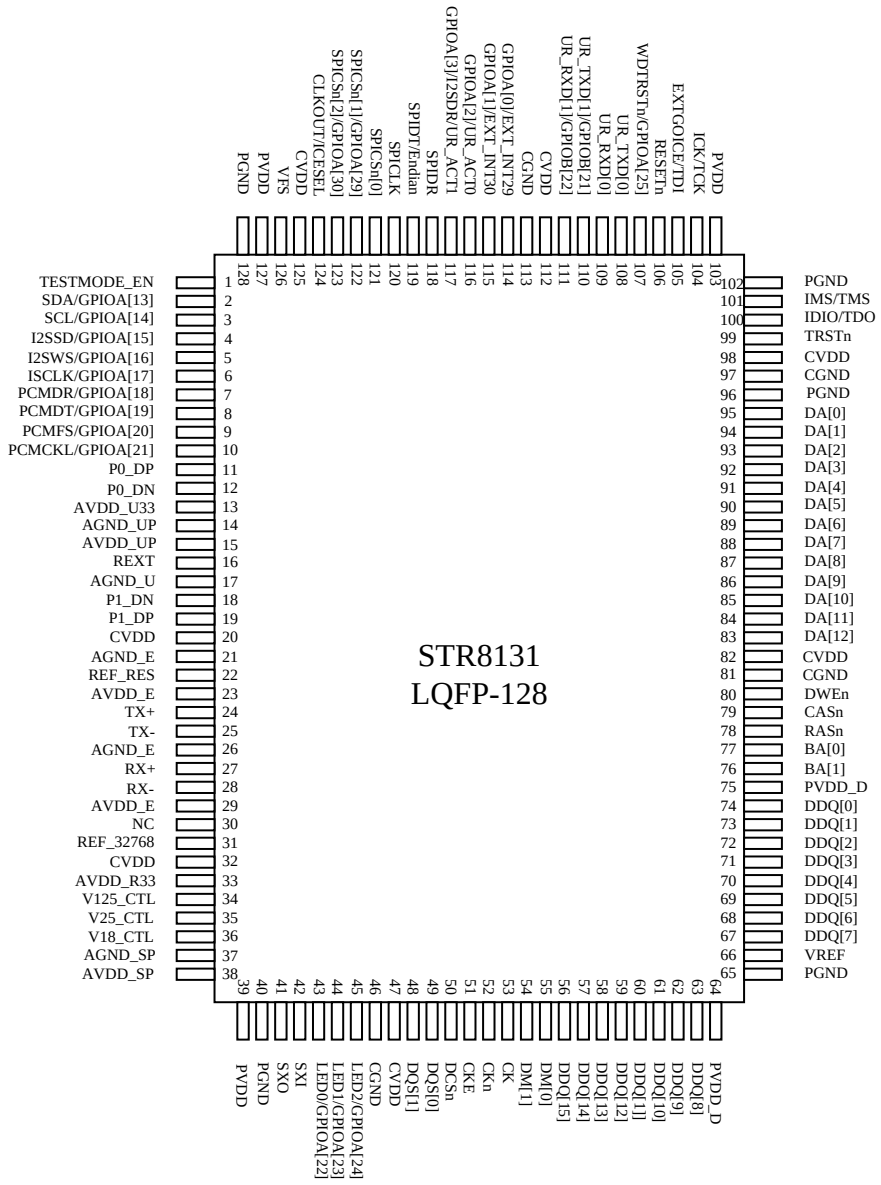


Figure 15. STR8131/CNS2131 Package Pin-out

2.1.2 STR8132/CNS2132 (PQFP-128)

The package pin-out for STR8132/CNS2132 is shown in the following figure.

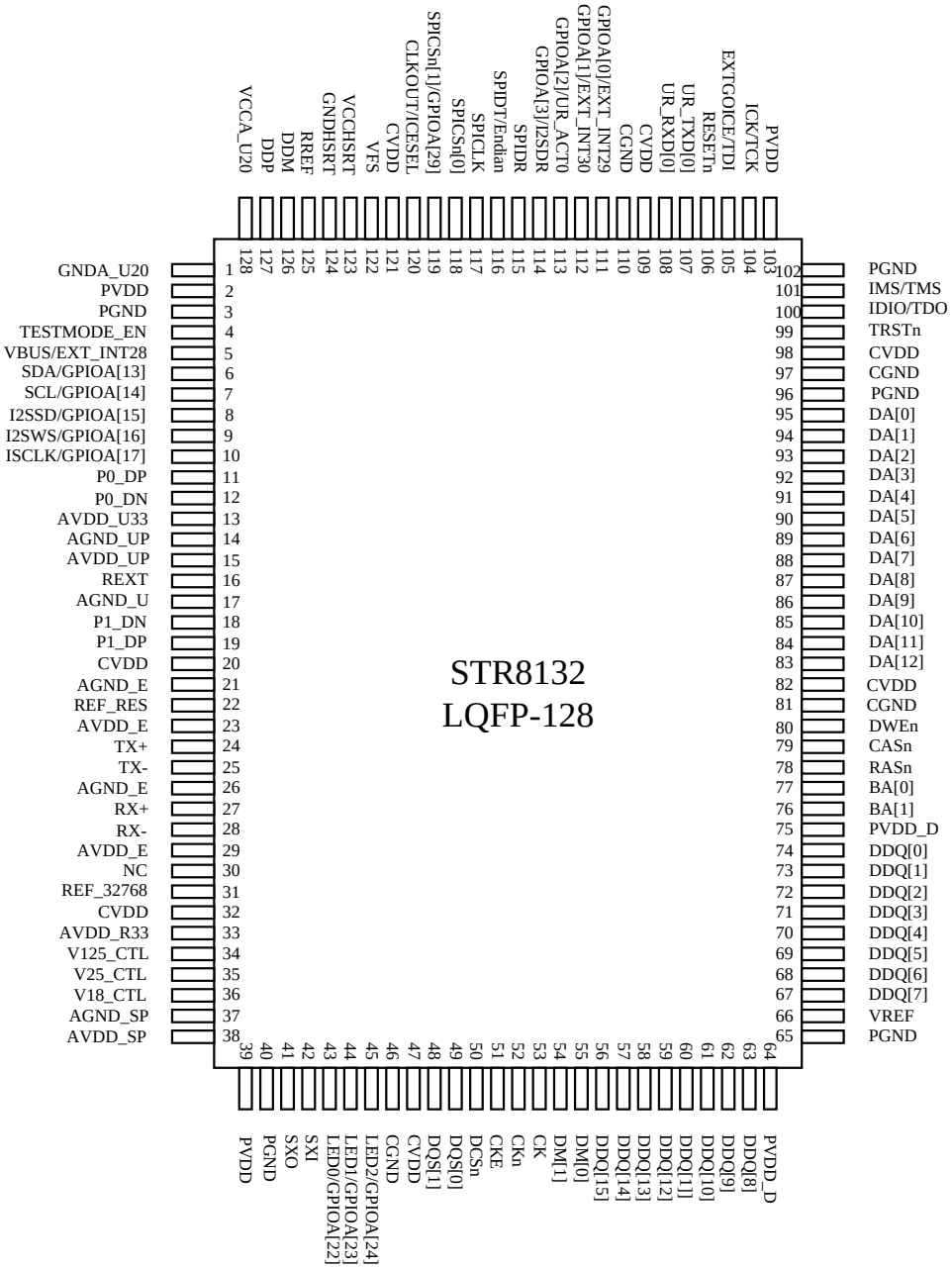


Figure 16. STR8132/CNS2132 Package Pin-out

2.1.3 STR8133/CNS2133 (LFBGA-269)

The package pin out for STR8133/CNS2133 is shown in the following figure.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
A	VCCA_U20	DDM	VCCHSRT	RREF	INTn[0]/EXT_INT5	GNTn[0]/GPIOB[29]	AD[5]	CBEn[0]	AD[9]	AD[15]	PAR	DEVSELn/GPIOB[24]	FRAMEn	CBEn[2]	AD[17]	AD[19]	GND
B	GNDA_U20	DDP	GNDHSRT	GPIOA[1]/EXT_INT30	GNTn[1]/GPIOB[31]	AD[0]	AD[6]	AD[8]	AD[11]	AD[12]	CBEn[1]	PERRn/GPIOB[26]	IRDYn	AD[16]	AD[18]	UR_RXD[0]	UR_TXD[0]
C	VBUS/EXT_INT28	SPIDR/GPIOA[26]	GPIOA[3]/I2SDR/UR_ACT1	SPICSn[1]/GPIOA[29]	INTn[1]/EXT_INT6	AD[1]	REQn[1]/GPIOB[30]	GPIOA[2]/UR_ACT0	AD[3]	AD[4]	AD[13]	STOPn/GPIOB[25]	TRDYn/GPIOB[23]	UR_RXD[1]/GPIOB[22]	ICK/TCK	EXTGOICE/TDI	AD[20]
D	I2SSD/GPIOA[15]	SCL/GPIOA[14]	SDA/GPIOA[13]	CLKOUT/ICESEL	TESTMODE_EN	AD[2]	GPIOA[0]/EXT_INT29	REQn[0]/GPIOB[28]	AD[7]	AD[10]	AD[14]	SERRn/GPIOB[27]	UR_TXD[1]/GPIOB[21]	WDTRSTn/GPIOA[25]	IMS/TMS	AD[21]	AD[22]
E	PCMDR/GPIOA[18]	I2SCLK/GPIOA[17]	I2SWS/GPIOA[16]	SPICLK/GPIOA[27]	AVDD_U33	PVDD	PVDD	GPIOA[4]	GPIOA[5]	GPIOA[6]	VFS	CVDD	CVDD	RESETn	TRSTn	AD[23]	CBEn[3]
F	PCMDT/GPIOA[19]	PCMFS/GPIOA[20]	SPICSn[2]/GPIOA[30]	SPICSn[3]/GPIOA[31]	AVDD_UP	AVDD_U						CVDD	PVDD	IDIO/TDO	AD[24]	AD[25]	AD[26]
G	P0_DP	P0_DN	SPICSn[0]/GPIOA[28]	SPIDT/Endian	AGND_UP		GND	GND	GND	GND	GND		PVDD	AD[30]	AD[29]	AD[28]	AD[27]
H	P1_DN	P1_DP	PCMCLK/GPIOA[21]	REXT	AGND_U		GND	GND	GND	GND	GND		PVDD	PCIRSTn	SWAITn[2]/GPIOB[15]	PCICLK	AD[31]
J	RX-	RX+	GPIOA[7]	LED2/GPIOA[24]	CVDD		GND	GND	GND	GND	GND		PVDD_D	SWAITn[1]/GPIOB[14]	SCEn[1]/GPIOB[11]	SCEn[0]	SCEn[2]/GPIOB[12]
K	TX-	TX+	REF_RES	AGND_E	AGND_E		GND	GND	GND	GND	GND		PVDD_D	DA[6]	DA[4]	SWEn	DA[0]
L	NC	LED1/GPIOA[23]	AGND_E	LED0/GPIOA[22]	AVDD_E		GND	GND	GND	GND	GND		VREF	SOEn	DA[8]	DA[3]	DA[1]
M	REF_32768	V25_CTL	V125_CTL	IDECS0n/GPIOB[7]	AVDD_E	NC						CVDD	CVDD	DA[11]	DA[10]	DA[7]	DA[2]
N	SXO	V18_CTL	DIORn/GPIOB[9]	DIOWn/GPIOB[10]	AVDD_E	CVDD	AGND_R	AVDD_SP	AGND_SP	AVDD_R33	PVDD	PVDD	PVDD	CASn	DA[12]	DDQ[0]	DA[5]
P	SXI	IORDY/GPIOB[3]	DMACKn/GPIOB[5]	INTRQ/GPIOB[6]	SDQ[1]	SDQ[4]	SDQ[3]	SDQ[10]/SA[18]	SA[4]	SA[12]	CKE/SPIBoot	DM[1]	RASn	BA[1]	DA[9]	DDQ[2]	DDQ[1]
R	DMARQ/GPIOB[4]	IDECS1n/GPIOB[8]	SDQ[6]	SDQ[0]	SDQ[2]	SDQ[5]	SDQ[13]/SA[21]	SA[5]	SA[8]	SA[11]	SA[15]	DCSn	DWEn	DM[0]	BA[0]	DDQ[4]	DDQ[3]
T	SDQ[9]/SA[17]	SDQ[12]/SA[20]	SDQ[7]	SDQ[8]/SA[16]	SDQ[11]/SA[19]	SDQ[14]/SA[22]	SA[7]	SA[9]	SA[13]	DQS[1]	DQS[0]	DDQ[14]	DDQ[12]	DDQ[10]	DDQ[8]	DDQ[6]	DDQ[5]
U	GND	SDQ[15]/SA[23]	SA[0]	SA[1]	SA[2]	SA[3]	SA[6]	SA[10]	SA[14]	CKn	CK	DDQ[15]	DDQ[13]	DDQ[11]	DDQ[9]	DDQ[7]	GND

Figure 17. STR8133/CNS2133 Package Pin-out

2.1.4 STR8181/CNS2181 (PQFP-128)

The package pin out for STR8181/CNS2181 is shown in the following figure.

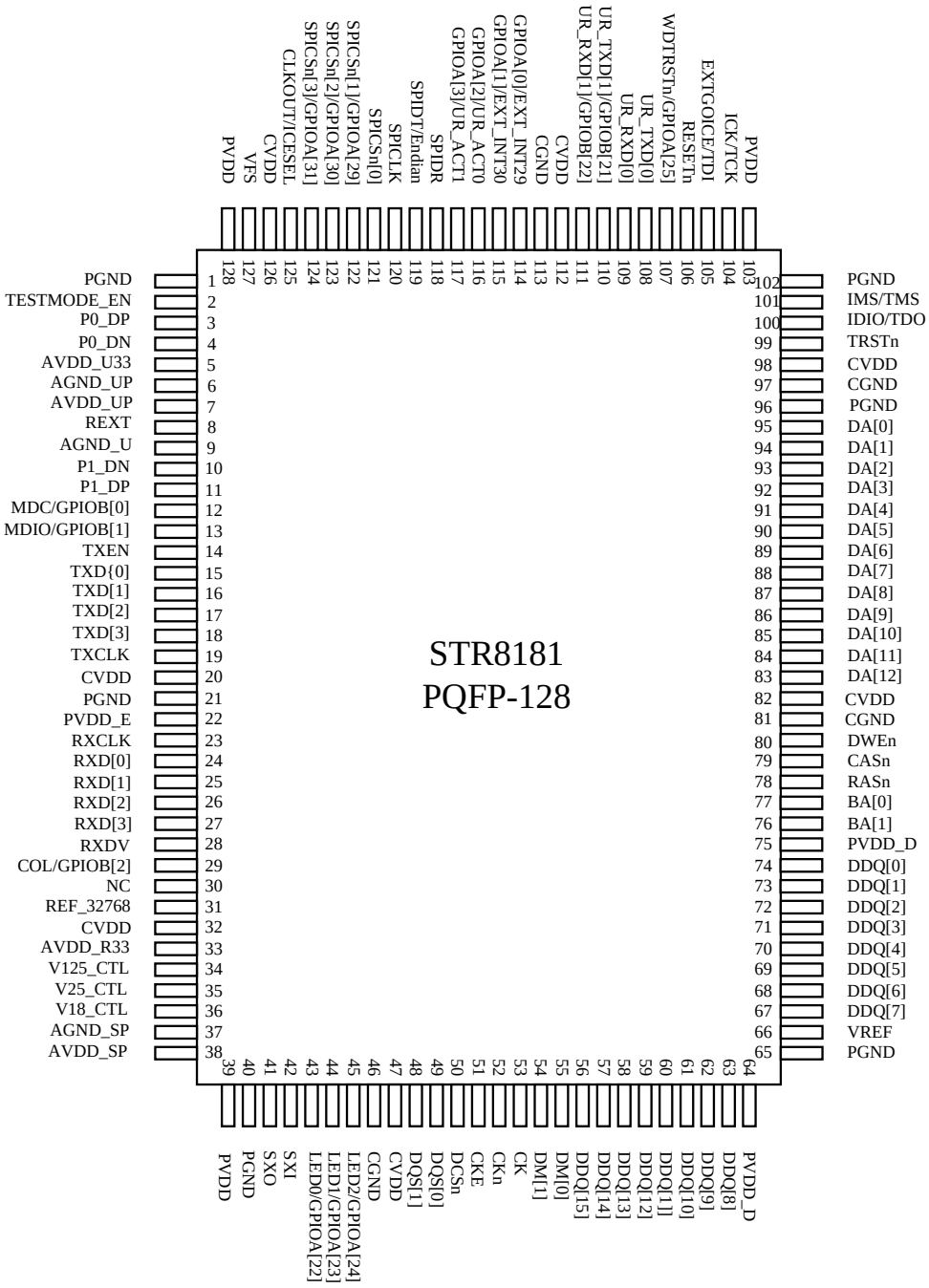


Figure 18. STR8181/CNS2181 Package Pin-out

2.1.5 STR8182/CNS2182 (LFBGA-269)

The package pin out for STR8182/CNS2182 is shown in the following figure.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
A	GND	GPIOA[1]/EXT_INT30	INTn[1]/EXT_INT6	GNTn[0]/GPIOB[29]	AD[0]	AD[1]	AD[5]	CBEn[0]	AD[9]	AD[15]	PAR	DEVSELn/GPIOB[24]	FRAMEn	CBEn[2]	AD[17]	AD[19]	GND
B	SPIDR/GPIOA[26]	GPIOA[4]	GPIOA[3]/I2SDR/UR_ACT1	INTn[0]/EXT_INT5	GNTn[1]/GPIOB[31]	AD[2]	AD[6]	AD[8]	AD[11]	AD[12]	CBEn[1]	PERRn/GPIOB[26]	IRDYn	AD[16]	AD[18]	UR_RXD[0]	UR_TXD[0]
C	SPICSn[2]/GPIOA[30]	SPICSn[0]/GPIOA[28]	SPICLK/GPIOA[27]	GPIOA[5]	SPICSn[1]/GPIOA[29]	SPIDT/Endian	REQn[1]/GPIOB[30]	GPIOA[2]/UR_ACT0	AD[3]	AD[4]	AD[13]	STOPn/GPIOB[25]	TRDYn/GPIOB[23]	UR_RXD[1]/GPIOB[2	ICK/TCK	EXTGOICE/TDI	AD[20]
D	SPICSn[3]/GPIOA[31]	CLKOUT/ICESEL	NC	NC	NC	GPIOA[6]	GPIOA[0]/EXT_INT29	REQn[0]/GPIOB[28]	AD[7]	AD[10]	AD[14]	SERRn/GPIOB[27]	UR_TXD[1]/GPIOB[2	WDTRSTn/GPIOA[25]	IMS/TMS	AD[21]	AD[22]
E	I2SSD/GPIOA[15]	EXT_INT28	SDA/GPIOA[13]	TESTMODE_EN	AVDD_U33	PVDD	NC	GND	GND	NC	VFS	CVDD	CVDD	RESETn	TRSTn	AD[23]	CBEn[3]
F	PCMDT/GPIOA[19]	I2SWS/GPIOA[16]	PCMDR/GPIOA[18]	I2SCLK/GPIOA[17]	AVDD_UP	AVDD_U						CVDD	PVDD	IDIO/TDO	AD[24]	AD[25]	AD[26]
G	P0_DP	P0_DN	PCMFS/GPIOA[20]	SCL/GPIOA[14]	AGND_UP		GND	GND	GND	GND	GND		PVDD	AD[30]	AD[29]	AD[28]	AD[27]
H	P1_DN	P1_DP	MDIO/GPIOB[1]	REXT	AGND_U		GND	GND	GND	GND	GND		PVDD	PCIRSTn	SWAITn[2]/GPIOB[15]	PCICLK	AD[31]
J	TXD[0]	TXEN	TXD[1]	PCMCLK/GPIOA[21]	CVDD		GND	GND	GND	GND	GND		PVDD_D	SWAITn[1]/GPIOB[14]	SCEn[1]/GPIOB[11]	SCEn[0]	SCEn[2]/GPIOB[12]
K	TXCLK	RXDV	RXD[0]	RXD[1]	PVDD_E		GND	GND	GND	GND	GND		PVDD_D	DA[6]	DA[4]	SWEn	DA[0]
L	NC	TXD[3]	RXD[2]	MDC/GPIOB[0]	RXCLK		GND	GND	GND	GND	GND		VREF	SOEn	DA[8]	DA[3]	DA[1]
M	REF_32768	V25_CTL	V125_CTL	IDECS0n/GPIOB[7]	TXD[2]	COL/GPIOB[2]						CVDD	CVDD	DA[11]	DA[10]	DA[7]	DA[2]
N	SXO	V18_CTL	DIORn/GPIOB[9]	DIOWn/GPIOB[10]	RXD[3]	CVDD	AGND_R	AVDD_SP	AGND_SP	AVDD_R33	PVDD	PVDD	PVDD	CASn	DA[12]	DDQ[0]	DA[5]
P	SXI	IORDY/GPIOB[3]	DMACKn/GPIOB[5]	INTRQ/GPIOB[6]	SDQ[1]	SDQ[4]	SDQ[3]	SDQ[10]/SA[18]	SA[4]	SA[12]	CKE/SPIBoot	DM[1]	RASn	BA[1]	DA[9]	DDQ[2]	DDQ[1]
R	DMARQ/GPIOB[4]	IDECS1n/GPIOB[8]	SDQ[6]	SDQ[0]	SDQ[2]	SDQ[5]	SDQ[13]/SA[21]	SA[5]	SA[8]	SA[11]	SA[15]	DCSn	DWEn	DM[0]	BA[0]	DDQ[4]	DDQ[3]
T	SDQ[9]/SA[17]	SDQ[12]/SA[20]	SDQ[7]	SDQ[8]/SA[16]	SDQ[11]/SA[19]	SDQ[14]/SA[22]	SA[7]	SA[9]	SA[13]	DQS[1]	DQS[0]	DDQ[14]	DDQ[12]	DDQ[10]	DDQ[8]	DDQ[6]	DDQ[5]
U	GND	SDQ[15]/SA[23]	SA[0]	SA[1]	SA[2]	SA[3]	SA[6]	SA[10]	SA[14]	CKn	CK	DDQ[15]	DDQ[13]	DDQ[11]	DDQ[9]	DDQ[7]	GND

Figure 19. STR8182/CNS2182 Package Pin-out

2.1.6 STR8182X/CNS2182X (LFBGA-269)

The package pin out for STR8182X/CNS2182X is shown in the following figure.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
A	VCCA_U20	DDM	VCCHSRT	RREF	INTn[0]/EXT_INT5	GNTn[0]/GPIOB[29]	AD[5]	CBEn[0]	AD[9]	AD[15]	PAR	DEVSELn/GPIOB[24]	FRAMEn	CBEn[2]	AD[17]	AD[19]	GND
B	GNDA_U20	DDP	GNDHSRT	GPIOA[1]/EXT_INT30	GNTn[1]/GPIOB[31]	AD[0]	AD[6]	AD[8]	AD[11]	AD[12]	CBEn[1]	PERRn/GPIOB[26]	IRDYn	AD[16]	AD[18]	UR_RXD[0]	UR_TXD[0]
C	VBUS/EXT_INT28	SPIDR/GPIOA[26]	GPIOA[3]/I2SDR/UR_ACT1	SPICSn[1]/GPIOA[29]	INTn[1]/EXT_INT6	AD[1]	REQn[1]/GPIOB[30]	GPIOA[2]/UR_ACT0	AD[3]	AD[4]	AD[13]	STOPn/GPIOB[25]	TRDYn/GPIOB[23]	UR_RXD[1]/GPIOB[2	ICK/TCK	EXTGOICE/TDI	AD[20]
D	I2SSD/GPIOA[15]	SCL/GPIOA[14]	SDA/GPIOA[13]	CLKOUT/ICESEL	TESTMODE_EN	AD[2]	GPIOA[0]/EXT_INT29	REQn[0]/GPIOB[28]	AD[7]	AD[10]	AD[14]	SERRn/GPIOB[27]	UR_TXD[1]/GPIOB[2	WDTRSTn/GPIOA[25]	IMS/TMS	AD[21]	AD[22]
E	PCMDR/GPIOA[18]	I2SCLK/GPIOA[17]	I2SWS/GPIOA[16]	SPICLK/GPIOA[27]	AVDD_U33	PVDD	PVDD	GPIOA[4]	GPIOA[5]	GPIOA[6]	VFS	CVDD	CVDD	RESETn	TRSTn	AD[23]	CBEn[3]
F	PCMDT/GPIOA[19]	PCMFS/GPIOA[20]	SPICSn[2]/GPIOA[30]	SPICSn[3]/GPIOA[31]	AVDD_UP	AVDD_U						CVDD	PVDD	IDIO/TDO	AD[24]	AD[25]	AD[26]
G	P0_DP	P0_DN	SPICSn[0]/GPIOA[28]	SPIDT/Endian	AGND_UP		GND	GND	GND	GND	GND		PVDD	AD[30]	AD[29]	AD[28]	AD[27]
H	P1_DN	P1_DP	PCMCLK/GPIOA[21]	REXT	AGND_U		GND	GND	GND	GND	GND		PVDD	PCIRSTn	WAITn[2]/GPIOB[15]	PCICLK	AD[31]
J	TXD[0]	TXEN	TXD[1]	MDIO/GPIOB[1]	CVDD		GND	GND	GND	GND	GND		PVDD_D	WAITn[1]/GPIOB[14]	SCEn[1]/GPIOB[11]	SCEn[0]	SCEn[2]/GPIOB[12]
K	TXCLK	RXDV	RXD[0]	RXD[1]	PVDD_E		GND	GND	GND	GND	GND		PVDD_D	DA[6]	DA[4]	SWEn	DA[0]
L	RXCLK	TXD[3]	RXD[2]	MDC/GPIOB[0]	COL/GPIOB[2]		GND	GND	GND	GND	GND		VREF	SOEn	DA[8]	DA[3]	DA[1]
M	REF_32768	V25_CTL	V125_CTL	IDECS0n/GPIOB[7]	TXD[2]	NC						CVDD	CVDD	DA[11]	DA[10]	DA[7]	DA[2]
N	SXO	V18_CTL	DIORn/GPIOB[9]	DIOWn/GPIOB[10]	RXD[3]	CVDD	AGND_R	AVDD_SP	AGND_SP	AVDD_R33	PVDD	PVDD	PVDD	CASn	DA[12]	DDQ[0]	DA[5]
P	SXI	IORDY/GPIOB[3]	DMACKn/GPIOB[5]	INTRQ/GPIOB[6]	SDQ[1]	SDQ[4]	SDQ[3]	SDQ[10]/SA[18]	SA[4]	SA[12]	CKE/SPIBoot	DM[1]	RASn	BA[1]	DA[9]	DDQ[2]	DDQ[1]
R	DMARQ/GPIOB[4]	IDECS1n/GPIOB[8]	SDQ[6]	SDQ[0]	SDQ[2]	SDQ[5]	SDQ[13]/SA[21]	SA[5]	SA[8]	SA[11]	SA[15]	DCSn	DWEn	DM[0]	BA[0]	DDQ[4]	DDQ[3]
T	SDQ[9]/SA[17]	SDQ[12]/SA[20]	SDQ[7]	SDQ[8]/SA[16]	SDQ[11]/SA[19]	SDQ[14]/SA[22]	SA[7]	SA[9]	SA[13]	DQS[1]	DQS[0]	DDQ[14]	DDQ[12]	DDQ[10]	DDQ[8]	DDQ[6]	DDQ[5]
U	GND	SDQ[15]/SA[23]	SA[0]	SA[1]	SA[2]	SA[3]	SA[6]	SA[10]	SA[14]	CKn	CK	DDQ[15]	DDQ[13]	DDQ[11]	DDQ[9]	DDQ[7]	GND

Note: The pin name in red color shows the differen between STR8182/CNS2182 and STR8182X/CNS2182X.

Figure 20. STR8182X/CNS2182X Package Pin-out

2.1.7 CNS2135 (PQFP-128) Package Pin-Out and Pin Name

The package pin out for CNS2135 is shown in the following figure.

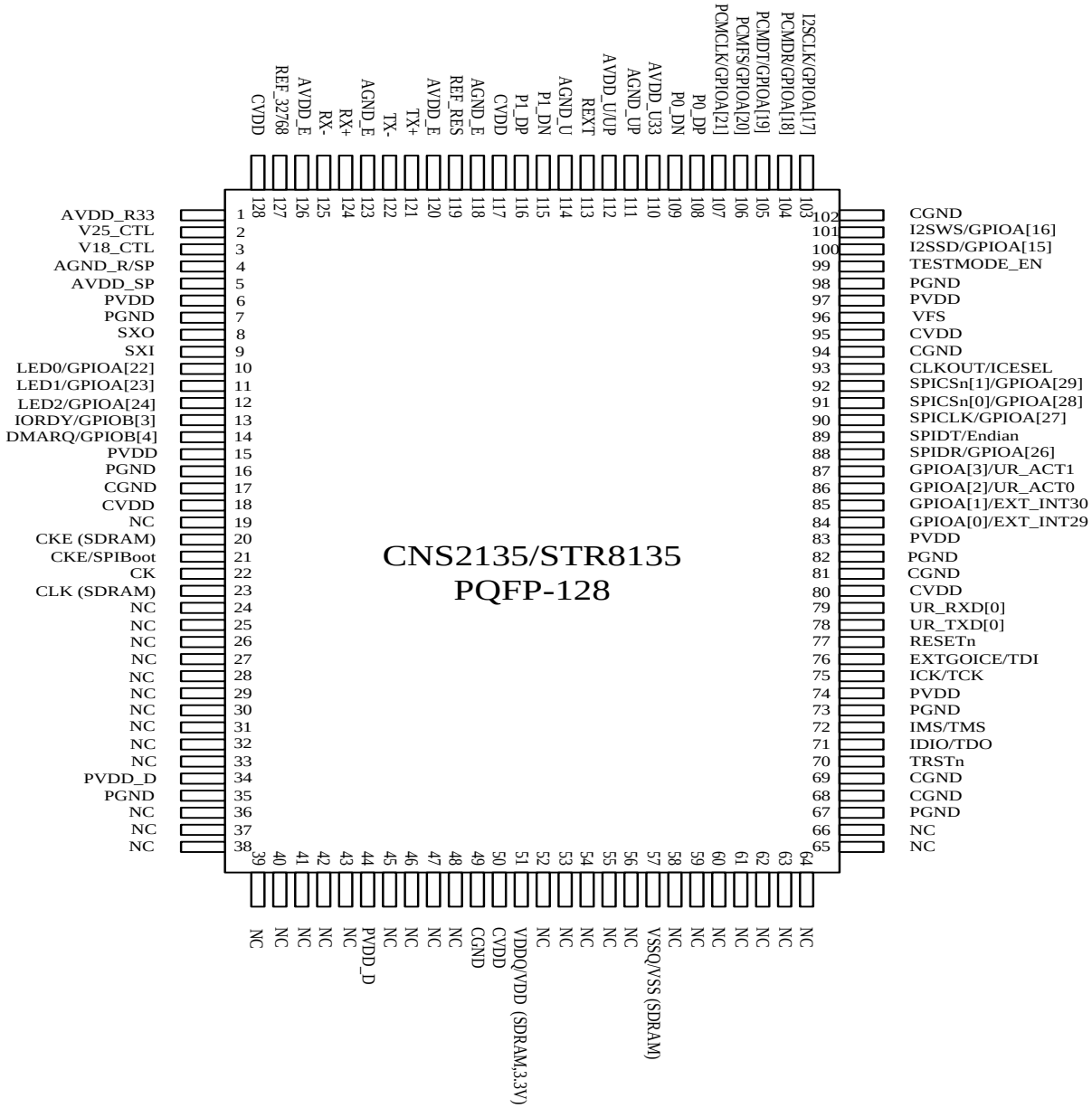


Figure 21. CNS2135 Package Pin-out

2.2 Package Pin-Number vs. Pin-Name

2.2.1 STR8131/CNS2131 (QFP-128)

Table 16. List of Pin Name and Pin Number for STR8131/CNS2131 Package

No	Pin Name	No	Pin Name	No	Pin Name	No	Pin Name
1	TESTMODE_EN	33	AVDD_R33	65	PGND	97	CGND
2	SDA/GPIOA[13]	34	V125_CTL	66	VREF	98	CVDD
3	SCL/GPIOA[14]	35	V25_CTL	67	DDQ[7]	99	TRSTn
4	I2SSD/GPIOA[15]	36	V18_CTL	68	DDQ[6]	100	IDIO/TDO
5	I2SWS/GPIOA[16]	37	AGND_R/SP	69	DDQ[5]	101	IMS/TMS
6	I2SCLK/GPIOA[17]	38	AVDD_SP	70	DDQ[4]	102	PGND
7	PCMDR/GPIOA[18]	39	PVDD	71	DDQ[3]	103	PVDD
8	PCMDT/GPIOA[19]	40	PGND	72	DDQ[2]	104	ICK/TCK
9	PCMFS/GPIOA[20]	41	SXO	73	DDQ[1]	105	EXTGOICE/TDI
10	PCMCLK/GPIOA[21]	42	SXI	74	DDQ[0]	106	RESETn
11	P0_DP	43	LED0/GPIOA[22]	75	PVDD_D	107	WDTRSTn/GPIOA[25]
12	P0_DN	44	LED1/GPIOA[23]	76	BA[1]	108	UR_TXD[0]
13	AVDD_U33	45	LED2/GPIOA[24]	77	BA[0]	109	UR_RXD[0]
14	AGND_UP	46	CGND	78	RASn	110	UR_TXD[1]/GPIOB[21]
15	AVDD_U/UP	47	CVDD	79	CASn	111	UR_RXD[1]/GPIOB[22]
16	REXT	48	DQS[1]	80	DWEn	112	CVDD
17	AGND_U	49	DQS[0]	81	CGND	113	CGND
18	P1_DN	50	DCSn	82	CVDD	114	GPIOA[0]/EXT_INT29
19	P1_DP	51	CKE/SPIBoot	83	DA[12]	115	GPIOA[1]/EXT_INT30
20	CVDD	52	CKn	84	DA[11]	116	GPIOA[2]/UR_ACT0
21	AGND_E	53	CK	85	DA[10]	117	GPIOA[3]/I2SDR/ UR_ACT1
22	REF_RES	54	DM[1]	86	DA[9]	118	SPIDR
23	AVDD_E	55	DM[0]	87	DA[8]	119	SPIDT/Endian
24	TX+	56	DDQ[15]	88	DA[7]	120	SPICLK
25	TX-	57	DDQ[14]	89	DA[6]	121	SPICSn[0]
26	AGND_E	58	DDQ[13]	90	DA[5]	122	SPICSn[1]/GPIOA[29]
27	RX+	59	DDQ[12]	91	DA[4]	123	SPICSn[2]/GPIOA[30]
28	RX-	60	DDQ[11]	92	DA[3]	124	CLKOUT/ICESEL
29	AVDD_E	61	DDQ[10]	93	DA[2]	125	CVDD
30	NC	62	DDQ[9]	94	DA[1]	126	VFS
31	REF_32768	63	DDQ[8]	95	DA[0]	127	PVDD
32	CVDD	64	PVDD_D	96	PGND	128	PGND

2.2.2 STR8132/CNS2132 (QFP-128)

Table 17. List of Pin Name and Pin Number for STR8132/CNS2132 Package

No	Pin Name	No	Pin Name	No	Pin Name	No	Pin Name
1	GNDA_U20	33	AVDD_R33	65	PGND	97	CGND
2	PVDD	34	V125_CTL	66	VREF	98	CVDD
3	PGND	35	V25_CTL	67	DDQ[7]	99	TRSTn
4	TESTMODE_EN	36	V18_CTL	68	DDQ[6]	100	IDIO/TDO
5	VBUS/EXT_INT28	37	AGND_R/SP	69	DDQ[5]	101	IMS/TMS
6	SDA/GPIOA[13]	38	AVDD_SP	70	DDQ[4]	102	PGND
7	SCL/GPIOA[14]	39	PVDD	71	DDQ[3]	103	PVDD
8	I2SSD/GPIOA[15]	40	PGND	72	DDQ[2]	104	ICK/TCK
9	I2SWS/GPIOA[16]	41	SXO	73	DDQ[1]	105	EXTGOICE/TDI
10	I2SCLK/GPIOA[17]	42	SXI	74	DDQ[0]	106	RESETn
11	P0_DP	43	LED0/GPIOA[22]	75	PVDD_D	107	UR_TXD[0]
12	P0_DN	44	LED1/GPIOA[23]	76	BA[1]	108	UR_RXD[0]
13	AVDD_U33	45	LED2/GPIOA[24]	77	BA[0]	109	CVDD
14	AGND_UP	46	CGND	78	RASn	110	CGND
15	AVDD_U/UP	47	CVDD	79	CASn	111	GPIOA[0]/EXT_INT29
16	REXT	48	DQS[1]	80	DWEn	112	GPIOA[1]/EXT_INT30
17	AGND_U	49	DQS[0]	81	CGND	113	GPIOA[2]/UR_ACT0
18	P1_DN	50	DCSn	82	CVDD	114	GPIOA[3]/I2SDR
19	P1_DP	51	CKE/SPIBoot	83	DA[12]	115	SPIDR
20	CVDD	52	CKn	84	DA[11]	116	SPIDT/Endian
21	AGND_E	53	CK	85	DA[10]	117	SPICLK
22	REF_RES	54	DM[1]	86	DA[9]	118	SPICSn[0]
23	AVDD_E	55	DM[0]	87	DA[8]	119	SPICSn[1]/GPIOA[29]
24	TX+	56	DDQ[15]	88	DA[7]	120	CLKOUT/ICESEL
25	TX-	57	DDQ[14]	89	DA[6]	121	CVDD
26	AGND_E	58	DDQ[13]	90	DA[5]	122	VFS
27	RX+	59	DDQ[12]	91	DA[4]	123	VCCHSRT
28	RX-	60	DDQ[11]	92	DA[3]	124	GNDHSRT
29	AVDD_E	61	DDQ[10]	93	DA[2]	125	RREF
30	NC	62	DDQ[9]	94	DA[1]	126	DDM
31	REF_32768	63	DDQ[8]	95	DA[0]	127	DDP
32	CVDD	64	PVDD_D	96	PGND	128	VCCA_U20

2.2.3 STR8133/CNS2133 (LFBGA-269)

Table 18. List of Pin Name and Pin Number for STR8133/CNS2133 Package

No	Pin Name	No	Pin Name	No	Pin Name	No	Pin Name
A1	VCCA_U20	B1	GNDA_U20	C1	VBUS/EXT_INT28	D1	I2SSD/GPIOA[15]
A2	DDM	B2	DDP	C2	SPIDR/GPIOA[26]	D2	SCL/GPIOA[14]
A3	VCCHSRT	B3	GNDHSRT	C3	GPIOA[3]/I2SDR/ UR_ACT1	D3	SDA/GPIOA[13]
A4	RREF	B4	GPIOA[1]/EXT_INT 30	C4	SPICSn[1]/GPIOA[29]	D4	CLKOUT/ICESEL
A5	INTn[0]/EXT_INT5	B5	GNTn[1]/GPIOB[31]	C5	INTn[1]/EXT_INT6	D5	TESTMODE_EN
A6	GNTn[0]/GPIOB[29]	B6	AD[0]	C6	AD[1]	D6	AD[2]
A7	AD[5]	B7	AD[6]	C7	REQn[1]/GPIOB[30]	D7	GPIOA[0]/EXT_INT29
A8	CBEn[0]	B8	AD[8]	C8	GPIOA[2]/UR_ACT	D8	REQn[0]/GPIOB[28]

					0		
A9	AD[9]	B9	AD[11]	C9	AD[3]	D9	AD[7]
A10	AD[15]	B10	AD[12]	C10	AD[4]	D10	AD[10]
A11	PAR	B11	CBEn[1]	C11	AD[13]	D11	AD[14]
A12	DEVSELn/GPIOB[24]	B12	PERRn/GPIOB[26]	C12	STOPn/GPIOB[25]	D12	SERRn/GPIOB[27]
A13	FRAMEn	B13	IRDYn	C13	TRDYn/GPIOB[23]	D13	UR_TXD[1]/GPIOB[21]
A14	CBEn[2]	B14	AD[16]	C14	UR_RXD[1]/GPIOB[22]	D14	WDTRSTn/GPIOA[25]
A15	AD[17]	B15	AD[18]	C15	ICK/TCK	D15	IMS/TMS
A16	AD[19]	B16	UR_RXD[0]	C16	EXTGOICE/TDI	D16	AD[21]
A17	GND	B17	UR_TXD[0]	C17	AD[20]	D17	AD[22]
No	Pin Name	No	Pin Name	No	Pin Name	No	Pin Name
E1	PCMDR/GPIOA[18]	F1	PCMDT/GPIOA[19]	G1	P0_DP	H1	P1_DN
E2	I2SCLK/GPIOA[17]	F2	PCMFS/GPIOA[20]	G2	P0_DN	H2	P1_DP
E3	I2SWS/GPIOA[16]	F3	SPICSn[2]/GPIOA[30]	G3	SPICSn[0]/GPIOA[28]	H3	PCMCLK/GPIOA[21]
E4	SPICLK/GPIOA[27]	F4	SPICSn[3]/GPIOA[31]	G4	SPIDT/Endian	H4	REXT
E5	AVDD_U33	F5	AVDD_UP	G5	AGND_UP	H5	AGND_U
E6	PVDD	F6	AVDD_U	G6		H6	
E7	PVDD	F7		G7	GND	H7	GND
E8	GPIOA[4]	F8		G8	GND	H8	GND
E9	GPIOA[5]	F9		G9	GND	H9	GND
E10	GPIOA[6]	F10		G10	GND	H10	GND
E11	VFS	F11		G11	GND	H11	GND
E12	CVDD	F12	CVDD	G12		H12	
E13	CVDD	F13	PVDD	G13	PVDD	H13	PVDD
E14	RESETn	F14	IDIO/TDO	G14	AD[30]	H14	PCIRSTn
E15	TRSTn	F15	AD[24]	G15	AD[29]	H15	WAITn[2]/GPIOB[15]
E16	AD[23]	F16	AD[25]	G16	AD[28]	H16	PCICLK
E17	CBEn[3]	F17	AD[26]	G17	AD[27]	H17	AD[31]
No	Pin Name	No	Pin Name	No	Pin Name	No	Pin Name
J1	RX-	K1	TX-	L1	NC	M1	REF_32768
J2	RX+	K2	TX+	L2	LED1/GPIOA[23]	M2	V25_CTL
J3	GPIOA[7]	K3	REF_RES	L3	AGND_E	M3	V125_CTL
J4	LED2/GPIOA[24]	K4	AGND_E	L4	LED0/GPIOA[22]	M4	IDECS0n/GPIOB[7]
J5	CVDD	K5	AGND_E	L5	AVDD_E	M5	AVDD_E
J6		K6		L6		M6	NC
J7	GND	K7	GND	L7	GND	M7	
J8	GND	K8	GND	L8	GND	M8	
J9	GND	K9	GND	L9	GND	M9	
J10	GND	K10	GND	L10	GND	M10	
J11	GND	K11	GND	L11	GND	M11	
J12		K12		L12		M12	CVDD
J13	PVDD_D	K13	PVDD_D	L13	VREF	M13	CVDD
J14	WAITn[1]/GPIOB[14]	K14	DA[6]	L14	SOEn	M14	DA[11]
J15	SCEn[1]/GPIOB[11]	K15	DA[4]	L15	DA[8]	M15	DA[10]
J16	SCEn[0]	K16	SWEn	L16	DA[3]	M16	DA[7]
J17	SCEn[2]/GPIOB[12]	K17	DA[0]	L17	DA[1]	M17	DA[2]
No	Pin Name	No	Pin Name	No	Pin Name	No	Pin Name
N1	SXO	P1	SXI	R1	DMARQ/GPIOB[4]	T1	SDQ[9]/SA[17]
N2	V18_CTL	P2	IORDY/GPIOB[3]	R2	IDECS1n/GPIOB[8]	T2	SDQ[12]/SA[20]

N3	DIORn/GPIOB[9]	P3	DMACKn/GPIOB[5]	R3	SDQ[6]	T3	SDQ[7]
N4	DIOWn/GPIOB[10]	P4	INTRQ/GPIOB[6]	R4	SDQ[0]	T4	SDQ[8]/SA[16]
N5	AVDD_E	P5	SDQ[1]	R5	SDQ[2]	T5	SDQ[11]/SA[19]
N6	CVDD	P6	SDQ[4]	R6	SDQ[5]	T6	SDQ[14]/SA[22]
N7	AGND_R	P7	SDQ[3]	R7	SDQ[13]/SA[21]	T7	SA[7]
N8	AVDD_SP	P8	SDQ[10]/SA[18]	R8	SA[5]	T8	SA[9]
N9	AGND_SP	P9	SA[4]	R9	SA[8]	T9	SA[13]
N10	AVDD_R33	P10	SA[12]	R10	SA[11]	T10	DQS[1]
N11	PVDD	P11	CKE/SPIBoot	R11	SA[15]	T11	DQS[0]
N12	PVDD	P12	DM[1]	R12	DCSn	T12	DDQ[14]
N13	PVDD	P13	RASn	R13	DWEn	T13	DDQ[12]
N14	CASn	P14	BA[1]	R14	DM[0]	T14	DDQ[10]
N15	DA[12]	P15	DA[9]	R15	BA[0]	T15	DDQ[8]
N16	DDQ[0]	P16	DDQ[2]	R16	DDQ[4]	T16	DDQ[6]
N17	DA[5]	P17	DDQ[1]	R17	DDQ[3]	T17	DDQ[5]
No	Pin Name						
U1	GND						
U2	SDQ[15]/SA[23]						
U3	SA[0]						
U4	SA[1]						
U5	SA[2]						
U6	SA[3]						
U7	SA[6]						
U8	SA[10]						
U9	SA[14]						
U10	CKn						
U11	CK						
U12	DDQ[15]						
U13	DDQ[13]						
U14	DDQ[11]						
U15	DDQ[9]						
U16	DDQ[7]						
U17	GND						

2.2.4 STR8181/CNS2181 (QFP-128)

Table 19. List of Pin Name and Pin Number for STR8181/CNS2181 Package

No	Pin Name	No	Pin Name	No	Pin Name	No	Pin Name
1	PGND	33	AVDD_R33	65	PGND	97	CGND
2	TESTMODE_EN	34	V125_CTL	66	VREF	98	CVDD
3	P0_DP	35	V25_CTL	67	DDQ[7]	99	TRSTn
4	P0_DN	36	V18_CTL	68	DDQ[6]	100	IDIO/TDO
5	AVDD_U33	37	AGND_R/SP	69	DDQ[5]	101	IMS/TMS
6	AGND_UP	38	AVDD_SP	70	DDQ[4]	102	PGND
7	AVDD_U/UP	39	PVDD	71	DDQ[3]	103	PVDD
8	REXT	40	PGND	72	DDQ[2]	104	ICK/TCK
9	AGND_U	41	SXO	73	DDQ[1]	105	EXTGOICE/TDI
10	P1_DN	42	SXI	74	DDQ[0]	106	RESETn
11	P1_DP	43	LED0/GPIOA[22]	75	PVDD_D	107	WDTRSTn/GPIOA[25]
12	MDC/GPIOB[0]	44	LED1/GPIOA[23]	76	BA[1]	108	UR_TXD[0]
13	MDIO/GPIOB[1]	45	LED2/GPIOA[24]	77	BA[0]	109	UR_RXD[0]
14	TXEN	46	CGND	78	RASn	110	UR_TXD[1]/GPIOB[21]
15	TXD[0]	47	CVDD	79	CASn	111	UR_RXD[1]/GPIOB[22]

16	TXD[1]	48	DQS[1]	80	DWEn	112	CVDD
17	TXD[2]	49	DQS[0]	81	CGND	113	CGND
18	TXD[3]	50	DCSn	82	CVDD	114	GPIOA[0]/EXT_INT29
19	TXCLK	51	CKE/SPIBoot	83	DA[12]	115	GPIOA[1]/EXT_INT30
20	CVDD	52	CKn	84	DA[11]	116	GPIOA[2]/UR_ACT0
21	PGND	53	CK	85	DA[10]	117	GPIOA[3]/UR_ACT1
22	PVDD_E	54	DM[1]	86	DA[9]	118	SPIDR
23	RXCLK	55	DM[0]	87	DA[8]	119	SPIDT/Endian
24	RXD[0]	56	DDQ[15]	88	DA[7]	120	SPICLK
25	RXD[1]	57	DDQ[14]	89	DA[6]	121	SPICSn[0]
26	RXD[2]	58	DDQ[13]	90	DA[5]	122	SPICSn[1]/GPIOA[29]
27	RXD[3]	59	DDQ[12]	91	DA[4]	123	SPICSn[2]/GPIOA[30]
28	RXDV	60	DDQ[11]	92	DA[3]	124	SPICSn[3]/GPIOA[31]
29	COL/GPIOB[2]	61	DDQ[10]	93	DA[2]	125	CLKOUT/ICESEL
30	NC	62	DDQ[9]	94	DA[1]	126	CVDD
31	REF_32768	63	DDQ[8]	95	DA[0]	127	VFS
32	CVDD	64	PVDD_D	96	PGND	128	PVDD

2.2.5 STR8182/CNS2182 (LFBGA-269)

Table 20. List of Pin Name and Pin Number for STR8182/CNS2182 Package

No	Pin Name	No	Pin Name	No	Pin Name	No	Pin Name
A1	GND	B1	SPIDR/GPIOA[26]	C1	SPICSn[2]/GPIOA[30]	D1	SPICSn[3]/GPIOA[31]
A2	GPIOA[1]/EXT_INT30	B2	GPIOA[4]	C2	SPICSn[0]/GPIOA[28]	D2	CLKOUT/ICESEL
A3	INTn[1]/EXT_INT	B3	GPIOA[3]/I2SDR/UR_ACT1	C3	SPICLK/GPIOA[27]	D3	NC
A4	GNTn[0]/GPIOB[29]	B4	INTn[0]/EXT_INT	C4	GPIOA[5]	D4	NC
A5	AD[0]	B5	GNTn[1]/GPIOB[31]	C5	SPICSn[1]/GPIOA[29]	D5	NC
A6	AD[1]	B6	AD[2]	C6	SPIDT/Endian	D6	GPIOA[6]
A7	AD[5]	B7	AD[6]	C7	REQn[1]/GPIOB[30]	D7	GPIOA[0]/EXT_INT29
A8	CBEn[0]	B8	AD[8]	C8	GPIOA[2]/UR_ACT0	D8	REQn[0]/GPIOB[28]
A9	AD[9]	B9	AD[11]	C9	AD[3]	D9	AD[7]
A10	AD[15]	B10	AD[12]	C10	AD[4]	D10	AD[10]
A11	PAR	B11	CBEn[1]	C11	AD[13]	D11	AD[14]
A12	DEVSELn/GPIOB[24]	B12	PERRn/GPIOB[26]	C12	STOPn/GPIOB[25]	D12	SERRn/GPIOB[27]
A13	FRAMEn	B13	IRDYn	C13	TRDYn/GPIOB[23]	D13	UR_TXD[1]/GPIOB[21]
A14	CBEn[2]	B14	AD[16]	C14	UR_RXD[1]/GPIOB[22]	D14	WDTRSTn/GPIOA[25]
A15	AD[17]	B15	AD[18]	C15	ICK/TCK	D15	IMS/TMS
A16	AD[19]	B16	UR_RXD[0]	C16	EXTGOICE/TDI	D16	AD[21]
A17	GND	B17	UR_TXD[0]	C17	AD[20]	D17	AD[22]
No	Pin Name	No	Pin Name	No	Pin Name	No	Pin Name
E1	I2SSD/GPIOA[15]	F1	PCMDT/GPIOA[19]	G1	P0_DP	H1	P1_DN
E2	EXT_INT28	F2	I2SWS/GPIOA[16]	G2	P0_DN	H2	P1_DP
E3	SDA/GPIOA[13]	F3	PCMDR/GPIOA[18]	G3	PCMFS/GPIOA[20]	H3	MDIO/GPIOB[1]
E4	TESTMODE_EN	F4	I2SCLK/GPIOA[17]	G4	SCL/GPIOA[14]	H4	REXT
E5	AVDD_U33	F5	AVDD_UP	G5	AGND_UP	H5	AGND_U
E6	PVDD	F6	AVDD_U	G6		H6	
E7	NC	F7		G7	GND	H7	GND

E8	GND	F8		G8	GND	H8	GND
E9	GND	F9		G9	GND	H9	GND
E10	NC	F10		G10	GND	H10	GND
E11	VFS	F11		G11	GND	H11	GND
E12	CVDD	F12	CVDD	G12		H12	
E13	CVDD	F13	PVDD	G13	PVDD	H13	PVDD
E14	RESETn	F14	IDIO/TDO	G14	AD[30]	H14	PCIRSTn
E15	TRSTn	F15	AD[24]	G15	AD[29]	H15	WAITn[2]/GPIOB[15]
E16	AD[23]	F16	AD[25]	G16	AD[28]	H16	PCICLK
E17	CBEn[3]	F17	AD[26]	G17	AD[27]	H17	AD[31]
No	Pin Name	No	Pin Name	No	Pin Name	No	Pin Name
J1	TXD[0]	K1	TXCLK	L1	NC	M1	REF_32768
J2	TXEN	K2	RXDV	L2	TXD[3]	M2	V25_CTL
J3	TXD[1]	K3	RXD[0]	L3	RXD[2]	M3	V125_CTL
J4	PCMCLK/GPIOA[21]	K4	RXD[1]	L4	MDC/GPIOB[0]	M4	IDECS0n/GPIOB[7]
J5	CVDD	K5	PVDD_E	L5	RXCLK	M5	TXD[2]
J6		K6		L6		M6	COL/GPIOB[2]
J7	GND	K7	GND	L7	GND	M7	
J8	GND	K8	GND	L8	GND	M8	
J9	GND	K9	GND	L9	GND	M9	
J10	GND	K10	GND	L10	GND	M10	
J11	GND	K11	GND	L11	GND	M11	
J12		K12		L12		M12	CVDD
J13	PVDD_D	K13	PVDD_D	L13	VREF	M13	CVDD
J14	WAITn[1]/GPIOB[14]	K14	DA[6]	L14	SOEn	M14	DA[11]
J15	SCEn[1]/GPIOB[11]	K15	DA[4]	L15	DA[8]	M15	DA[10]
J16	SCEn[0]	K16	SWEn	L16	DA[3]	M16	DA[7]
J17	SCEn[2]/GPIOB[12]	K17	DA[0]	L17	DA[1]	M17	DA[2]
No	Pin Name	No	Pin Name	No	Pin Name	No	Pin Name
N1	SXO	P1	SXI	R1	DMARQ/GPIOB[4]	T1	SDQ[9]/SA[17]
N2	V18_CTL	P2	IORDY/GPIOB[3]	R2	IDECS1n/GPIOB[8]	T2	SDQ[12]/SA[20]
N3	DIORn/GPIOB[9]	P3	DMACKn/GPIOB[5]	R3	SDQ[6]	T3	SDQ[7]
N4	DIOWn/GPIOB[10]	P4	INTRQ/GPIOB[6]	R4	SDQ[0]	T4	SDQ[8]/SA[16]
N5	RXD[3]	P5	SDQ[1]	R5	SDQ[2]	T5	SDQ[11]/SA[19]
N6	CVDD	P6	SDQ[4]	R6	SDQ[5]	T6	SDQ[14]/SA[22]
N7	AGND_R	P7	SDQ[3]	R7	SDQ[13]/SA[21]	T7	SA[7]
N8	AVDD_SP	P8	SDQ[10]/SA[18]	R8	SA[5]	T8	SA[9]
N9	AGND_SP	P9	SA[4]	R9	SA[8]	T9	SA[13]
N10	AVDD_R33	P10	SA[12]	R10	SA[11]	T10	DQS[1]
N11	PVDD	P11	CKE/SPIBoot	R11	SA[15]	T11	DQS[0]
N12	PVDD	P12	DM[1]	R12	DCSn	T12	DDQ[14]
N13	PVDD	P13	RASn	R13	DWEn	T13	DDQ[12]
N14	CASn	P14	BA[1]	R14	DM[0]	T14	DDQ[10]
N15	DA[12]	P15	DA[9]	R15	BA[0]	T15	DDQ[8]
N16	DDQ[0]	P16	DDQ[2]	R16	DDQ[4]	T16	DDQ[6]
N17	DA[5]	P17	DDQ[1]	R17	DDQ[3]	T17	DDQ[5]
No	Pin Name						
U1	GND						
U2	SDQ[15]/SA[23]						
U3	SA[0]						
U4	SA[1]						
U5	SA[2]						
U6	SA[3]						
U7	SA[6]						

U8	SA[10]						
U9	SA[14]						
U10	CKn						
U11	CK						
U12	DDQ[15]						
U13	DDQ[13]						
U14	DDQ[11]						
U15	DDQ[9]						
U16	DDQ[7]						
U17	GND						

2.2.6 STR8182X/CNS2182X (LFBGA-269)

Table 21. List of Pin Name and Pin Number for STR8182X/CNS2182X Package

No	Pin Name	No	Pin Name	No	Pin Name	No	Pin Name
A1	VCCA_U20	B1	GNDA_U20	C1	EXT_INT28	D1	I2SSD/GPIOA[15]
A2	DDM	B2	DDP	C2	SPIDR/GPIOA[26]	D2	SCL/GPIOA[14]
A3	VCCHSRT	B3	GNDHSRT	C3	GPIOA[3]/I2SDR/ UR_ACT1	D3	SDA/GPIOA[13]
A4	RREF	B4	GPIOA[1]/EXT_INT30	C4	SPICSN[1]/GPIOA[29]	D4	CLKOUT/ICESEL
A5	INTN[0]/EXT_INT5	B5	GNTN[1]/GPIOB[31]	C5	INTN[1]/EXT_INT6	D5	TESTMODE_EN
A6	GNTN[0]/GPIOB[29]	B6	AD[0]	C6	AD[1]	D6	AD[2]
A7	AD[5]	B7	AD[6]	C7	REQN[1]/GPIOB[30]	D7	GPIOA[0]/EXT_INT29
A8	CBEN[0]	B8	AD[8]	C8	GPIOA[2]/UR_ACT0	D8	REQN[0]/GPIOB[28]
A9	AD[9]	B9	AD[11]	C9	AD[3]	D9	AD[7]
A10	AD[15]	B10	AD[12]	C10	AD[4]	D10	AD[10]
A11	PAR	B11	CBEN[1]	C11	AD[13]	D11	AD[14]
A12	DEVSELN/GPIOB[24]	B12	PERRN/ GPIOB[26]	C12	STOPN/GPIOB[25]	D12	SERRN/GPIOB[27]
A13	FRAMEN	B13	IRDYN	C13	TRDYN/GPIOB[23]	D13	UR_TXD[1]/GPIOB[21]
A14	CBEN[2]	B14	AD[16]	C14	UR_RXD[1]/ GPIOB[22]	D14	WDTRSTN/GPIOA[25]
A15	AD[17]	B15	AD[18]	C15	ICK/TCK	D15	IMS/TMS
A16	AD[19]	B16	UR_RXD[0]	C16	EXTGOICE/TDI	D16	AD[21]
A17	GND	B17	UR_TXD[0]	C17	AD[20]	D17	AD[22]
No	Pin Name	No	Pin Name	No	Pin Name	No	Pin Name
E1	PCMDR/GPIOA[18]	F1	PCMDT/ GPIOA[19]	G1	P0_DP	H1	P1_DN
E2	I2SCLK/GPIOA[17]	F2	PCMFS/GPIOA[20]	G2	P0_DN	H2	P1_DP
E3	I2SWS/GPIOA[16]	F3	SPICSN[2]/ GPIOA[30]	G3	SPICSN[0]/ GPIOA[28]	H3	PCMCLK/GPIOA[21]
E4	SPICLK/GPIOA[27]	F4	SPICSN[3]/ GPIOA[31]	G4	SPIDT/ENDIAN	H4	REXT
E5	AVDD_U33	F5	AVDD_UP	G5	AGND_UP	H5	AGND_U
E6	PVDD	F6	AVDD_U	G6		H6	
E7	PVDD	F7		G7	GND	H7	GND
E8	GPIOA[4]	F8		G8	GND	H8	GND
E9	GPIOA[5]	F9		G9	GND	H9	GND
E10	GPIOA[6]	F10		G10	GND	H10	GND
E11	VFS	F11		G11	GND	H11	GND

E12	CVDD	F12	CVDD	G12		H12	
E13	CVDD	F13	PVDD	G13	PVDD	H13	PVDD
E14	RESETN	F14	IDIO/TDO	G14	AD[30]	H14	PCIRSTN
E15	TRSTN	F15	AD[24]	G15	AD[29]	H15	WAITN[2]/GPIOB[15]
E16	AD[23]	F16	AD[25]	G16	AD[28]	H16	PCICLK
E17	CBEN[3]	F17	AD[26]	G17	AD[27]	H17	AD[31]
No	Pin Name	No	Pin Name	No	Pin Name	No	Pin Name
J1	TXD[0]/RX-	K1	TXCLK/RX+	L1	RXCLK/AVDD_E	M1	REF_32768
J2	TXEN/LED1/ GPIOA[23]	K2	RXDV/ ANA_TESTA	L2	TXD[3]/ANA_TESTB	M2	V25_CTL
J3	TXD[1]/GPIOA[7]	K3	RXD[0]/TX-	L3	RXD[2]/REF_RES	M3	V125_CTL
J4	MDIO/GPIOB[1]/ LED2/GPIOA[24]	K4	RXD[1]/TX+	L4	MDC/GPIOB[0]/ LED0/GPIOA[22]	M4	IDECS0N/GPIOB[7]
J5	CVDD	K5	PVDD_E/U2-NC-44	L5	COL/GPIOB[2]/ AGND_E	M5	TXD[2]/AVDD_E
J6		K6		L6		M6	NC
J7	GND	K7	GND	L7	GND	M7	
J8	GND	K8	GND	L8	GND	M8	
J9	GND	K9	GND	L9	GND	M9	
J10	GND	K10	GND	L10	GND	M10	
J11	GND	K11	GND	L11	GND	M11	
J12		K12		L12		M12	CVDD
J13	PVDD_D	K13	PVDD_D	L13	VREF	M13	CVDD
J14	WAITN[1]/ GPIOB[14]	K14	**DA[6]**	L14	SOEN	M14	DA[11]
J15	SCEN[1]/GPIOB[11]	K15	DA[4]	L15	DA[8]	M15	DA[10]
J16	SCEN[0]	K16	SWEN	L16	DA[3]	M16	DA[7]
J17	SCEN[2]/GPIOB[12]	K17	DA[0]	L17	DA[1]	M17	DA[2]
No	Pin Name	No	Pin Name	No	Pin Name	No	Pin Name
N1	SXO	P1	SXI	R1	DMARQ/GPIOB[4]	T1	SDQ[9]/SA[17]
N2	V18_CTL	P2	IORDY/ GPIOB[3]	R2	IDECS1N/GPIOB[8]	T2	SDQ[12]/SA[20]
N3	DIORN/GPIOB[9]	P3	DMACKN/ GPIOB[5]	R3	SDQ[6]	T3	SDQ[7]
N4	DIOWN/GPIOB[10]	P4	INTRQ/ GPIOB[6]	R4	SDQ[0]	T4	SDQ[8]/SA[16]
N5	RXD[3]/AGND_E	P5	SDQ[1]	R5	SDQ[2]	T5	SDQ[11]/SA[19]
N6	CVDD	P6	SDQ[4]	R6	SDQ[5]	T6	SDQ[14]/SA[22]
N7	AGND_R	P7	SDQ[3]	R7	SDQ[13]/SA[21]	T7	SA[7]
N8	AVDD_SP	P8	SDQ[10]/ SA[18]	R8	SA[5]	T8	SA[9]
N9	AGND_SP	P9	SA[4]	R9	SA[8]	T9	SA[13]
N10	AVDD_R33	P10	SA[12]	R10	SA[11]	T10	DQS[1]
N11	PVDD	P11	CKE/SPIBOOT	R11	SA[15]	T11	DQS[0]
N12	PVDD	P12	DM[1]	R12	DCSN	T12	DDQ[14]
N13	PVDD	P13	RASN	R13	DWEN	T13	DDQ[12]
N14	CASN	P14	BA[1]	R14	DM[0]	T14	DDQ[10]
N15	DA[12]	P15	DA[9]	R15	BA[0]	T15	DDQ[8]
N16	DDQ[0]	P16	DDQ[2]	R16	DDQ[4]	T16	DDQ[6]
N17	DA[5]	P17	DDQ[1]	R17	DDQ[3]	T17	DDQ[5]
No	Pin Name						
U1	GND						
U2	SDQ[15]/SA[23]						
U3	SA[0]						
U4	SA[1]						
U5	SA[2]						

U6	SA[3]						
U7	SA[6]						
U8	SA[10]						
U9	SA[14]						
U10	CKN						
U11	CK						
U12	DDQ[15]						
U13	DDQ[13]						
U14	DDQ[11]						
U15	DDQ[9]						
U16	DDQ[7]						
U17	GND						

Note: The pin name in red color shows the differen between STR8182/CNS2182 and STR8182X/CNS2182X.

2.2.7 CNS2135 (QFP-128)

Table 22. List of Pin Name and Pin Number for CNS2135 Package

No	Pin Name	No	Pin Name	No	Pin Name	No	Pin Name
1	AVDD_R33	33	NC	65	NC	97	PVDD
2	V25_CTL	34	PVDD_D	66	NC	98	PGND
3	V18_CTL	35	PGND	67	PGND	99	TESTMODE_EN
4	AGND_R/SP	36	NC	68	CGND	100	I2SSD/GPIOA[15]
5	AVDD_SP	37	NC	69	CGND	101	I2SWS/GPIOA[16]
6	PVDD	38	NC	70	TRSTn	102	CGND
7	PGND	39	NC	71	IDIO/TDO	103	I2SCLK/GPIOA[17]
8	SXO	40	NC	72	IMS/TMS	104	PCMDR/GPIOA[18]
9	SXI	41	NC	73	PGND	105	PCMDT/GPIOA[19]
10	LED0/GPIOA[22]	42	NC	74	PVDD	106	PCMFS/GPIOA[20]
11	LED1/GPIOA[23]	43	NC	75	ICK/TCK	107	PCMCLK/GPIOA[21]
12	LED2/GPIOA[24]	44	PVDD_D	76	EXTGOICE/TDI	108	P0_DP
13	IORDY/GPIOB[3]	45	NC	77	RESETn	109	P0_DN
14	DMARQ/GPIOB[4]	46	NC	78	UR_TXD[0]	110	AVDD_U33
15	PVDD	47	NC	79	UR_RXD[0]	111	AGND_UP
16	PGND	48	NC	80	CVDD	112	AVDD_U/UP
17	CGND	49	CGND	81	CGND	113	REXT
18	CVDD	50	CVDD	82	PGND	114	AGND_U
19	NC	51	VDDQ/VDD (SDRAM,3.3V)	83	PVDD	115	P1_DN
20	CKE (SDRAM)	52	NC	84	GPIOA[0]/EXT_INT29	116	P1_DP
21	CKE/SPIBoot	53	NC	85	GPIOA[1]/EXT_INT30	117	CVDD
22	CK	54	NC	86	GPIOA[2]/UR_ACT0	118	AGND_E
23	CLK (SDRAM)	55	NC	87	GPIOA[3]/UR_ACT1	119	REF_RES
24	NC	56	NC	88	SPIDR/GPIOA[26]	120	AVDD_E
25	NC	57	VSSQ/VSS (SDRAM)	89	SPIDT/Endian	121	TX+
26	NC	58	NC	90	SPICLK/GPIOA[27]	122	TX-
27	NC	59	NC	91	SPICSn[0]/GPIOA[28]	123	AGND_E
28	NC	60	NC	92	SPICSn[1]/GPIOA[29]	124	RX+
29	NC	61	NC	93	CLKOUT/ICESEL	125	RX-
30	NC	62	NC	94	CGND	126	AVDD_E
31	NC	63	NC	95	CVDD	127	REF_32768
32	NC	64	NC	96	VFS	128	CVDD

3 System Address Map and Register Descriptions

3.1 Memory Mapping

The following Table outlines the 32-bit address memory map of all function blocks, external memory, and external devices.

Table 23. Memory Map

Function	Base Address	Size of Region	Descriptions
Alias Memory	0x0000_0000	256MB	**Alias Memory Region.** Memory map has two states, one is Reset State; the other is Re-map State. After power on reset, Memory map is at the Reset State, and the Flash/SRAM Memory Region or SPI Serial Flash Memory Region is mapped to the region, and system can be cold boot from external Flash Memory. The selection of parallel/serial flash booting is by the external pull down/up resister on CKE pin. After the "Remap_Enable" bit (bit 0 of Memory Re-map Register of MISC Register Region) is set to 1, Memory map is changed to Re-map State, and the DDR/SDR SDRAM Memory Region is mapped to the region.
Flash/SRAM Memory Bank 0	0x1000_0000	16MB	**Flash/SRAM Memory Bank 0 Region.** External Flash memory (Bank 0) can be accessed through this region.
Flash/SRAM Memory Bank 1	0x1100_0000	16MB	**Flash/SRAM Memory Bank 1 Region.** External Flash or SRAM memory (Bank 1) can be accessed through this region.
Flash/SRAM Memory Bank 2	0x1200_0000	16MB	**Flash/SRAM Memory Bank 2 Region.** External Flash or SRAM memory (Bank 2) can be accessed through this region.
Flash/SRAM Memory Bank 3	0x1300_0000	16MB	**Flash/SRAM Memory Bank 3 Region.** External Flash or SRAM memory (Bank 3) can be accessed through this region.
Reserved	0x1400_0000	16MB	
Reserved	0x1500_0000	16MB	
Reserved	0x1600_0000	16MB	
Reserved	0x1700_0000	16MB	
IDE Device Register Space	0x1800_0000	16MB	**IDE Device Register Region** IDE device registers can be accessed through this region.
Reserved	0x1900_0000	112MB	
SDR/DDR SDRAM Memory	0x2000_0000	256MB	**SDR/DDR SDRAM Memory Region** External SDR/DDR SDRAM memory can be accessed through this region.
SPI Serial Flash Memory	0x3000_0000	256MB	**SPI Serial Flash Memory Region** External SPI Serial Flash Memory (SPI Bank 0) can be accessed through this region.
Reserved	0x4000_0000	256MB	

Reserved	0x5000_0000	256MB	
Generic DMA Register	0x6000_0000	256MB	**Generic DMA (GDMA) Register Region** The registers of GDMA can be accessed through this region.
NIC Register	0x7000_0000	16MB	**GbE Controller(NIC) Register Region** The registers of GbE Controller can be accessed through this region.
SPI/PCM/TWI/IS Register	0x7100_0000	16MB	**SPI/PCM/TWI/I2S Register Region** The registers of SPI/PCM/TWI/I2S can be accessed through this region.
SDR/DDR SDRAM Control Register	0x7200_0000	16MB	**SDR/DDR SDRAM Control Register Region.** The registers of SDR/DDR SDRAM Controller can be accessed through this region.
Static Memory Control Register	0x7300_0000	16MB	**Static Memory Control Register Region** The registers of Static Memory Controller (SMC) can be accessed through this region.
IDE Control Register	0x7400_0000	16MB	**IDE Control Register Region** The registers of IDE Controller can be accessed through this region.
Reserved	0x7500_0000	16MB	
MISC Register	0x7600_0000	16MB	**MISC Register Region** Memory re-map control register, and PCI Bridge capability registers, and AHB bus control register are collected at this region.
Power Management Register	0x7700_0000	16MB	**Power Management Register Region** The registers of Clock and Power Management can be accessed through this region.
UART0 Register	0x7800_0000	8MB	**UART 0 Register Region** The registers of UART 0 can be accessed through this region.
UART1 Register	0x7880_0000	8MB	**UART 1 Register Region** The registers of UART 1 can be accessed through this region.
Timer Register	0x7900_0000	16MB	**Timer Register Region** The registers of Timer can be accessed through this region.
Watch Dog Timer Register	0x7A00_0000	16MB	**Watch Dog Timer (WDT) Register Region** The registers of WDT can be accessed through this region.
Real Time Counter Register	0x7B00_0000	16MB	**Real Time Counter (RTC) Register Region** The registers of RTC can be accessed through this region.
GPIOA Register	0x7C00_0000	8MB	**GPIOA Register Region** The registers of GPIOA can be accessed through this region.
GPIOB Register	0x7C80_0000	8MB	**GPIOB Register Region** The registers of GPIOB can be accessed through this region.
Reserved	0x7D00_0000	48MB	Reserved
Reserved	0x8000_0000	256MB	Reserved
Reserved	0x9000_0000	256MB	Reserved
PCI Configuration Data Register	0xA000_0000	64MB	**PCI Configuration Data Register Region** The embedded PCI Bridge provides a configuration window (CONFIG_DATA and CONFIG_ADDR registers)

			for Host CPU to configure all of attached PCI devices and the PCI Bridge itself. CONFIG_DATA is the one and only one register at this region.
PCI Configuration Address Register	0xA400_0000	64MB	**PCI Configuration Address Register Region** The embedded PCI Bridge provides a configuration window (CONFIG_DATA and CONFIG_ADDR registers) for Host CPU to configure all of attached PCI devices and the PCI Bridge itself. CONFIG_ADDR is the one and only one register at this region.
PCI I/O Space	0xA800_0000	128MB	**PCI I/O Space Region** All of AHB bus transaction at this region will be translated to PCI bus I/O space transaction.
PCI Memory Space	0xB000_0000	256MB	**PCI Memory Space Region** All of AHB bus transaction at this region will be translated to PCI bus Memory space transaction.
USB1.1 Configuration Register	0xC000_0000	64MB	**USB1.1 Configuration Register Region** USB1.1 Configuration registers can be accessed through this region.
USB1.1 Operation Register	0xC400_0000	64MB	**USB1.1 Operation Register Region** USB1.1 Operation registers can be accessed through this region.
USB2.0 Configuration Register	0xC800_0000	64MB	**USB2.0 Configuration Register Region** USB2.0 Configuration registers can be accessed through this region.
USB2.0 Operation Register	0xCC00_0000	64MB	**USB2.0 Operation Register Region** USB2.0 Operation registers can be accessed through this region.
USB 1.1/2.0 Device Register	0xD000_0000	256MB	**USB 1.1/2.0 Device Register Region** USB 1.1/2.0 Device registers can be accessed through this region
Reserved	0xE000_0000	256MB	Reserved
Reserved	0xF000_0000	256MB -4KB	Reserved
Vector Interrupt Control Register	0xFFFF_F000	4KB	**Vector Interrupt Control Register Region** The registers of Interrupt Controller can be accessed through this region.

3.2 Register Overview

Table 24. Generic DMA Registers

Address	Register Name	Short Name	Details
0x00	Interrupt Status	INTR_STAT	
0x04	Terminal Count Interrupt Status	INTR_TC	
0x08	Terminal Count Interrupt Status Clear	INTR_TC_CLR	
0x0C	Error Interrupt Status	INTR_ERR	
0x10	Error Interrupt Status Clear	INTR_ERR_CLR	
0x14	Terminal Count Status	TC_STAT	
0x18	Error Status	ERR_STAT	
0x1C	Channel Enable Status	CH_EN_FLAG	
0x20	Channel Busy Status	CH_BUSY_FLAG	
0x24	Main Configuration Status	CSR	
0x28	Synchronous	SYNC	
0x100	Channel 0 Control	C0_CSR	
0x120	Channel 1 Control	C1_CSR	
0x140	Channel 2 Control	C2_CSR	
0x160	Channel 3 Control	C3_CSR	
0x180	Channel 4 Control	C4_CSR	
0x1A0	Channel 5 Control	C5_CSR	
0x1C0	Channel 6 Control	C6_CSR	
0x1E0	Channel 7 Control	C7_CSR	
0x104	Channel 0 Configuration	C0_CFG	
0x124	Channel 1 Configuration	C1_CFG	
0x144	Channel 2 Configuration	C2_CFG	
0x164	Channel 3 Configuration	C3_CFG	
0x184	Channel 4 Configuration	C4_CFG	
0x1A4	Channel 5 Configuration	C5_CFG	
0x1C4	Channel 6 Configuration	C6_CFG	
0x1E4	Channel 7 Configuration	C7_CFG	
0x108	Channel 0 Source Address	C0_SrcAddr	
0x128	Channel 1 Source Address	C1_SrcAddr	
0x148	Channel 2 Source Address	C2_SrcAddr	
0x168	Channel 3 Source Address	C3_SrcAddr	
0x188	Channel 4 Source Address	C4_SrcAddr	
0x1A8	Channel 5 Source Address	C5_SrcAddr	

0x1C8	Channel 6 Source Address	C6_SrcAddr	
0x1E8	Channel 7 Source Address	C7_SrcAddr	
0x10C	Channel 0 Destination Address	C0_DstAddr	
0x12C	Channel 1 Destination Address	C1_DstAddr	
0x14C	Channel 2 Destination Address	C2_DstAddr	
0x16C	Channel 3 Destination Address	C3_DstAddr	
0x18C	Channel 4 Destination Address	C4_DstAddr	
0x1AC	Channel 5 Destination Address	C5_DstAddr	
0x1CC	Channel 6 Destination Address	C6_DstAddr	
0x1EC	Channel 7 Destination Address	C7_DstAddr	
0x110	Channel 0 Linked List Descriptor Pointer	C0_LLP	
0x130	Channel 1 Linked List Descriptor Pointer	C1_LLP	
0x150	Channel 2 Linked List Descriptor Pointer	C2_LLP	
0x170	Channel 3 Linked List Descriptor Pointer	C3_LLP	
0x190	Channel 4 Linked List Descriptor Pointer	C4_LLP	
0x1B0	Channel 5 Linked List Descriptor Pointer	C5_LLP	
0x1D0	Channel 6 Linked List Descriptor Pointer	C6_LLP	
0x1F0	Channel 7 Linked List Descriptor Pointer	C7_LLP	
0x114	Channel 0 Transfer Size	C0_SIZE	
0x134	Channel 1 Transfer Size	C1_SIZE	
0x154	Channel 2 Transfer Size	C2 SIZE	
0x174	Channel 3 Transfer Size	C3 SIZE	
0x194	Channel 4 Transfer Size	C4_SIZE	
0x1B4	Channel 5 Transfer Size	C5_SIZE	
0x1D4	Channel 6 Transfer Size	C6_SIZE	
0x1F4	Channel 7 Transfer Size	C7_SIZE	

Table 25. Giga NIC Registers

Address	Register Name	Short Name	Details
0x000	PHY Control Register 0	PHY_CTRL0	
0x004	PHY Control Register 1	PHY_CTRL1	
0x008	MAC Configuration	MAC_CFG	
0x00C	Flow Control Configuration	FC_CFG	
0x010	ARL Configuration	ARL_CFG	
0x014	My MAC High Byte	My_MAC_H	
0x018	My MAC Low Byte	My_MAC_L	
0x01C	Hash Table Control	HASH_CTRL	

0x020	My VLAN ID Control	VLAN_CTRL	
0x024	My VLAN ID 0 – 1	VLAN_ID_0_1	
0x028	My VLAN ID 2 – 3	VLAN_ID_2_3	
0x030	DMA Configuration	DMA_CFG	
0x034	TX_DMA Control	TX_DMA_CTRL	
0x038	RX_DMA Control	RX_DMA_CRTL	
0x03C	TX Descriptor Pointer	TX_DPTR	
0x040	RX Descriptor Pointer	RX_DPTR	
0x044	TX Descriptor Base Address	TX_BASE_ADDR	
0x048	RX Descriptor Base Address	RX_BASE_ADDR	
0x04C	Delayed Interrupt Configuration	DLY_INT_CFG	
0x050	Interrupt Status	INT	
0x054	Interrupt Mask	INT_MASK	
0x058	Test 0 (Clock Skew Setting)	TEST0	
0x05C	Test 1 (Queue Status)	TEST1	
0x060	Extended Configuration Register	Extend_CFG	
0x100	RX OK Packet Counter	C_RXOKPKT	
0x104	RX OK Byte Counter	C_RXOKBYTE	
0x108	RX Runt Packet Counter	C_RXRUNT	
0x10C	RX Over Size Packet Counter	C_RXLONG	
0x110	RX No Buffer Drop Packet Counter	C_RXDROP	
0x114	RX CRC Error Packet Counter	C_RXCRC	
0x118	RX ARL Drop Packet Counter	C_RXARLDROP	
0x11C	My VLAN ID Mismatch Drop Counter	C_RXVLANDROP	
0x120	RX Check Sum Error Packet Counter	C_RXCSERR	
0x124	RX Pause Frame Packet Counter	C_RXPAUSE	
0x128	TX OK Packet Counter	C_TXOKPKT	
0x12C	TX OK Byte Counter	C_TXOKBYTE	
0x130	TX Collision Counter/Pause Frame Counter	C_TXPAUSECOL	

Table 26. SPI/PCM/TWI/I2S Registers

Address	Register Name	Short Name	Details
0x20	TWI Control	TWI_CTRL	
0x24	TWI Time-Out	TWI_TIMEOUT	
0x28	TWI Slave Address	TWI_SLAVE_ADDR	
0x2C	TWI Write Data	TWI_WR_DATA	
0x30	TWI Read Data	TWI_RD_DATA	

0x34	TWI Interrupt Status	TWI_INTR_STAT	
0x38	TWI Interrupt Enable	TWI_INTR_ENA	
0x40	SPI Configuration	SPI_CFG	
0x44	SPI Service Status	SPI_STAT	
0x48	SPI Bit Rate	SPI_BIT_RATE	
0x4C	SPI Transmit Control	SPI_TX_CTRL	
0x50	SPI Transmit Data	SPI_TX_DATA	
0x54	SPI Receive Control	SPI_RX_CTRL	
0x58	SPI Receive Data	SPI_RX_DATA	
0x5C	SPI FIFO Transmit Configuration	SPI_FIFO_TX_CFG	
0x60	SPI FIFO Transmit Control	SPI_FIFO_TX_CTRL	
0x64	SPI FIFO Receive Configuration	SPI_FIFO_RX_CFG	
0x68	SPI Interrupt Status	SPI_INTR_STAT	
0x6C	SPI Interrupt Enable	SPI_INTR_ENA	
0x80	PCM Configuration 0	PCM_CFG_0	
0x84	PCM Configuration 1	PCM_CFG_1	
0x88	PCM Channel 0 Configuration	PCM_CH0_CFG	
0x8C	PCM Channel 1 Configuration	PCM_CH1_CFG	
0x90	PCM Channel 2 Configuration	PCM_CH2_CFG	
0x94	PCM Channel 3 Configuration	PCM_CH3_CFG	
0x98	PCM Transmit Data[31:0]	PCM_TX_DATA_L	
0x9C	PCM Transmit Data[63:32]	PCM_TX_DATA_H	
0xA0	PCM Receive Data[31:0]	PCM_RX_DATA_L	
0xA4	PCM Receive Data[63:32]	PCM_RX_DATA_H	
0xA8	PCM Interrupt Status	PCM_INTR_STAT	
0xAC	PCM Interrupt Enable	PCM_INTR_ENA	
0xC0	I2S Configuration	I2S_CFG	
0xC4	I2S Right Transmit Data	I2S_RIGHT_TX_DATA	
0xC8	I2S Left Transmit Data	I2S_LEFT_TX_DATA	
0xCC	I2S Right Receive Data	I2S_RIGHT_RX_DATA	
0xD0	I2S Left Receive Data	I2S_LEFT_RX_DATA	
0xD4	I2S Interrupt Status	I2S_INTR_STAT	
0xD8	I2S Interrupt Enable	I2S_INTR_ENA	

Table 27. DDR/SDR SDRAM Controller Registers

Address	Register Name	Short Name	Details
0x00	Memory Interface Configure Register	MEM_CFG	

0x04	DRAM Parameter Configuration	DRAM_CFG	
0x08	Power ON Initial Control Register	POWER_ON_INIT_CTRL	
0x10	DRAM Timing Parameter Register 0	DRAM_TIMING0	
0x14	DRAM Timing Parameter Register 1	DRAM_TIMING1	
0x18	DRAM Timing Parameter Register 2	DRAM_TIMING2	
0x1C	Pre-Read Time Out Disable Register	PREREAD_TIMEOUT_DIS	
0x20	Pre-Read Enable	PREREAD_ENA	
0x24	Pre-Read Time Out Register 0	PREREAD_TIMEOUT0	
0x28	Pre-Read Time Out Register 1	PREREAD_TIMEOUT1	
0x30	DDQ Output Delay Control Register (DDR Only)	DDQ_OUT_DLY_CTRL	
0x34	DQS Input Delay Control Register (DDR Only)	DDQ_IN_DLY_CTRL	
0x3C	Pad Power Down Register	PAD_PWR_DOWN	

Table 28. Static Memory Controller Registers

Address	Register Name	Short Name	Details
0x00	Memory Bank 0 Configuration Register	MEM_BNK0_CFG	
0x04	Memory Bank 0 Timing Parameter Register	MEM_BNK0_TIMING	
0x08	Memory Bank 1 Configuration Register	MEM_BNK1_CFG	
0x0C	Memory Bank 1 Timing Parameter Register	MEM_BNK1_TIMING	
0x10	Memory Bank 2 Configuration Register	MEM_BNK2_CFG	
0x14	Memory Bank 2 Timing Parameter Register	MEM_BNK2_TIMING	
0x18	Memory Bank 3 Configuration Register	MEM_BNK3_CFG	
0x1C	Memory Bank 3 Timing Parameter Register	MEM_BNK3_TIMING	

Table 29. IDE Controller Registers

Base Address: 0x7400_0000

Address	Register Name	Short Name	Details
0x00	IDE PIO mode Control Register		
0x04	IDE Drive0 PIO Timing Configuration Register		
0x08	IDE Drive1 PIO Timing Configuration Register		
0x0C	IDE Drive0 DMA Timing Configuration Register		
0x10	IDE Drive1 DMA Timing Configuration Register		
0x14	IDE Ultra DMA mode Timing Configuration Register		
0x18	IDE DMA and Ultra DMA mode Control Register		
0x1C	IDE Status and Control Register		

0x20	IDE DMA Descriptor Table Pointer Register		
0x24	IDE to USB Fast Path Access Window Register		
0x28	IDE to USB Fast Path DMA Burst Size Register		

Table 30. IDE Device Registers

Base Address: 0x1800_0000

0x20	Data Register		
0x24	Error Register (Read)		
0x24	Feature Register (Write)		
0x28	Sector Count Register		
0x2C	LBA Low Register		
0x30	LBA MID Register		
0x34	LBA High Register		
0x38	Device Register		
0x3C	Command Register (Write)		
0x3E	Status Register (Read)		
0x40	Device Control Register (Write)		
0x40	Alternate Status Register (Read)		

Table 31. Miscellaneous Registers

Address	Register Name	Short Name	Details
0x00	Memory Re-map Register		
0x04	Chip Configuration Register		
0x10	PCI Control and Broken Mask Register		
0x14	PCI Broken Status Register		
0x18	PCI Device ID and Vendor ID Register		
0x1C	USB Host PHY Control and Test Register		
0x20	GPIO_A Pin Enable Register	GPIOA_EN	
0x24	GPIO_B Pin Enable Register	GPIOB_EN	
0x28	GPIOA Pull Up/Down Configuration Register		
0x2C	GPIOA Drive Strength Configuration Register		
0x30	Fast Ethernet PHY LED Configuration Register		
0x40	HSDMA Control and Status Register		
0x50	HSDMA Master 0 Address Register	M0Addr	
0x54	HSDMA Master 1 Address Register	M1Addr	
0x58	HSDMA Linked List Descriptor Pointer	HSDMA_LLP	

0x5C	HSDMA Transfer Size Register	HSDMA_TOT_SIZE	

Table 32. Clock and Power Management Registers

Address	Register Name	Short Name	Details
0x00	Clock gate control register 0 for AHB and APB devices		
0x04	Clock gate control register 1 for AHB and APB devices		
0x08	Software reset control		
0xC0	System clock control register		
0x10	PLL/Hard Macro Power Down Control Register		
0x14	CPU Initialization Register		
0x1C	Pad Drive Strength Control Register		
0x20	USB Device Power Management Register		
0x24	Regulator Control Register		
0x28	Reserved		
0x2C	PLLx2250 Control Register		

Table 33. UART0 and UART1 Registers

Address	Register Name	Short Name	Details
0x00	Receive Buffer Register/Transmit Holding Register/Baud-Rate Divisor Latch	RBR / THR / DLL	
0x04	Interrupt Enable Register / Baud-Rate Divisor Latch	IER / DLM	
0x08	Interrupt Identification Register / Pre-scalar Register	IIR / FCR / PSR	
0x0C	Line Control Register (LCR)	LCR	
0x10	UART Control Register (UCR)	UCR	
0x14	UART Line Status /Test Control Register	LSR	
0x1C	Scratch Pad Register (SPR)	SPR	

Table 34. Timer Registers

Address	Register Name	Short Name	Details
0x00	Timer1 Counter Register		
0x04	Timer1 Auto Reload Value Register		
0x08	Timer1 Match Value 1 Register		
0x0C	Timer1 Match Value 2 Register		

0x10	Timer2 Counter Register		
0x14	Timer2 Auto Reload Value Register		
0x18	Timer2 Match Value 1 Register		
0x1C	Timer2 Match Value 2 Register		
0x30	Timer 1 and 2 Control Register		
0x34	Interrupt Status Register		
0x38	Interrupt Mask Register		
0x40	Free Running Timer		
0x44	Free Running Timer Control Register		

Table 35. Watch Dog Timer Registers

Address	Register Name	Short Name	Details
0x00	Watch Dog Timer Counter Register		
0x04	Watch Dog Timer Counter Auto-reload Register		
0x08	Watch Dog Timer Counter Restart Register		
0x0C	Watch Dog Timer Control Register		
0x10	Watch Dog Timer Status Register		
0x14	Watch Dog Timer Clear Register		
0x18	Watch Dog Timer Interrupt Length Register		

Table 36. Real Time Counter Registers

Address	Register Name	Short Name	Details
0x00	RTC Second Register		
0x04	RTC Minute Register		
0x08	RTC Hour Register		
0x0C	RTC Day Register		
0x10	RTC Second Alarm Register		
0x14	RTC Minute Alarm Register		
0x18	RTC Hour Alarm Register		
0x1C	RTC Record Register		
0x20	RTC control Register		
0x34	Interrupt Status Register		

Table 37. GPIOA and GPIOB Controller Registers

Address	Register Name	Short Name	Details

0x00	GPIO Data Output Register	GpioDataOut	
0x04	GPIO Data Input Register	GpioDataIn	
0x08	GPIO Direction Register	PinDir	
0x10	GPIO Data Bit Set Register	GpioDataSet	
0x14	GPIO Data Bit Clear Register	GpioDataClear	
0x20	GPIO Interrupt Enable Register	IntrEnable	
0x24	GPIO Interrupt Raw Status Register	IntrRawState	
0x28	GPIO Interrupt Masked Status Register	IntrMaskedState	
0x2C	GPIO Interrupt Mask Register	IntrMask	
0x30	GPIO Interrupt Clear Register	IntrClear	
0x34	GPIO Interrupt Trigger Method Register	IntrTrigger	
0x38	GPIO Interrupt Trigger by Both Edges Register	IntrBoth	
0x3C	GPIO Interrupt Trigger by Rising-/Falling-Edge or High/Low level Register	IntrRiseNeg	
0x40	GPIO Bounce Enable Register	BounceEnable	
0x44	GPIO Bounce clock pre-scale Register	BouncePreScale	

Table 38. PCI Configuration Registers

Address	Register Name	Short Name	Details
0xA000_0000	PCI Configuration Data	CONFIG_DATA	
0xA400_0000	PCI Configuration Address	CONFIG_ADDR	

Table 39. USB Host 1.1 Configuration Registers

Address	Register Name	Short Name	Details
0x04-05	Command Register		
0x44	Operational Mode Enable Register		

Table 40. USB Host 1.1 Operation Registers

Address	Register Name	Short Name	Details
0x00	HcRevision Register	HcRevision	
0x04	HcControl Register	HcControl	
0x08	HcCommandStatus Register	HcCommandStatus	
0x0C	HcInterruptStatus Register	HcInterruptStatus	
0x10	HcInterruptEnable Register	HcInterruptEnable	

0x14	HcInterruptDisable Register	HcInterruptDisable	
0x18	HcHCCA Register	HcHCCA	
0x1C	HcPeriodCurrentED Register	HcPeriodCurrentED	
0x20	HcControlHeadED Register	HcControlHeadED	
0x24	HcControlCurrentED Register	HcControlCurrentED	
0x28	HcBulkHeadED Register	HcBulkHeadED	
0x2C	HcBulkCurrentED Register	HcBulkCurrentED	
0x30	HcDoneHead Register	HcDoneHead	
0x34	HcFmInterval Register	HcFmInterval	
0x38	HcFmRemaining Register	HcFmRemaining	
0x3C	HcFmNumber Register	HcFmNumber	
0x40	HcPeriodicStart Register	HcPeriodicStart	
0x44	HcLSThreshold Register	HcLSThreshold	
0x48	HcRhDescriptorA Register	HcRhDescriptorA	
0x4C	HcRhDescriptorB Register	HcRhDescriptorB	
0x50	HcRhStatus Register	HcRhStatus	
0x54	HcRhPortStatus [1] Register (USB Port 0)	HcRhPortStatus[1]	
0x58	HcRhPortStatus [2] Register (USB Port 1)	HcRhPortStatus[2]	

Table 41. USB Host 2.0 Configuration Registers

Address	Register Name	Short Name	Details
0x04-05	Command Register		
0x40-43	Operational Mode Enable Register		

Table 42. USB Host 2.0 Operation Registers

Address	Register Name	Short Name	Details
0x00	Capability Registers Length	CAPLENGTH	
0x02-03	Host Controller Interface Version Number	HCIVERSION	
0x04-07	Structure Parameters	HCSPARAMS	
0x08-0B	Capability Parameters	HCCPARAMS	
0x20-23	USB2.0 Command Register	USB2CMD	
0x24-27	USB2.0 Status Register	USB2STS	
0x28-2C	USB2.0 Interrupt Enable Register	USB2INTR	
0x2C-2F	Frame Index Register	FRINDEX	
0x34-37	Periodic Frame List Base Address Register	PERIODICLISTBASE	
0x38-3B	Current Asynchronous List Address Register	ASYNCLISTBASE	

0x60-63	Configure Flag Register	CONFIGFLAG	
0x64-67	Port 0 Status and Control Register	PORTSC0	
0x68-6B	Port 1 Status and Control Register	PORTSC1	

Table 43. USB 1.1/2.0 Device Controller Registers

Address	Register Name	Short Name	Details
0x00	Main Control Register		
0x01	Device Address Register		
0x02	Test Register		
0x04	SOF Frame Number Register Byte 0		
0x05	SOF Frame Number Register Byte 1		
0x06	SOF Mask Timer Register Byte 0		
0x07	SOF Mask Timer Register Byte 1		
0x08	PHY Test Mode Selector Register		
0x09	Vendor Specific IO Control Register		
0x0A	Vendor Specific IO Status Register		
0x0B	CX Configuration and Status Register		
0x0C	Endpoint 0 Data Port Register Byte 0		
0x10	Interrupt Group Mask Register		
0x11	Interrupt Mask Register Byte 0		
0x12	Interrupt Mask Register Byte 1		
0x13	Interrupt Mask Register Byte 2		
0x15	Interrupt Mask Register Byte 4		
0x16	Interrupt Mask Register Byte 5		
0x17	Interrupt Mask Register Byte 6		
0x18	Interrupt Mask Register Byte 7		
0x19	Receive Zero-length Data Packet Register Byte 0		
0x1A	Receive Zero-length Data Packet Register Byte 1		
0x1C	FIFO Empty Byte 0		
0x1D	FIFO Empty Byte 1		
0x1E	Initial Value of Random Pattern		
0x1F	Byte Count of Random Pattern		
0x20	Interrupt Group Register		
0x21	Interrupt Source Register Byte 0		
0x22	Interrupt Source Register Byte 1		
0x23	Interrupt Source Register Byte 2		
0x25	Interrupt Source Register Byte 4		

0x26	Interrupt Source Register Byte 5		
0x27	Interrupt Source Register Byte 6		
0x28	Interrupt Source Register Byte 7		
0x29	Isochronous Sequential Error Register Byte 0		
0x2A	Isochronous Sequential Error Register Byte 1		
0x2B	Isochronous Sequential Abort Register Byte 0		
0x2C	Isochronous Sequential abort Register Byte 1		
0x2D	Transferred Zero-length Register Byte 0		
0x2E	Transferred Zero-length Register Byte 1		
0x2F	Idle Counter		
0x30-37	Endpoint x Map Register, x = 1~8.		
0x3F	HBF Data Byte Count		
0x(40+2(x-1)); x =1~8	IN Endpoint x MaxPacketSize Register Low Byte		
0x(41+2(x-1)); x =1~8	IN Endpoint x MaxPacketSize Register High Byte		
0x(60+2(x-1)); x =1~8	OUT Endpoint x MaxPacketSize Register Low Byte		
0x(61+2(x-1)); x =1~8	OUT Endpoint x MaxPacketSize Register High Byte		
0x7E	DMA Mode Enable Register Low Byte		
0x7F	DMA Mode Enable Register High Byte		
0x80-8F	FIFOx Map Register, x = 1~15		
0x90-9F	FIFOx Configuration Register, x = 1~15		
0xA0-AF	FIFOx Instruction Register, x = 1~15		
0xB0-BF	FIFOx Byte-Count Register Low Byte, x = 1~15		
0xC0-FC	Data Port Register		

Table 44. Vector Interrupt Controller Registers

Address	Register Name	Short Name	Details
0x00	Interrupt Raw Status Register		
0x04	Edge Interrupt Source Clear Register		
0x08	Interrupt Mask Register		
0x0C	Interrupt Mask Clear Register		

0x10	Interrupt Trigger Mode Register		
0x14	Interrupt Trigger Level Register		
0x18	FIQ Select Register		
0x1C	IRQ Status Register		
0x20	FIQ Status Register		
0x24	Software Interrupt Register		
0x28	Software Interrupt Clear Register		
0x2C	Software Priority Mask Register		
0x34	Power Management Interrupt Register		
0x40-BF	Vector Address 0 ~ 31 Register		
0x0C0 - 0x13F	Interrupt 0 ~ 31 Priority Register		
0x140	IRQ Vector Address Register		
0x144	VIC Control Register		

Table 45. Embedded FE PHY Management Registers

Address	Register Name	Short Name	Details
0x00	MII Control Register		
0x01	MII Status Register		
0x02	PHY Identifier Register-High		
0x03	PHY Identifier Register-Low		
0x04	Auto-Negotiation Advertisement Register		
0x05	Auto-Negotiation Link Partner Base Page Ability Register		
0x06	Auto-Negotiation Expansion Register		
0x1F	Page Selection Register		

3.3 Generic DMA

3.3.1 Interrupt Status

Short Name: INTR_STAT
Address: 0x00

Table 46. Interrupt Status

Bits	Type	Name	Description	Default
7	RO	INT[7]	**Status of GDMA Interrupts after Masking (Channel-7).** 0: no pending interrupt at channel-7. 1: with pending interrupt at channel-7.	0x0
6	RO	INT[6]	**Status of GDMA Interrupts after Masking (Channel-6).** 0: no pending interrupt at channel-6. 1: with pending interrupt at channel-6.	0x0
5	RO	INT[5]	**Status of GDMA Interrupts after Masking (Channel-5).** 0: no pending interrupt at channel-5. 1: with pending interrupt at channel-5.	0x0
4	RO	INT[4]	**Status of GDMA Interrupts after Masking (Channel-4).** 0: no pending interrupt at channel-4. 1: with pending interrupt at channel-4.	0x0
3	RO	INT[3]	**Status of GDMA Interrupts after Masking (Channel-3).** 0: no pending interrupt at channel-3. 1: with pending interrupt at channel-3.	0x0
2	RO	INT[2]	**Status of GDMA Interrupts after Masking (Channel-2).** 0: no pending interrupt at channel-2. 1: with pending interrupt at channel-2.	0x0
1	RO	INT[1]	**Status of GDMA Interrupts after Masking (Channel-1).** 0: no pending interrupt at channel-1. 1: with pending interrupt at channel-1.	0x0
0	RO	INT[0]	**Status of GDMA Interrupts after Masking (Channel-0).** 0: no pending interrupt at channel-0. 1: with pending interrupt at channel-0.	0x0

3.3.2 Terminal Count Interrupt Status

Short Name: INTR_TC
Address: 0x04

Table 47. Terminal Count Interrupt Status Register

Bits	Type	Name	Description	Default
7	RO	INT_TC[7]	**Status of GDMA Terminal Count Interrupts**	0x0

			After Masking (Channel-7) 0: no pending interrupt at channel-7. 1: with pending interrupt at channel-7.	
6	RO	INT_TC[6]	**Status of GDMA Terminal Count Interrupts After Masking (Channel-6)** 0: no pending interrupt at channel-6. 1: with pending interrupt at channel-6.	0x0
5	RO	INT_TC[5]	**Status of GDMA Terminal Count Interrupts After Masking (Channel-5)** 0: no pending interrupt at channel-5. 1: with pending interrupt at channel-5.	0x0
4	RO	INT_TC[4]	**Status of GDMA Terminal Count Interrupts After Masking (Channel-4)** 0: no pending interrupt at channel-4. 1: with pending interrupt at channel-4.	0x0
3	RO	INT_TC[3]	**Status of GDMA Terminal Count Interrupts After Masking (Channel-3)** 0: no pending interrupt at channel-3. 1: with pending interrupt at channel-3.	0x0
2	RO	INT_TC[2]	**Status of GDMA Terminal Count Interrupts After Masking (Channel-2)** 0: no pending interrupt at channel-2. 1: with pending interrupt at channel-2.	0x0
1	RO	INT_TC[1]	**Status of GDMA Terminal Count Interrupts After Masking (Channel-1)** 0: no pending interrupt at channel-1. 1: with pending interrupt at channel-1.	0x0
0	RO	INT_TC[0]	**Status of GDMA Terminal Count Interrupts After Masking (Channel-0)** 0: no pending interrupt at channel-0. 1: with pending interrupt at channel-0.	0x0

3.3.3 Terminal Count Interrupt Status Clear

Short Name: INTR_TC_CLR
Address: 0x08

Table 48. Terminal Count Interrupt Status Clear

Bits	Type	Name	Description	Default
7	WC	INT_TC_CLR[7]	Write 1 to clear the INT_TC[7] and TC[7] status	
6	WC	INT_TC_CLR[6]	Write 1 to clear the INT_TC[6] and TC[6] status	
5	WC	INT_TC_CLR[5]	Write 1 to clear the INT_TC[5] and TC[5] status	
4	WC	INT_TC_CLR[4]	Write 1 to clear the INT_TC[4] and TC[4] status	
3	WC	INT_TC_CLR[3]	Write 1 to clear the INT_TC[3] and TC[3] status	
2	WC	INT_TC_CLR[2]	Write 1 to clear the INT_TC[2] and TC[2] status	
1	WC	INT_TC_CLR[1]	Write 1 to clear the INT_TC[1] and TC[1] status	
0	WC	INT_TC_CLR[0]	Write 1 to clear the INT_TC[0] and TC[0] status	

3.3.4 Error Interrupt Status

Short Name: INTR_ERR
Address: 0x0C

Table 49. Error Interrupt Status

Bits	Type	Name	Description	Default
7	RO	INT_ERR[7]	**Status of GDMA Error Interrupts After Masking (Channel-7)** 0: No pending interrupt. 1: With pending interrupt.	0x0
6	RO	INT_ERR[6]	**Status of GDMA Error Interrupts After Masking (Channel-6).** 0: No pending interrupt. 1: With pending interrupt.	0x0
5	RO	INT_ERR[5]	**Status of GDMA Error Interrupts After Masking (Channel-5).** 0: No pending interrupt. 1: With pending interrupt.	0x0
4	RO	INT_ERR[4]	**Status of GDMA Error Interrupts After Masking (Channel-4).** 0: No pending interrupt. 1: With pending interrupt.	0x0
3	RO	INT_ERR[3]	**Status of GDMA Error Interrupts After Masking (Channel-3)** 0: No pending interrupt. 1: With pending interrupt.	0x0
2	RO	INT_ERR[2]	**Status of GDMA Error Interrupts After Masking (Channel-2).** 0: No pending interrupt. 1: With pending interrupt.	0x0
1	RO	INT_ERR[1]	**Status of GDMA Error Interrupts After Masking (Channel-1).** 0: No pending interrupt. 1: With pending interrupt.	0x0
0	RO	INT_ERR[0]	**Status of GDMA Error Interrupts After Masking (Channel-0).** 0: No pending interrupt. 1: With pending interrupt.	0x0

3.3.5 Error Interrupt Status Clear

Short Name: INTR_ERR_CLR
Address: 0x10

Table 50. Error Interrupt Status Clear

Bits	Type	Name	Description	Default
7	WO	INT_ERR_CLR[7]	Write 1 to clear the INT_ERR[7] and ERR[7] status	
6	WO	INT_ERR_CLR[6]	Write 1 to clear the INT_ERR[6] and ERR[6] status	
5	WO	INT_ERR_CLR[5]	Write 1 to clear the INT_ERR[5] and ERR[5] status	
4	WO	INT_ERR_CLR[4]	Write 1 to clear the INT_ERR[4] and ERR[4] status	
3	WO	INT_ERR_CLR[3]	Write 1 to clear the INT_ERR[3] and ERR[3] status	

2	WO	INT_ERR_CLR[2]	Write 1 to clear the INT_ERR[2] and ERR[2] status	
1	WO	INT_ERR_CLR[1]	Write 1 to clear the INT_ERR[1] and ERR[1] status	
0	WO	INT_ERR_CLR[0]	Write 1 to clear the INT_ERR[0] and ERR[0] status	

3.3.6 Terminal Count Status

Short Name: TC_STAT
Address: 0x14

Table 51. Terminal Count Status

Bits	Type	Name	Description	Default
7	RO	TC[7]	**Status of GDMA Terminal Count (Channel-7).** 0: No terminal count status. 1: With terminal count status.	0x0
6	RO	TC[6]	**Status of GDMA Terminal Count (Channel-6).** 0: No terminal count status. 1: With terminal count status.	0x0
5	RO	TC[5]	**Status of GDMA Terminal Count (Channel-5).** 0: No terminal count status. 1: With terminal count status.	0x0
4	RO	TC[4]	**Status of GDMA Terminal Count (Channel-4).** 0: No terminal count status. 1: With terminal count status.	0x0
3	RO	TC[3]	**Status of GDMA Terminal Count (Channel-3).** 0: No terminal count status. 1: With terminal count status.	0x0
2	RO	TC[2]	**Status of GDMA Terminal Count (Channel-2).** 0: No terminal count status. 1: With terminal count status.	0x0
1	RO	TC[1]	**Status of GDMA Terminal Count (Channel-1).** 0: No terminal count status. 1: With terminal count status.	0x0
0	RO	TC[0]	**Status of GDMA Terminal Count (Channel-0).** 0: No terminal count status. 1: With terminal count status.	0x0

3.3.7 Error Status

Short Name: ERR_STAT
Address: 0x18

Table 52. Error Status

Bits	Type	Name	Description	Default
7	RO	ERR[7]	**Status of GDMA Error (Channel-7).** 0: No error status. 1: With error status.	0x0
6	RO	ERR[6]	**Status of GDMA Error (Channel-6).** 0: No error status. 1: With error status.	0x0
5	RO	ERR[5]	**Status of GDMA error (Channel-5).** 0: No error status. 1: With error status.	0x0
4	RO	ERR[4]	**Status of GDMA error (Channel-4).** 0: No error status. 1: With error status.	0x0
3	RO	ERR[3]	**Status of GDMA Error (Channel-3).** 0: No error status. 1: With error status.	0x0
2	RO	ERR[2]	**Status of GDMA Error (Channel-2).** 0: No error status. 1: With error status.	0x0
1	RO	ERR[1]	**Status of GDMA error (Channel-1).** 0: No error status. 1: With error status.	0x0
0	RO	ERR[0]	**Status of GDMA error (Channel-0).** 0: No error status. 1: With error status.	0x0

3.3.8 Channel Enable Status

Short Name: CH_EN_FLAG
Address: 0x1C

Table 53. Channel Enable Status

Bits	Type	Name	Description	Default
7	RO	CH7_EN_FLAG	**Status of Channel 7 CH_EN on the C7_CSR register.** 0: CH_EN = 0. 1: CH_EN = 1.	0x0
6	RO	CH6_EN_FLAG	**Status of Channel 6 CH_EN on the C6_CSR register.** 0: CH_EN = 0. 1: CH_EN = 1.	0x0
5	RO	CH5_EN_FLAG	**Status of Channel 5 CH_EN on the C5_CSR register.** 0: CH_EN = 0. 1: CH_EN = 1.	0x0
4	RO	CH4_EN_FLAG	**Status of Channel 4 CH_EN on the C4_CSR register.** 0: CH_EN = 0. 1: CH_EN = 1.	0x0
3	RO	CH3_EN_FLAG	**Status of Channel 3 CH_EN on the C3_CSR register.**	0x0

			0: CH_EN = 0. 1: CH_EN = 1.	
2	RO	CH2_EN_FLAG	**Status of Channel 2 CH_EN on the C2_CSR register.** 0: CH_EN = 0. 1: CH_EN = 1.	0x0
1	RO	CH1_EN_FLAG	**Status of Channel 1 CH_EN on the C1_CSR register.** 0: CH_EN = 0. 1: CH_EN = 1.	0x0
0	RO	CH0_EN_FLAG	**Status of Channel 0 CH_EN on the C0_CSR register.** 0: CH_EN = 0. 1: CH_EN = 1.	0x0

3.3.9 Channel Busy Status

Short Name: CH_BUSY_FLAG
Address: 0x20

Table 54. Channel Busy Status

Bits	Type	Name	Description	Default
7	RO	CH7_BUSY_FLAG	**Status of Channel 7 BUSY on the C7_CFG Register.** 0: BUSY = 0. 1: BUSY = 1.	0x0
6	RO	CH6_BUSY_FLAG	**Status of Channel 6 BUSY on the C6_CFG Register.** 0: BUSY = 0. 1: BUSY = 1.	0x0
5	RO	CH5_BUSY_FLAG	**Status of Channel 5 BUSY on the C5_CFG Register.** 0: BUSY = 0. 1: BUSY = 1.	0x0
4	RO	CH4_BUSY_FLAG	**Status of Channel 4 BUSY on the C4_CFG Register**. 0: BUSY = 0. 1: BUSY = 1.	0x0
3	RO	CH3_BUSY_FLAG	**Status of Channel 3 BUSY on the C3_CFG Register.** 0: BUSY = 0. 1: BUSY = 1.	0x0
2	RO	CH2_BUSY_FLAG	**Status of Channel 2 BUSY on the C2_CFG Register.** 0: BUSY = 0. 1: BUSY = 1.	0x0
1	RO	CH1_BUSY_FLAG	**Status of Channel 1 BUSY on the C1_CFG Register.** 0: BUSY = 0. 1: BUSY = 1.	0x0
0	RO	CH0_BUSY_FLAG	**Status of Channel 0 BUSY on the C0_CFG Register**. 0: BUSY = 0.	0x0

			1: BUSY = 1.	

3.3.10 Main Configuration Status

Short Name: CSR
Address: 0x24

Table 55. Main Configuration Status

Bits	Type	Name	Description	Default
2	RW	M1ENDIAN	**Master 1 Endian Configuration.** 0: Little-endian mode. 1: Big-endian mode	0x0
1	RW	M0ENDIAN	**Master 0 Endian Configuration.** 0: Little-endian mode. 1: Big-endian mode	0x0
0	RW	DMACEN	**GDMA Controller Enable.** 0: Disable. 1: Enable.	0x0

3.3.11 Synchronous

Short Name: SYNC
Address: 0x28

Table 56. Synchronous

Bits	Type	Name	Description	Default
7	RW	CH7_REQ_SYNC	**Hardware DMA Request Synchronization Logic Enable for Channel 7** 0: Disable. 1: Enable.	0x0
6	RW	CH6_REQ_SYNC	**Hardware DMA Request Synchronization Logic Enable for Channel 6** 0: Disable. 1: Enable.	0x0
5	RW	CH5_REQ_SYNC	**Hardware DMA Request Synchronization Logic Enable for Channel 5** 0: Disable. 1: Enable.	0x0
4	RW	CH4_REQ_SYNC	**Hardware DMA Request Synchronization Logic Enable for Channel 4** 0: Disable. 1: Enable.	0x0
3	RW	CH3_REQ_SYNC	**Hardware DMA Request Synchronization Logic Enable for Channel 3** 0: Disable. 1: Enable.	0x0
2	RW	CH2_REQ_SYNC	**Hardware DMA Request Synchronization Logic Enable for Channel 2** 0: Disable. 1: Enable.	0x0
1	RW	CH1_REQ_SYNC	**Hardware DMA Request Synchronization**	0x0

			Logic Enable for Channel 1 0: Disable. 1: Enable.	
0	RW	CH0_REQ_SYNC	**Hardware DMA Request Synchronization Logic Enable for Channel 0** 0: Disable. 1: Enable.	0x0

3.3.12 Channel 0 - 7 Control

Short Name: C0_CSR – C7_CSR
Address: 0x100 – Channel 0 Control (C0_CSR)
Address: 0x120 – Channel 1 Control (C1_CSR)
Address: 0x140 – Channel 2 Control (C2_CSR)
Address: 0x160 – Channel 3 Control (C3_CSR)
Address: 0x180 – Channel 4 Control (C4_CSR)
Address: 0x1A0 – Channel 5 Control (C5_CSR)
Address: 0x1C0 – Channel 6 Control (C6_CSR)
Address: 0x1E0 – Channel 7 Control (C7_CSR)

Table 57. Channel 0 – 7 Control

Bits	Type	Name	Description	Default
31	RW	TC_MSK	**Descriptor Terminal Count Interrupt Mask** 0: Pass Interrupt 1: Mask Interrupt **Note: When TC_MSK of related Link List Descriptor is 1, then the TC Interrupt will be suppressed when the data movement, pointed by the descriptor, is completed.**	0x0
28:25	RW	HHST_Sel	**Hardware Hand Shake Target Select** 0: PCM-TX-0 1: PCM-RX-0 2: SPI-TX 3: SPI-RX 4: I2S-TX-Left 5: I2S-TX-Right 6: UART0-TX 7: UART0-RX 8: UART1-TX 9: UART1-RX A: USB Device Reserved B: I2S-RX-Left C: I2S-RX-Right D: PCM-TX-1 E: PCM-RX-1 F: Reserved	0xF
23:22	RW	CHPRI	**Channel Priority Level.** 3: highest priority. 2: 2nd high priority. 1: 3rd high priority. 0: lowest priority (default).	0x0
21	RW	PROT3	**HPROT[3]: Protection Information for**	0x0

			Cacheable. 0: Not cacheable (default). 1: Cacheable.	
20	RW	PROT2	**HPROT[2]: Protection Information for Bufferable.** 0: Not bufferable (default). 1: Bufferable.	0x0
19	RW	PROT1	**HPROT[1]: Protection information for the mode** 0: User mode (default). 1: Privileged mode.	0x0
18:16	RW	SRC_SIZE	**Source Burst Size Selection.** 0: burst size = 1(default). 1: burst size = 4. 2: burst size = 8. 3: burst size = 16. 4: burst size = 32. 5: burst size = 64. 6: burst size = 128. 7: burst size = 256. ***Note:*** *Source burst size is not relative to the HBRUST (AHB signals), it just means how many transfers before the DMA re-arbitrate among active channels.*	0x0
15	WO	ABT	**Transaction Abort.** Write 1 to this bit will cause the DMA to stop its current transfer, set the ERR bit and assert interrupt.	0x0
13:11	RW	SRC_WIDTH	**Source Transfer Width.** The hardware automatically packs and unpacks the data as required. 0: Transfer width is 8 bits. 1: Transfer width is 16 bits. 2: Transfer width is 32 bits (default). Others: Reserved. ***Notice:*** *If source transfer width < destination transfer width, DMA will pack input data. For example: source transfer width = 8bit, destination transfer width = 32bit, then DMA will pack four 8bit source data and transfer one 32bit data.* ***Limitation****: Don't' set* ***SRCAD_CTL*** *= 01(decrement source address) when pack function works, DMA will have a wrong action.* *If source transfer width > destination transfer width, DMA will unpack input data. For example: source transfer width = 32bit, destination transfer width = 8bit, then DMA will unpack one 32bit source data and transfer four 8bit data to destination.*	0x2
10:8	RW	DST_WIDTH	**Destination Transfer Width.** The hardware automatically packs and unpacks the data as required. 0: Transfer width is 8 bits. 1: Transfer width is 16 bits.	0x2

			2: Transfer width is 32 bits (default). Others: Reserved.	
7	RW	MODE	**Mode of Operation.** 0: Normal Mode (default). 1: Hardware Handshake Mode.	0x0
6:5	RW	SRCAD_CTL	**Source Address Control.** 0: Increment source address (default) 1: Decrement source address 2: Fixed source address. 3: Reserved. **Notice**: Don't' set **SRCAD_CTL** = 1(decrement source address) when pack function works, DMA will have a wrong action.	0x0
4:3	RW	DSTAD_CTL	**Destination Address Control.** 0: Increment destination address (default). 1: Decrement destination address. 2: Fixed destination address. 3: Reserved	0x0
2	RW	SRC_SEL	**Selection of AHB Master as Source.** 0: AHB Master 0 is the source (default). 1: AHB Master 1 is the source.	0x0
1	RW	DST_SEL	**Selection of AHB Master as Destination.** 0: AHB Master 0 is the destination (default). 1: AHB Master 1 is the destination.	0x0
0	RW	CH_EN	**Channel Enabled** 0: Disable (default). 1: Enable.	0x0

3.3.13 Channel 0 - 7 Configuration

Short Name: C0_CFG – C7_CFG
Address: 0x104 – Channel 0 Configuration (C0_CFG)
Address: 0x124 – Channel 1 Configuration (C1_CFG)
Address: 0x144 – Channel 2 Configuration (C2_CFG)
Address: 0x164 – Channel 3 Configuration (C3_CFG)
Address: 0x184 – Channel 4 Configuration (C4_CFG)
Address: 0x1A4 – Channel 5 Configuration (C5_CFG)
Address: 0x1C4 – Channel 6 Configuration (C6_CFG)
Address: 0x1E4 – Channel 7 Configuration (C7_CFG)

Table 58. Channel 0 – 7 Configuration

Bits	Type	Name	Description	Default
19:16	RO	LLP_CNT	**LLP Counter** When the above Channel Enable is set to 1 (while the DMA transfer is enabled), the LLP_CNT is reset to zero. For each descriptor transaction finished, the LLP_CNT increases by one. Note that when the last transaction is finished, the LLP_CNT also increases by one. When the above Transaction Abort bit is set to 1	0x0

			manually, the chain transfer is stopped and the LLP_CNT also increases by one.	
8	RO	BUSY	The DMA channel is BUSY.	0x0
2	RW	INT_ABT_MSK	**Channel Abort Interrupt Mask.** 0: interrupt enabled. 1: Mask interrupt (default).	0x1
1	RW	INT_ERR_MSK	**Channel Error Interrupt Mask.** 0: interrupt enabled. 1: Mask interrupt (default).	0x1
0	RW	INT_TC_MASK	**Channel Terminal Count Interrupt Mask**. This bit is per channel TC Interrupt Mask. When each descriptor transaction complete, an interrupt will be generated when the bit is 0 and descriptor's TC_MSK bit is 0. 0: interrupt enabled 1: interrupt disabled (default)	0x1

3.3.14 Channel 0 - 7 Source Address

Short Name: C0_SrcAddr – C7_SrcAddr
Address: 0x108 –Channel 0 Source Address (C0_SrcAddr)
Address: 0x128 –Channel 1 Source Address (C1_SrcAddr)
Address: 0x148 –Channel 2 Source Address (C2_SrcAddr)
Address: 0x168 –Channel 3 Source Address (C3_SrcAddr)
Address: 0x188 –Channel 4 Source Address (C4_SrcAddr)
Address: 0x1A8 –Channel 5 Source Address (C5_SrcAddr)
Address: 0x1C8 –Channel 6 Source Address (C6_SrcAddr)
Address: 0x1E8 –Channel 7 Source Address (C7_SrcAddr)

Table 59. Channel 0 – 7 Source Address

Bits	Type	Name	Description	Default
31:0	RW	SrcAddr	**Source Address**	Undefined

3.3.15 Channel 0 - 7 Destination Address

Short Name: C0_DstAddr – C7_DstAddr
Address: 0x10C –Channel 0 Destination Address (C0_DstAddr)
Address: 0x12C –Channel 1 Destination Address (C1_DstAddr)
Address: 0x14C –Channel 2 Destination Address (C2_DstAddr)
Address: 0x16C –Channel 3 Destination Address (C3_DstAddr)
Address: 0x18C –Channel 4 Destination Address (C4_DstAddr)
Address: 0x1AC –Channel 5 Destination Address (C5_DstAddr)
Address: 0x1CC –Channel 6 Destination Address (C6_DstAddr)
Address: 0x1EC –Channel 7 Destination Address (C7_DstAddr)

Table 60. Channel 0 – 7 Destination Address

Bits	Type	Name	Description	Default
31:0	RW	DstAddr	**Destination Address**	Undefined

3.3.16 Channel 0 - 7 Linked List Descriptor Pointer

Short Name: C0_LLP – C7_LLP
Address: 0x110 –Channel 0 Linked List Descriptor Pointer (C0_LLP)
Address: 0x130 –Channel 1 Linked List Descriptor Pointer (C1_LLP)
Address: 0x150 –Channel 2 Linked List Descriptor Pointer (C2_LLP)
Address: 0x170 –Channel 3 Linked List Descriptor Pointer (C3_LLP)
Address: 0x190 –Channel 4 Linked List Descriptor Pointer (C4_LLP)
Address: 0x1B0 –Channel 5 Linked List Descriptor Pointer (C5_LLP)
Address: 0x1D0 –Channel 6 Linked List Descriptor Pointer (C6_LLP)
Address: 0x1F0 –Channel 7 Linked List Descriptor Pointer (C7_LLP)

Table 61. Channel 0 – 7 Linked List Descriptor Pointer

Bits	Type	Name	Description	Default
31:2	RW	LLPAddr	**Linked List Descriptor Pointer Address** Note: When LLPAddr = 0, means this channel's Linked List function is disabled.	0x00000000
0	RW	LLP_Master	**Master for loading the next LLP** 0 = load the next LLP from the AHB Master 0 1 = load the next LLP from the AHB Master 1	0x0

3.3.17 Channel 0 - 7 Transfer Size

Short Name: C0_SIZE – C7_CIZE
Address: 0x114 –Channel 0 Transfer Size (C0_SIZE)
Address: 0x134 –Channel 1 Transfer Size (C1_SIZE)
Address: 0x154 –Channel 2 Transfer Size (C2_SIZE)
Address: 0x174 –Channel 3 Transfer Size (C3_SIZE)
Address: 0x194 –Channel 4 Transfer Size (C4_SIZE)
Address: 0x1B4 –Channel 5 Transfer Size (C5_SIZE)
Address: 0x1D4 –Channel 6 Transfer Size (C6_SIZE)
Address: 0x1F4 –Channel 7 Transfer Size (C7_SIZE)

Table 62. Channel 0 – 7 Transfer Size

Bits	Type	Name	Description	Default
11:0	RW	TOT_SIZE	**Total Transfer Number of Unit.** NOTE: The transfer unit depends on the source width. For example: SRC_WIDTH=0x0, unit: 8bit. SRC_WIDTH=0x1, unit: 16bit. SRC_WIDTH=0x2, unit: 32bit.	Undefined

3.4 Gigabit Ethernet Contoller (GEC)

3.4.1 PHY Control Register 0

Short Name: PHY_CTRL0
Address: 0x000

Table 63. PHY Control Register 0

Bits	Type	Name	Description	Default
31:16	RW	rw_data	**Read/Write Data** For write command, the write data should be ready here before issuing write command. For read command, when rw_ok is asserted, the register data is ready here.	0x0000
15	R/WC	rw_ok	**Read/Write Command Has Completed** write 1 to clear	0x0
14	RW	rd_cmd	**Read Command,** self clear	0x0
13	RW	wt_cmd	**Write Command**, self clear	0x0
12:8	RW	phy_register	**PHY Register Address**	0x00
4:0	RW	phy_addr	**PHY Address** Note, internal FE PHY address is 0x00.	0x00

3.4.2 PHY Control Register 1

Short Name: PHY_CTRL1
Address: 0x004

Table 64. PHY Control Register 1

Bits	Type	Name	Description	Default
31	RW	auto_poll_dis	**PHY Auto Polling Disable** 0: To enable PHY auto polling, 1: To disable PHY auto polling.	0x0
28:24	RW	Phy_addr_auto	**The PHY address used for auto-polling.** Note, internal FE PHY Address is 0x00.	0x00
18	RW	Internal_PHY_Sel	**Select Internal 10/100 PHY** 0: Select RGMII/MII/Reverse MII interface 1: Select Internal 10/100 PHY Note: When bit 1 of e-Fuse is one, the Internal_PHY_Sel bit can be programmed to be 0 or 1; otherwise, it will be kept at 0 always. In other words, internal Fast Ethernet PHY will be disabled when bit 1 of e-Fuse is 0.	0x1
17	RW	RGMII_phy	**RGMII_PHY used.** Setting rgmii_phy will enable the MAC to use RGMII interface and operate at 10/100/1000 triple speeds. Otherwise, MAC will use MII (or reverse-MII, if rev_MII_RGMII is set) interface and can only operate at 10/100 mode of operation. 1: RGMII interface is used 0: MII or reverse-MII interface is used Note: When bit 0 of e-Fuse is one, the RGMII_phy bit can be programmed to be 0 or 1; otherwise, it will be kept at 0 always. In other words, RGMII interface will be disabled when bit 0 of e-Fuse is 0.	0x0
16	RW	rev_MII_RGMII	**Reverse MII Mode Enable.**	0x0

			0: Normal MII/RGMII mode (MAC side) 1: Reverse MII mode (PHY side) *Note: when in reverse MII mode, the interface must be configured by force mode.* Inter-connection in reverse MII mode: (STR81XX/ CNS21XX side) (External MAC side) RXC → TXC TXC → RXC TXD → RXD TXEN → RXDV RXD ← TXD RXDV ← TXEN COL → OL	
14	RW	txc_check_en	**TX Clock Period Checking Enable.** If more than 400ns, disable the MAC port.	0x1
13	RW	force_fc_tx	**Force TX flow control when Auto-Negotiation disabled (only for 1000Mbps mode)** 0: TX flow control OFF. 1: TX flow control ON.	0x1
12	RW	force_fc_rx	**Force Rx Flow Control when Auto-Negotiation disabled.** **10/100Mbps mode:** 0: Flow Control OFF 1: Flow Control ON **1000Mbps mode:** 0: RX Flow Control OFF 1: RX Flow Control ON	0x1
11	RW	force_duplex	**Force Duplex when Auto-Negotiation disabled.** 0: Half-duplex. 1: Full-duplex.	0x1
10:9	RW	force_speed	**Force Speed when Auto-Negotiation disabled.** 0: 10Mbps. 1: 100Mbps. 2: 1000Mbps. 3: reserved.	0x1
8	RW	AN_en	**Auto-Negotiation Enable.** Setting this bit will enable PHY auto-negotiation and use the polling results from PHY as the operation modes. Clearing this bit will disable PHY auto-negotiation and use forced operation modes (forced_spd, forced_dpx…). Also, the forced operation modes will be written to PHY to make the PHY and MAC operations consistent.	0x1
7	RW	MI_Dis	**Management Interface (MDC/MDIO) disable** 0: Active 1: Disable	0x0
6	RO	fc_tx_st	**TX Flow Control Status** (only for 1000Mbps mode).	

			0: TX Flow Control OFF 1: TX Flow Control ON	
5	RO	fc_rx_st	**RX Flow Control Status.** **10/100Mbps mode:** 0: Flow Control OFF 1: Flow Control ON **1000Mbps mode:** 0: RX Flow Control OFF 1: RX Flow Control ON.	
4	RO	duplex_st	**Duplex Status.** 0: Half-duplex. 1: Full-duplex.	
3:2	RO	speed_st	**Speed Status.** 0: 10Mbps. 1: 100Mbps. 2: 1000Mbps. 3: reserved.	
1	RO	txc_st	**TX Clock Status.** 0: Normal. 1: No TXC, or clock period too long.	
0	RO	link_st	**PHY Link Status.** 0: Link down. 1: Link up.	

3.4.3 MAC Configuration

Short Name: MAC_CFG
Address: 0x008

Table 65. MAC Configuration

Bits	Type	Name	**Description**	Default
31	RW	NIC_PD	**NIC Power Down** Power down NIC totally. When assert, the following WoL bit is no effect. Note: When NIC_PD bit asserted, the NIC will not to receive packets any more, and transmit all of queued TX packets. When completed, the following NIC_PD_Ready bit will be asserted. Then CPU will inform Power Management block to gate off NIC clocks.	0x0
30	RW	WoL	**Wake on LAN Enable** Check each RX packet, if it is a Magic Packet, assert WoL interrupt. When asserted, TX MAC is powered down, and RX MAC only check Magic Packet and not forward any packets to system memory.	0x0
29	R/WC	NIC_PD_Ready	**NIC Power Down Ready** NIC has finished sending all received packets to CPU and transmit all of queued TX packets. NIC is in IDLE status and is ready for powering down. Write 1 to clear.	0x0
26	RW	TX_CKS_En	**TX IP/TCP/UDP Checksum offload**	0x0

			Enable The offload engine will generate IPv4/TCP/UDP checksum according to TX Descriptor's assignment.	
25	RW	RX_CKS_En	**RX IP/TCP/UDP Checksum offload Enable** Checksum check will be done at all received IPv4/TCP/UDP packets.	0x0
24	RW	Acpt_CKS_Err	**Accept Check Sum Error packets** 0: Discard Check Sum Error packets 1: Accept Check Sum Error packets	0x0
23	RW	IST_En	**Inter Switch Tag Enable** Handle all the Rx packets as VLAN tagged packets. It is dedicated for inter-switch tag application.	0x0
22	RW	VLAN_stripping	**VLAN Tag Stripping.** 0: Keep original VLAN tag in RX packet. 1: Strip VLAN tag from RX packet. Note: no matter VLAN tag stripped or not, the VLAN tag info will be stamped at RX Descriptor. And inserting VLAN tag in TX packet or not, depend on TX Descriptor's assignment.	0x1
21	RW	Acpt_CRC_Err	**Accept CRC Error packets** 0: Discard CRC Error packets 1: Accept CRC Error packets	0x0
20	RW	crc_stripping	**CRC Stripping.** 0: Keep CRC in RX packet, and not to generate CRC when transmission. 1: Strip CRC when RX and generate CRC when TX.	0x1
18	RW	Acpt_Long_Pkt	**Accept Oversized Packets** 0: Discard oversized packets 1: Accept oversized packets	0x0
17:16	RW	max_len	**Maximum Packet Length.** 0: 1518 bytes. 1: 1522 bytes. 2: 1536 bytes 3: Reserved	0x1
14:10	R/W	IPG	**Inter Packet Gap** **IPG[4:2]** represent the duration of the first 2/3 of inter-packet-gap. The number for this value is set to be # of byte clk –1 **IPG[1:0]** represent the duration of the second 1/3 of inter-packet-gap. The number for this value is set to be # of byte clk –1	0x1F (12-byte IPG)
9	R/W	Do_Not_Skip	**Don't skip 16 consecutive collisions packet** 1: Allow MAC to retransmit a packet after 16 consecutive collisions 0: Standard way to drop a packet after 16 consecutive collisions	0x0
8	R/W	Fast_Retry	**Collision Fast Back-off**	0x0

			1: Force MAC to do fast back-off (0,1,2,3 slots) after collision. 0: Standard exponential back-off	

3.4.4 Flow Control Configuration

Short Name: FC_CFG
Address: 0x00C

Table 66. Flow Control Configuration

Bits	Type	Name	Description	Default
27:16	RW	Send_Pause_Th	**Send Pause On Frame Threshold.** Unit: Byte No matter RX Descriptor is available or not, when the threshold in RX FIFO is reached, send pause-on frame when full-duplex or assert back-pressure when half-duplex. When data byte count in RX FIFO is less than Send_Pause_RLS, send pause-off frame when full-duplex or assert back-pressure when half-duplex.	0x3B0
8	RW	uc_pause_dis	**Disable to treat unicast pause frames as 802.3x pause frames.** 0: Enable uc_pause 1: Disable uc_pause Note that the following conditions must be met to treat a packet as a unicast pause frame 1. DA is My_MAC 2. TYPE and OP-code fields match pause frame format When disable uc_pause, unicast pause frame will be forwarded to CPU.	0x1
7	R/W	BP_Enable	**Half Duplex Back Pressure Enable** 1: Enable back-pressure 0: Disable back-pressure	0x1
6	R/W		**Reserved** The bit must be kept at 0.	0x0
5	R/W	max_bp_col_en	**Pass-one-every-N backpressure collision policy enable** 1: When consecutive collisions exceed the max_bp_col_cnt, the embedded MAC will stop back-pressure and allow one successful incoming packet to avoid excessive collisions seen by external hubs. 0: allow unlimited back-pressure collisions in order to prevent packet loss in back-pressure enabled environment.	0x0
4:0	R/W	max_bp_col_cnt	**Max backpressure collision count** Max allowed consecutive collisions in back	0x0C

			pressure process in half-duplex mode operation. The value is meaningful only when max_bp_col_en is set	

3.4.5 ARL Configuration

Short Name: ARL_CFG
Address: 0x010

Table 67. ARL Configuration

Bits	Type	Name	Description	Default
4	RW	Misc_mode	**Miscellaneous mode** Bypass MAC DA matching step and all of incoming CRC good packets will be received to CPU. 0: disable 1: enable, My_MAC_Only is no effect.	0x0
3	RW	My_MAC_Only	**My MAC Only** 0: My MAC or BC or Hash Table hit packets are received. 1: Only My MAC or BC packets are received.	0x0
2	RW	CPU_learn_dis	**From CPU SA Learning Disable.** 0: Learn TX packet SA into hash table 1: Disable auto learning. Note: From MAC SA is always not learned into hash table.	0x1
1	RW	rev_mc_filter	**Reserved Multicast Address Filtering.** 0: Forward to CPU 1: Drop *Note: Reserved MC:* *01-80-C2-00-00-00 (BPDU)* *01-80-C2-00-00-02 ~ 0F* *Note: It is independent with hash table matching result.*	0x0
0	RW	hash_alg	**MAC Address Hashing Algorithm.** 0: Direct mode, using DA last 9-bit [40, 7-0] as hashing address. 1: using 32 bit CRC [8:0] of DA as hashing address.	0x0

3.4.6 My MAC High Byte

Short Name: My_MAC_H
Address: 0x014

Table 68. My MAC High Byte

Bits	Type	Name	Description	Default
15:0	RW	My_MAC[47:32]	**My MAC [47:32]** *Note: The first received/transmitted byte of*	0x0000

			DA from network is deposited at high byte first (My-MAC[47:40]). For example, Pause Frame MAC[47:0] = 0x0180c2000001.	

3.4.7 My MAC Low Byte

Short Name: My_MAC_L
Address: 0x018

Table 69. My MAC Low Byte

Bits	Type	Name	Description	Default
31:0	RW	My_MAC[31:0]	**My MAC [31:0]**	0x00000000

3.4.8 Hash Table Control

Short Name: HASH_CTRL
Address: 0x01C

Table 70. Hash Table Control

Bits	Type	Name	Description	Default
17	RO	Hash Table BIST Done	**Hash Table BIST Done** After Power On Reset or Software Reset, the Build In Self Test Logic will auto-check Hash Table's validation of each bit. When completed, all bits of the Hash Table are cleared to 0. 0: Under test. 1: Self Test complete, and the following flag is valid.	0x0
16	RO	Hash Table BIST OK	**Hash Table BIST OK** After Power On Reset or Software Reset, the Build In Self Test Logic will auto-check Hash Table's validation of each bit. When completed, all bits of the Hash Table are cleared to 0. 0: The hash table is with manufacture fault. 1: Self Test OK.	0x0
14	RW	Command Start	**Hash Access Command Start** When this bit is asserted by software, the following Hash Table access command will be executed. When completed, the bit will be cleared by hardware.	0x0
13	RW	Hash Access Command	**Hash Access Command** 0: read 1: write	0x0
12	RW	Hash Bit Data	**Hash Bit Data** For write command, the hash bit data should be prepared before issuing the write command. For read command, the read hash bit data is showed here when read complete.	0x0

8:0	RW	Hash Bit Address	**Hash Bit Address**	0x000

3.4.9 My VLAN ID Control

Short Name: VLAN_CTRL
Address: 0x020

Table 71. My VLAN ID Control

Bits	Type	Name	Description	Default
3:0	RW	My_VID0~3_EN	**My VID 0~3 Filter Enable** When at least one of My-VIDs' are enabled, RX MAC will compare tagged VID of received packet with enabled My_VIDs. If one of them matched, the packet will be received. When not matched, the packet will be dropped, and the related MIB counter will be increased by 1. 0: Don't filter packet by VLAN ID. 1: Filter packet by My_VID0 2: Filter packet by My_VID1 4: Filter packet by My_VID2 8: Filter packet by My_VID3 Others: Filter packet by enabled My_VIDs. Note: When received packet is without VLAN tag, the VLAN filter setting is no effect. Note: When received packet is tagged with VID 0x000 or 0xFFF, the packet is forwarded to CPU.	0x0

3.4.10 My VLAN ID 0 – 1

Short Name: VLAN_ID_0_1
Address: 0x024

Table 72. My VLAN ID 0 - 1

Bits	Type	Name	Description	Default
27:16	RW	My_VID1	**My VLAN ID 1**	0x000
11:0	RW	My_VID0	**My VLAN ID 0**	0x000

3.4.11 My VLAN ID 2 – 3

Short Name: VLAN_ID_2_3
Address: 0x028

Table 73. My VLAN ID 2 - 3

Bits	Type	Name	Description	Default
27:16	RW	My_VID3	**My VLAN ID 3**	0x000
11:0	RW	My_VID2	**My VLAN ID 2**	0x000

3.4.12 DMA Configuration

Short Name: DMA_CFG
Address: 0x030

Table 74. DMA Configuration

Bits	Type	Name	Description	Default
16	RW	RX_Offset_2B_dis	**RX 2 Bytes Offset Disable** 0: de-align to (4*N+2) address offset. 1: align to word-aligned (4*N) address offset. **Note: The valid memory address pointed in the field seg_data_ptr of RX DMA descriptor shall be either 4*N+2 or 4*N-aligned.**	0x0
7:6	RW	TX_POLL_PERIOD	TX_DMA Auto-Poll Period 0: 1us, 1: 10us, 2: 100us 3: 1000us.	0x0
5	RW	TX_POLL_EN	TX_DMA Auto-Poll C-Bit Enable 0: disable, 1: when TX_EN=0, TX_DMA will auto-poll C-Bit of the TX descriptor in a period defined on TX_POLL_PERIOD.	0x0
4	RW	TX_SUSPEND	TX_DMA Suspend 0: TX_DMA can transmit the packet normally. 1: TX_DMA is suspended.	0x0
3:2	RW	RX_POLL_PERIOD	RX_DMA Auto-Poll Period 0: 1us, 1: 10us, 2: 100us 3: 1000us.	0x0
1	RW	RX_POLL_EN	RX_DMA Auto-Poll C-Bit Enable 0: disable, 1: when RX_EN=0, RX_DMA will auto-poll C-Bit of the RX descriptor in a period defined on RX_POLL_PERIOD.	0x0
0	RW	RX_SUSPEND	RX_DMA Suspend 0: RX_DMA can receive the packet normally. 1: RX_DMA is suspended.	0x0

3.4.13 TX_DMA Control

Short Name: TX_DMA_CTRL
Address: 0x034

Table 75. TX_DMA Control

Bits	Type	Name	Description	Default
0	RW	TX_EN	● Writing 1 will start the TX_DMA to transmit the packet if there are packets stored in DRAM chained by TX_Descriptor. ● Writing 0 will stop TX_DMA operation after the current ongoing whole packet is transmitted. ● TX_DMA will clear this bit, if it reads a descriptor with C=1.	0x0

3.4.14 RX_DMA Control

Short Name: RX_DMA_CRTL
Address: 0x038

Table 76. RX_DMA Control

Bits	Type	Name	Description	Default
0	RW	RX_EN	● Writing 1 will start RX_DMA to receive the incoming packets from NIC. If RX_DMA is already in the running mode (not in IDLE state, but including WAIT_RX_DATA state), writing 1 to this bit will take no effect. ● Writing 0 will stop RX_DMA after RX_DMA completes the current whole packet moving and return to IDLE state. If RX_DMA is already in the IDLE state, writing 0 will take no effect. ● This bit will be cleared by HW, if RX_DMA reads a descriptor with C=1	0x0

3.4.15 TX Descriptor Pointer

Short Name: TX_DPTR
Address: 0x03C

Table 77. TX Descriptor Pointer

Bits	Type	Name	Description	Default
31:4	RW	TXSD	**TX Descriptor Starting Address.** This field indicates the starting address of the TX Descriptor chain. TX_DMA reads the descriptor from this location when it is enabled. The address is cache line alignment.	0x0000000

3.4.16 RX Descriptor Pointer

Short Name: RX_DPTR

Address: 0x040

Table 78. RX Descriptor Pointer

Bits	Type	Name	Description	Default
31:4	RW	RXSD	**RX Descriptor Starting Address.** This field indicates the starting address of the RX Descriptor chain. RX_DMA reads the descriptor from this location when it is enabled. The address is cache-line alignment.	0x0000000

3.4.17 TX Descriptor Base Address

Short Name: TX_BASE_ADDR
Address: 0x044

Table 79. TX Descriptor Base Address

Bits	Type	Name	Description	Default
31:4	RW	TX_BASE	**Base Address of TX Descriptor** The address is with cache line alignment.	0x0000000

3.4.18 RX Descriptor Base Address

Short Name: RX_BASE_ADDR
Address: 0x048

Table 80. RX Descriptor Base Address

Bits	Type	Name	Description	Default
31:4	RW	RX_BASE	**Base Address of RX Descriptor** The address is with cache line alignment.	0x0000000

3.4.19 Delayed Interrupt Configuration

Short Name: DLY_INT_CFG
Address: 0x04C

Table 81. Delayed Interrupt Configuration

Bits	Type	Name	Description	Default
16	RW	DLEAY_INT_EN	1: Enable delayed interrupt mechanism. 0: Disable delayed interrupt mechanism.	0x0
15:8	RW	MAX_PEND_INT_CNT	**Specified Max # of pended RXRC_INT.** When the # of pended RXRC_INT equal or grater than the value specified here or interrupt pending time reach the limit(See bellow), an Final RXRC_INT is generated.	0x00
7:0	RW	MAX_PEND_TIME	**Specified Max pending time for the internal RXRC_INT.** When the pending time equal or greater MAX_PEND_TIME x 20us or , the # of pended RXRC_INT equal or grater than	0x00

			MAX_PEND_INT_CNT (see above), an Final RXRC_INT is generated	

3.4.20 Interrupt Status

Short Name: INT
Address: 0x050
It will generate level trigger interrupt, and all of the register's bits are "write 1 clear.

Table 82. Interrupt Status

Bits	Type	Name	Description	Default
4	R/WC	MAGIC_PKT_REC	**Magic packet Received** Asserted when a Magic packet received.	0x0
3	R/WC	MIB_counter_th	**Assert when any one MIB counter reach 0x8000-0000.**	0x0
2	R/WC	Port_status_chg	**Assert MAC Port Change Link State.** (link up ←→ link down).	0x0
1	R/WC	RX_FIFO_full	Assert when RX Buffer full.	0x0
0	R/WC	TX_FIFO_under_run	**Assert when TX Buffer under run when transmitting a packet.** Note: TX_FIFO should not under-run at store and forward scheme	0x0

3.4.21 Interrupt Mask

Short Name: INT_MASK
Address: 0x054

Table 83. Interrupt Mask

Bits	Type	Name	Description	Default
4:0	RW	Int_mask	Interrupt Mask of Interrupt Status Register bit 4~0.	0x1F

3.4.22 Test 0 (Clock Skew Setting)

Short Name: TEST0
Address: 0x058

Table 84. Test 0 (Clock Skew Setting)

Bits	Type	Name	Description	Default
3:2	RW	tx_skew	The skew adjustment between RGMII Tx clock and Tx data/enable. To meet RGMII 1.3 timing spec, please re-configure it as 0x2. 0: -0.5 ns. 1: -0.5 ns. 2: 0.0 ns. 3: 0.5 ns.	0x1
1:0	RW	rx_skew	The skew adjustment between RGMII Rx clock and Rx data/enable.	0x0

			0: 0 ns. 1: 1.5 ns. 2: 2.0 ns. 3: 2.5 ns.	

3.4.23 Test 1 (Queue Status)

Short Name: TEST1
Address: 0x05C

Table 85. Test 1 (Queue Status)

Bits	Type	Name	Description	Default
18	RW	IntLB_MII	MII internal loop back test mode enable	0x0
17	RW	ExtLB_MII	MII external loop back test mode enable	0x0
16	RW	sim_mode	**Simulation Mode to Accelerate Timers** 1: MIB counter reach 32 will trigger interrupt.	0x0
12:0	RO	free_byte_cnt	**Free Byte Count of RX FIFO.**	0x1000

3.4.24 Extended Configuration Register

Short Name: Extend_CFG
Address: 0x060

Table 86. Extended Configuration

Bits	Type	Name	Description	Default
27:16	RW	Send_Pause_RLS	**Send Pause-Off Frame Threshold** Unit: Byte	0x0

3.4.25 MIB Counters

0x100 - 0x137—MIB Counters
Note:
1. Less offset counter has higher priority to count packets when match. Basically, one packet is only counted in one counter, except to accept abnormal packets (CRC error, oversize, or L3/L4 check sum error) to system memory. When NIC is configured to forward abnormal packets to system memory, those packets will be counted in “RX OK Packet Counter”, and “RX OK Byte Counter” and related abnormal packet counter, for example, “RX Over Size Packet Counter”.
2. All these MIB counters are read clear.

3.4.26 RX OK Packet Counter

Short Name: C_RXOKPKT
Address: 0x100

Table 87. RX OK Packet Counter

Bits	Type	Name	Description	Default
31:0	RC	RXOKPkt	**Packet count of packets forwarding to system memory successfully.**	0x00000000

			Note, when NIC is configured to forward CRC Error packets or Over Size packets or L3/L4 Check Sum Error packets to CPU, those packets will be counted here also.	

3.4.27 RX OK Byte Counter

Short Name: C_RXOKBYTE
Address: 0x104

Table 88. RX OK Byte Counter

Bits	Type	Name	Description	Default
31:0	RC	RXOKByte	**Byte count of packets forwarding to system memory successfully.**	0x00000000

3.4.28 RX Runt Packet Counter

Short Name: C_RXRUNT
Address: 0x108

Table 89. RX Runt Packet Counter

Bits	Type	Name	Description	Default
31:0	RC	RXRunt	**Runt packets count (<64 bytes), no matter they are CRC good or not.**	0x00000000

3.4.29 RX Over Size Packet Counter

Short Name: C_RXLONG
Address: 0x10C

Table 90. RX Over Size Packet Counter

Bits	Type	Name	Description	Default
31:0	RC	RXOverSize	**Over Size packets count, no matter they are CRC good or drop or not.**	0x00000000

3.4.30 RX No Buffer Drop Packet Counter

Short Name: C_RXDROP
Address: 0x110

Table 91. RX No Buffer Drop Packet Counter

Bits	Type	Name	Description	Default
31:0	RC	RXNoBuf	**RX No Buffer drop packet counts, no matter CRC good or not.**	0x00000000

3.4.31 RX CRC Error Packet Counter

Short Name: C_RXCRC
Address: 0x114

Table 92. RX CRC Error Packet Counter

Bits	Type	Name	Description	Default
31:0	RC	RXCRC	**CRC Error packet count, no matter forward to CPU or not.**	0x00000000

3.4.32 RX ARL Drop Packet Counter

Short Name: C_RXARLDROP
Address: 0x118

Table 93. RX ARL Drop Packet Counter

Bits	Type	Name	Description	Default
31:0	RC	RXARL	**ARL drop packet count, CRC good.** ARL drop include My_MAC, Hash Table, and Reserved Multicast Packets filters.	0x00000000

3.4.33 My VLAN ID Mismatch Drop Counter

Short Name: C_RXVLANDROP
Address: 0x11C

Table 94. My VLAN ID Mismatch Drop Counter

Bits	Type	Name	Description	Default
31:0	RC	RXMyVID	**My VLAN ID Mismatch Drop count.** Packet drop count of My VLAN ID mismatch	0x00000000

3.4.34 RX Check Sum Error Packet Counter

Short Name: C_RXCSERR
Address: 0x120

Table 95. RX Check Sum Error Packet Counter

Bits	Type	Name	Description	Default
31:0	RC	RXCS	**Checksum Error, but CRC good packet count, no matter forward to CPU or not.**	0x00000000

3.4.35 RX Pause Frame Packet Counter

Short Name: C_RXPAUSE
Address: 0x124

Table 96. RX Pause Frame Packet Counter

Bits	Type	Name	Description	Default

31:0	RC	RXPause	**RX Pause Frame count, CRC good**	0x00000000

3.4.36 TX OK Packet Counter

Short Name: C_TXOKPKT
Address: 0x128

Table 97. TX OK Packet Counter

Bits	Type	Name	Description	Default
31:0	RC	TXOKPkt	**Packet count of successfully transmitted packets.**	0x00000000

3.4.37 TX OK Byte Counter

Short Name: C_TXOKBYTE
Address: 0x12C

Table 98. TX OK Byte Counter

Bits	Type	Name	Description	Default
31:0	RC	TXOKByte	**Byte count of successfully transmitted packets.**	0x00000000

3.4.38 TX Collision Counter / Pause Frame Counter

Short Name: C_TXPAUSECOL
Address: 0x130

Table 99. TX Collision Counter / Pause Frame Counter

Bits	Type	Name	Description	Default
31:0	RC	TXPause	**In Full Duplex mode, it is transmitted pause frame count.** **In Half Duplex mode, it is transmit collision count.**	0x00000000

3.5 SPI / PCM / TWI / I2S

3.5.1 TWI Control

Short Name: TWI_CTRL
Address: 0x20

Table 100. TWI Control

Bits	Type	Name	Description	Default
31	RW	TWI_EN	**TWI Enable** This bit enables or disables the TWI bus function. 0: The SDA and SCL outputs are in high	0x0

			impedance. SDA and SCL input signals are ignored; TWI is in the “not addressed” slave state. No any interrupt will be requested by TWI bus. 1: TWI bus enabled. TWI will enter the master mode.	
24	RW	TWI_SWAP_EN	**TWI Data Swap Enable** This bit determines if the byte-order of write data from APB to TWI_WR_DAT register or read data from TWI_RD_DAT register to APB are swapped. PWDATA is APB write bus and PRDATA is APB read bus. 0: no swap, TWI_WR_DAT = PWDATA[31:0], PRDATA = TWI_RD_DAT[31:0]. 1: data swapped, TWI_WR_DAT = { PWDATA[7:0], PWDATA[15:8], PWDATA[23:16], PWDATA[31:24] }, PRDATA = {TWI_RD_DAT [7:0], TWI_RD_DAT [15:8], TWI_RD_DAT [23:16], TWI_RD_DAT [31:24] }.	0x0
6	RW	TWI_RUN_START	**TWI Run Start** This bit determines if the TWI-bus runs the transfer operation. It will be auto-cleared by hardware after transfer done. 0: TWI-bus is idle, 1: let TWI-bus run the transfer operation.	0x0
5:4	RW	TWI_TRANSFER_CMD	**TWI Transfer Command** These bits determine which transfer command is assigned to the TWI-bus. 0: Only-Read operation, 1: Only-Write operation, 2: Write then read operation, 3: Read then Write operation.	0x0
3:2	RW	TWI_WRDAT_LEN	**TWI Write Data Length** These bits determine the length of the write data on TX buffer. 0: 1 bytes, the write data [7:0] is valid, 1: 2 bytes, the write data [15:0] is valid, 2: 3 bytes, the write data [23:0] is valid, 3: 4 bytes, the write data [31:0] is valid.	0x0
1:0	RW	TWI_RDDAT_LEN	**TWI Read Data Length** These bits determine the length of the read data on RX buffer. 0: 1 bytes, the read data [7:0] is valid, 1: 2 bytes, the read data [15:0] is valid, 2: 3 bytes, the read data [23:0] is valid, 3: 4 bytes, the read data [31:0] is valid.	0x0

3.5.2 TWI Time-Out

Short Name: TWI_TIMEOUT
Address: 0x24

Table 101. TWI Time-Out

Bits	Type	Name	Description	Default
16:8	RW	TWI_CLKDIV	**TWI Clock Divider** TWI Clock frequency = PCLK / 2*(TWI_CLKDIV + 1)	0x03E
7	RW	TWI_TIME_OUT_EN	**TWI Time-Out Enable Register** This bit is used as a time-out enable or disable for bus-stack error check. 0: Disable the time-out function, 1: Enable the time-out function.	0x0
6:0	RW	TWI_TIME_OUT_VAL	**TWI Time-Out Value Register** This defines the time-out period = (TIME_OUT_OUT_VAL + 1) * TWI clock cycle time.	0x40

3.5.3 TWI Slave Address

Short Name: TWI_SLAVE_ADDR
Address: 0x28

Table 102. TWI Slave Address

Bits	Type	Name	Description	Default
7:1	RW	TWI_SLAVE_ADDR	**TWI Slave Address** To determine which Slave is addressed. The most significant bit corresponds to the first bit received from the TWI-bus after a start condition. When TWI_SLAVE_ADDR[7:3] = 0x1E, 10-bit slave address mode is supported.	0x00

3.5.4 TWI Write Data

Short Name: TWI_WR_DATA
Address: 0x2C

Table 103. TWI Write Data

Bits	Type	Name	Description	Default
31:0	RW	TWI_WR_DAT	**TWI Write Data Register** It contains maximum 4 bytes of serial data to be transmitted. byte 0: bit 7:0, first transmitted, byte 1: bit 15:8, byte 2: bit 23:16, byte 3: bit 31:24.	0x00000000

3.5.5 TWI Read Data

Short Name: TWI_RD_DATA
Address: 0x30

Table 104. TWI Read Data

Bits	Type	Name	Description	Default
31:0	RO	TWI_RD_DAT	**TWI Read Data Register** It contains maximum 4 bytes of serial data to be received. byte 0: bit 7~0, first received, byte 1: bit 15:8, byte 2: bit 23:16, byte 3: bit 31:24.	0x00000000

3.5.6 TWI Interrupt Status

Short Name: TWI_INTR_STAT
Address: 0x34

Table 105. TWI Interrupt Status

Bits	Type	Name	Description	Default
15:8	RO	TWI_STATUS	**TWI Status Register** The three least significant bits are always zero. The five most significant bits contain the status code. There are 5 possible status codes. 0x20: Slave Address + W has been transmitted, and Slave's Not-Acknowledge(NACK) has been received. 0x30: Data byte in WR_DATA_REG has been transmitted, and NACK has been received. 0x48: Slave Address + R has been transmitted, and NACK has been received. 0x70: Bus error, SDA stuck low. 0x90: Bus error, SCL stuck low. 0xFF: Normal. Others: Reserved.	0xFF
1	R/WC	TWI_ACTDONE_FG	**TWI Action Done Status Flag** The bit is asserted, when TWI has finished bus action. A write of a 1 clears the corresponding bit and a write of a 0 has no effect. The bit also can be cleared after the RUN_START register set by the host. 0: Transfer command is on going or there is no transfer command, no interrupt, 1: Transfer command is done, interrupt is asserted if enabled.	0x0
0	R/WC	TWI_BUSERR_FG	**TWI Error Flag** The bit is asserted, when TWI has finished bus action and there is an error existing. The detail is described by status register. A write of a 1 clears	0x0

			the corresponding bit and a write of a 0 has no effect. 0: TWI is normal, no interrupt, 1: There is an error existing after TWI action done, interrupt is asserted if enabled.	

3.5.7 TWI Interrupt Enable

Short Name: TWI_INTR_ENA
Address: 0x38

Table 106. TWI Interrupt Enable

Bits	Type	Name	Description	Default
1	RW	TWI_ACTDONE_IEN	**TWI Action Done Interrupt Enable** The bit determines whether TWI-Bus Action Done Interrupt enable. 0: disable, 1: enable.	0x0
0	RW	TWI_BUSERR_IEN	**TWI Error Interrupt Enable** The bit determines whether TWI-Bus Error Interrupt enable. 0: disable, 1: enable.	0x0

3.5.8 SPI Configuration

Short Name: SPI_CFG
Address: 0x40

Table 107. SPI Configuration

Bits	Type	Name	Description	Default
31	RW	SPI_EN	**SPI Enable** This bit enables and disables the SPI interface operation. 0: SPI operation is disabled, 1: SPI operation is enabled.	0x1
30	RW	SPI_BOOT_HIGH_SPEED	**SPI High Speed Read For System Boot Up** This bit determines whether System Boot-Up code is read in high speed. 0: disable, 1: enable.	0x0
24	RW	SPI_SWAP_EN	**SPI Data Swap Enable** This bit determines if the byte-order of write data from APB to SPI_TXDAT register or read data from SPI_RXDAT register to APB are swapped. PWDATA is APB write bus and PRDATA is APB read bus. 0: no swap, SPI_TXDAT = PWDATA[31:0], PRDATA = SPI_RXDAT[31:0]. 1: data swapped, SPI_TXDAT = { PWDATA[7:0],	0x0

			PWDATA[15:8], PWDATA[23:16], PWDATA[31:24] }, PRDATA = { SPI_RXDAT[7:0], SPI_RXDAT[15:8], SPI_RXDAT[23:16], SPI_RXDAT[31:24] }.	
14	RW	SPI_CLKPOL	**SPI Clock Polarity** This register determines the logic state of the SPICLK pin between transmissions. 0: when no SPI data is sent, SPICLK is at low level 1: when no SPI data is sent, SPICLK is at high level	0x0
13	RW	SPI_CLKPHA	**SPI Clock Phase** This register controls the timing relationship between the serial clock and SPI data. 0: First SPICLK edge at 1/2 cycle after start of transmission 1: First SPICLK edge at start of transmission	0x0
12	RW	SPI_LBK	**SPI Loop back** Loop back mode allows module validation during device testing. This mode is valid only in master mode of the SPI. 0: SPI loop back mode disabled 1: SPI loop back mode enabled	0x0
11	RW	SPI_MASTER_EN	**SPI Network Mode Control** The bit determines whether the SPI is a network master or slave. 0: SPI configured as slave 1: SPI configured as master	0x0
10	RW	SPI_FFEN	**SPI FIFO Enhancements Enable** 0: SPI FIFO enhancements are disabled 1: SPI FIFO enhancements are enabled	0x0
9	RW	SPI_SERIAL_MODE	**SPI Serial Mode Selection** 0: General SPI, 1: Microprocessor Interface (MPI).	0x0
1:0	RW	SPI_CHAR_LEN	**SPI Character Length Control** These bits determine the number of bits to be shifted in or out for a single character during one shift sequence. 0: 8 bits, 1: 16 bits, 2: 24 bits, 3: 32 bits	0x0

3.5.9 SPI Service Status

Short Name: SPI_STAT
Address: 0x44

Table 108. SPI Service Status

Bits	Type	Name	Description	Default
0	RO	SPI_BUSY_STA	**SPI Channel Busy Status** The bit determines whether SPI-bus is busy. Only for the master SPI-bus. 0: SPI-bus is idle, 1: SPI-bus is busy.	0x0

3.5.10 SPI Bit Rate

Short Name: SPI_BIT_RATE
Address: 0x48

Table 109. SPI Bit Rate

Bits	Type	Name	Description	Default
2:0	RW	SPI_BIT_RATE	**SPI BIT RATE (Baud) Control** These bits determine the bit transfer rate if the SPI is the network master. There are 8 data-transfer rates that can be selected. 0: PCLK/1 1: PCLK/2 2: PCLK/4 3: PCLK/8 4: PCLK/16 5: PCLK/32 6: PCLK/64 7: PCLK/128	0x2

3.5.11 SPI Transmit Control

Short Name: SPI_TX_CTRL
Address: 0x4C

Table 110. SPI Transmit Control

Bits	Type	Name	Description	Default
2	RW	SPI_ TXDAT _EOF	**SPI Back-to-Back Transmit End** The bit determines whether the back-to-back transmit is end. The transmit data is the last one of the back-to-back transmit. The corresponding “SPI slave select” pin will keep low between transmits. And the pin is de-asserted, when the transmit data shift-out done. If the SPI-bus is slave, the bit is ignored. 0: The transmit data is not end. 1: The transmit data is end.	0x1
1:0	RW	SPI_TXCH_NUM	**SPI Serviced Channel Number** These bits determine which channel is serviced by SPI-bus. If the SPI-bus is slave, these bits are ignored. 0: channel 0 to be serviced, 1: channel 1 to be serviced,	0x0

			2: channel 2 to be serviced, 3: channel 3 to be serviced,	

3.5.12 SPI Transmit Data

Short Name: SPI_TX_DATA
Address: 0x50

Table 111. SPI Transmit Data

Bits	Type	Name	Description	Default
31:0	WO	SPI_TXDAT	**SPI Transmit Data Buffer** This register contains the data to be transmitted through the Transmit path. The MSB is the first bit to be transmitted and the LSB is the last bit. CHAR_LEN SPI_TXDAT 8 bit [7:0] valid, MSB bit is [7], 16 bit [15:0] valid, MSB bit is [15], 24 bit [23:0] valid, MSB bit is [23], 32 bit [31:0] valid, MSB bit is [31].	0x00000000

3.5.13 SPI Receive Control

Short Name: SPI_RX_CTRL
Address: 0x54

Table 112. SPI Receive Control

Bits	Type	Name	Description	Default
2	RO	SPI_ RXDAT _EOF	**SPI Back-to-Back Receive End** The bit determines whether the back-to-back receive is end. The receive data is the last one of the back-to-back receive. The corresponding “SPI slave select” pin will keep low between receives. And the pin is de-asserted, when the receive data shift-in done. 0: The receive data is not end. 1: The receive data is end.	0x1
1:0	RO	SPI_RXCH_NUM	**SPI Serviced Channel Number** These bits determine which channel is serviced by SPI-bus. If the SPI-bus is slave, these bits are ignored. 0: channel 0 to be serviced, 1: channel 1 to be serviced, 2: channel 2 to be serviced, 3: channel 3 to be serviced,	0x0

3.5.14 SPI Receive Data

Short Name: SPI_RX_DATA
Address: 0x58

Table 113. SPI Receive Data

Bits	Type	Name	Description	Default
31:0	RO	SPI_RXDAT	**SPI Receive Data Buffer** This register contains the data to be received through the Receive path. The MSB is the first bit to be received and the LSB is the last bit. CHAR_LEN SPI_RXDAT 8 bit [7:0] valid, MSB bit is [7], 16 bit [15:0] valid, MSB bit is [15], 24 bit [23:0] valid, MSB bit is [23], 32 bit [31:0] valid, MSB bit is [31].	0x00000000

3.5.15 SPI FIFO Transmit Configuration

Short Name: SPI_FIFO_TX_CFG
Address: 0x5C

Table 114. SPI FIFO Transmit Configuration

Bits	Type	Name	Description	Default
5:4	RW	SPI_TXFF_THRED	**SPI Transmit FIFO Interrupt Threshold** Transmit FIFO will generate interrupt (if enabled) when the FIFO status value is less than or equal to FIFO threshold. 0: 2 1: 4 2: 6 3: reserved	0x1
3:0	RO	SPI_TXFF_STATUS	**SPI Transmit FIFO status** 0: Transmit FIFO is empty, 1: Transmit FIFO has 1 word, 2: Transmit FIFO has 2 words, . . . 8: Transmit FIFO has 8 words, Others: reserved.	0x0

3.5.16 SPI FIFO Transmit Control

Short Name: SPI_FIFO_TX_CTRL
Address: 0x60

Table 115. SPI FIFO Transmit Control

Bits	Type	Name	Description	Default
4:0	RW	SPI_TXFF_DLY	**SPI FIFO Transmit Delay** These bits define the delay between every transfer from FIFO transmit buffer to transmit shift register. The delay is defined in the number SPI serial clock cycles. The 5 bit register could define a minimum delay of 0 serial clock cycles and a maximum of 31 serial cycles.	0x00

3.5.17 SPI FIFO Receive Configuration

Short Name: SPI_FIFO_RX_CFG
Address: 0x64

Table 116. SPI FIFO Receive Configuration

Bits	Type	Name	Description	Default
5:4	RW	SPI_RXFF_THRED	**SPI Receive FIFO Interrupt Threshold** Receive FIFO will generate interrupt (if enabled) when the FIFO status value is greater than or equal to the FIFO threshold value. 0: 2 1: 4 2: 6 3: reserved	0x1
3:0	RO	SPI_RXFF_STATUS	**SPI Receive FIFO status** 0: Receive FIFO is empty, 1: Receive FIFO has 1 word, 2: Receive FIFO has 2 words, . . . 8: Receive FIFO has 8 words, Others: reserved.	0x0

3.5.18 SPI Interrupt Status

Short Name: SPI_INTR_STAT
Address: 0x68

Table 117. SPI Interrupt Status

Bits	Type	Name	Description	Default
7	R/WC	SPI_TXBF_UNRN_FG	**SPI Transmit Buffer Under-Run Flag** The bit determines whether the transmit data is ready in the transmit buffer to shift-out. Ignored as FIFO enabled. Write 1 to clear this flag and write 0 no effect. 0: Transmit data is ready before shift-out, 1: It has occurred that transmit data is not ready before shift-out.	0x0
6	R/WC	SPI_RXBF_OVRN_FG	**SPI Receive Buffer Over-Run Flag** The bit determines whether the data of the receive buffer is read by the host before next receive data shift-in done. Ignored as FIFO enabled. Write 1 to clear this flag and write 0 no effect. 0: Receive data is read before next receive data shift-in done. 1: It has occurred that receive data in receive buffer is not read before next receive data shift-in done.	0x0
5	R/WC	SPI_TXFF_UNRN_FG	**SPI Transmit FIFO Under-run Flag** The bit determines whether the transmit data of the transmit FIFO is empty to shift-out. Only for FIFO enabled. Write 1 to clear this flag and write 0 no effect.	0x0

			0: TX FIFO under-run has not occurred 1: TX FIFO under-run has occurred	
4	R/WC	SPI_RXFF_OVRN_FG	**SPI Receive FIFO Over-Run Flag** The bit determines whether the data of the receive FIFO is full before next receive data shift-in done. Only for FIFO enabled. Write 1 to clear this flag and write 0 no effect. 0: RX FIFO over-run has not occurred 1: RX FIFO over-run has occurred	0x0
3	RO	SPI_TXBUF_FG	**SPI Transmit Buffer Status Flag** The bit is cleared after SW writes to SPI-bus Transmit Buffer. Ignored as FIFO enabled. 0: Transmit Output buffer is not empty, no interrupt, 1: Transmit Output buffer is empty, interrupt is asserted if enabled.	0x1
2	RO	SPI_RXBUF_FG	**SPI Receive Buffer Status Flag** The bit is cleared after SW reads from SPI-bus Receive Buffer. Ignored as FIFO enabled. 0: Receive Input buffer is not full, no interrupt, 1: Receive Input buffer is full, interrupt is asserted if enabled.	0x0
1	RO	SPI_TXFF_FG	**SPI TX FIFO Under Threshold Flag** It is cleared when the data is loaded into TX FIFO and not under threshold. Only for FIFO enabled. 0: TX FIFO under threshold has not occurred, 1: TX FIFO under threshold has occurred, interrupt is asserted if enabled.	0x1
0	RO	SPI_RXFF_FG	**SPI RX FIFO Over Threshold Flag** It is cleared when the data is gotten from RX FIFO and not over threshold. Only for FIFO enabled. 0: RX FIFO over threshold has not occurred, 1: RX FIFO over threshold has occurred or RX FIFO contains some data not over threshold but time-out, interrupt is asserted if enabled.	0x0

3.5.19 SPI Interrupt Enable

Short Name: SPI_INTR_ENA
Address: 0x6C

Table 118. SPI Interrupt Enable

Bits	Type	Name	Description	Default
7	RW	SPI_TXBFERR_INTEN	**SPI Transmit Buffer Under-Run Interrupt Enable** The bit determines whether SPI-Bus Transmit Buffer Under-Run Interrupt is enabled. Ignored as FIFO enabled. 0: disable,	0x0

			1: enable.	
6	RW	SPI_RXBFERR_INTEN	**SPI Receive Buffer Over-Run Interrupt Enable** The bit determines whether SPI-Bus Receive Buffer Over-Run Interrupt is enabled. Ignored as FIFO enabled. 0: disable, 1: enable.	0x0
5	RW	SPI_TXFFERR_INTEN	**SPI Transmit FIFO Under-run Interrupt Enable** The bit determines whether SPI-Bus Transmit FIFO Under-Run Interrupt is enabled. Only for FIFO enabled. 0: disable, 1: enable.	0x0
4	RW	SPI_RXFFERR_INTEN	**SPI RX FIFO Over-Run Interrupt Enable** The bit determines whether SPI-Bus Receive FIFO Over-Run Interrupt is enabled. Only for FIFO enabled. 0: disable, 1: enable.	0x0
3	RW	SPI_TXBF_INTEN	**SPI Transmit Buffer Interrupt Enable** The bit determines whether SPI-Bus Transmit Buffer Interrupt is enabled. Ignored as FIFO enabled. 0: disable, 1: enable.	0x0
2	RW	SPI_RXBF_INTEN	**SPI Receive Buffer Interrupt Enable** The bit determines whether SPI-Bus Receive Buffer Interrupt is enabled. Ignored as FIFO enabled. 0: disable, 1: enable.	0x0
1	RW	SPI_TXFF_INTEN	**SPI Transmit FIFO Interrupt Enable** The bit determines whether SPI-Bus Transmit FIFO Interrupt is enabled. Only for FIFO enabled. 0: disable, 1: enable.	0x0
0	RW	SPI_RXFF_INTEN	**SPI Receive FIFO Interrupt Enable** The bit determines whether SPI-Bus Receive FIFO Interrupt is enabled. Only for FIFO enabled. 0: disable, 1: enable.	0x0

3.5.20 PCM Configuration 0

Short Name: PCM_CFG_0
Address: 0x80

Table 119. PCM Configuration 0

Bits	Type	Name	Description	Default
31	RW	PCM_EN	**PCM Enable** This bit enables or disables the PCM operation. 0: PCM operation is disabled, 1: PCM operation is enabled.	0x0

24	RW	PCM_SWAP_EN	**PCM Data Swap Enable** This bit determines if the byte-order of write data from APB to PCM_TXDAT0/TXDAT1 registers or read data from PCM_RXDAT0/RXDATA1 registers to APB are swapped. PWDATA is APB write bus and PRDATA is APB read bus. 0: no swap, PCM_TXDAT = PWDATA[31:0], PRDATA = PCM_RXDATA[31:0]. 1: data swapped, PCM_TXDAT = { PWDATA[7:0], PWDATA[15:8], PWDATA[23:16], PWDATA[31:24] }, PRDATA = { PCM_RXDAT[7:0], PCM_RXDAT[15:8], PCM_RXDAT[23:16], PCM_RXDAT[31:24] }.	0x0
14	RW	GCI_EN	**GCI / IDL Operation Mode Control** This bit selects the GCI/IDL operation mode. 0: IDL operation (data bit rate = PCMCLK), 1: GCI operation (data bit rate = PCMCLK/2).	0x0
13	RW	PCM_MASTER_EN	**PCM Master/Slave Mode Operation Control** This bit selects master/slave mode of operation. 0: slave mode (PCMCLK and PCMSYNC are inputs), 1: master mode (PCMCLK and PCMSYNC are outputs).	0x0
12	RW	PCM_LBK	**PCM Loop Back Enable** Loop back mode allows module validation during device testing. This mode is valid only in master mode of the PCM. 0: disable, 1: enable.	0x0
2:0	RW	PCM_CLKRATE	**PCM Master Clock Rate Control** These bits determine the bit clock rate as the PCM is the network master. There are 6 clock rates that can be selected. Source clock is 4.096MHz. 0: source clock / 1, 1: source clock / 2, 2: source clock / 4, 3: source clock / 8, 4: source clock / 16, 5: source clock / 32, 6, 7: reserved. Note: Based on fixing Frame Sync rate at 8kHz, the bit clock rate has some constraints at clock rate vs. channel number and width, and the constraints are showed in following table	0x0

Table 120. Bit clock rate vs. channel number and width.

Rate Setting	PCM is configured as IDL	PCM is configured as GCI

source clock/1	to support up-to 4 channels with maximum data width 16-bit each;	to support up-to 4 channels with maximum data width 16-bit each;
source clock/2	to support up-to 4 channels with maximum data width 16-bit each;	to support up-to 4 channels with maximum data width 16-bit each;
source clock/4	to support up-to 4 channels with maximum data width 16-bit each;	to support up-to 2 channels with maximum data width 16-bit each;
source clock/8	to support up-to 2 channels with maximum data width 16-bit each;	to support only 1 channel with maximum data width 16-bit;
source clock/16	to support 1 channel with maximum data width 16-bit;	to support only 1 channel with data width 8-bit;
source clock/32	to support 1 channel with data width 8-bit;	n/a

3.5.21 PCM Configuration 1

Short Name: PCM_CFG_1
Address: 0x84

Table 121. PCM Configuration 1

Bits	Type	Name	Description	Default
15	RW	PCM_SYNCWIDE	**PCM Frame SYNC Width** This bit defines the width of the PCMSYNC generated by master. It is ignored as the PCM is the slave or the master with GCI operation enable. 0: PCMSYNC is short , and its width is one PCMCLK pulse, 1: PCMSYNC is long, and its width covers two PCMCLK pulses.	0x0

3.5.22 PCM Channel 0 – 3 Configuration

Short Name: PCM_CH0_CFG – PCM_CH3_CFG
Address: 0x88 – Channel 0 Configuration
Address: 0x8C – Channel 1 Configuration
Address: 0x90 – Channel 2 Configuration
Address: 0x94 – Channel 3 Configuration

Table 122. PCM Channel 0 – 3 Configuration

Bits	Type	Name	Description	Default
23	RW	CPM_CH_TIMESLOT_EN	**PCM Channel Time Slot Enable** The bit enables/disables the time slot of the channel. 0: the channel is disabled, 1: the channel is enabled.	0x0
22	RW	PCM_CH_DATAWIDTH	**PCM Width of Data** The bit defines the data width of the channel.	0x0

			0: 8 bit 1: 16 bit	
14:8	RW	PCM_CH_TX_DLY	**PCM Channel Transmit Offset** The channel's transmit offset from PCMSYNC in data bits.	0x00
6:0	RW	PCM_CH_RX_DLY	**PCM Channel Receive Offset** The channel's receive offset from PCMSYNC in data bits.	0x00

3.5.23 PCM Transmit Data[31:0]

Short Name: PCM_TX_DATA_L
Address: 0x98

Table 123. PCM Transmit Data[31:0]

Bits	Type	Name	Description	Default
31:0	WO	PCM_TXDAT0	**PCM Transmit Data[31:0]** This register contains the data to be transmitted through the Transmit paths. The MSB of each channel data is the first bit to be transmitted and the LSB is the last bit. When this register is written, it will clear PCM_TXBUF_EMPTY_FG flag. The data of each channel are placed from right to left. For example, when each channel is in 8-bit data mode and 4 channels are active, the transmit data are placed at the transmit register. Ch0 is placed at bit TXDAT0 [7:0], Ch1 is placed at bit TXDAT0 [15:8], Ch2 is placed at bit TXDAT0 [23:16], Ch3 is placed at bit TXDAT0 [31:24].	0x00000000

3.5.24 PCM Transmit Data[63:32]

Short Name: PCM_TX_DATA_H
Address: 0x9C

Table 124. PCM Transmit Data[63:32]

Bits	Type	Name	Description	Default
31:0	WO	PCM_TXDAT1	**PCM Transmit Data[63:32]** This register contains the data to be transmitted through the Transmit paths. The MSB of each channel data is the first bit to be transmitted and the LSB is the last bit. For example, when each channel is in 16-bit data mode, and 4 channels are active, the transmit data are placed at these two transmit registers. We must write PCM_TXDAT1 first then write PCM_TXDAT0. Ch0 is placed at TXDAT0[15:0]. Ch1 is placed at TXDAT0[31:16], Ch2 is placed at TXDAT1[15:0],	0x00000000

			Ch3 is placed at TXDAT1[31:16].	

3.5.25 PCM Receive Data[31:0]

Short Name: PCM_RX_DATA_L
Address: 0xA0

Table 125. PCM Receive Data[31:0]

Bits	Type	Name	Description	Default
31:0	RO	PCM_RXDAT0	**PCM Receive Data[31:0]** This register contains the data received through the Receive paths. The MSB of each channel data is the first bit to be received and the LSB is the last bit. When this register is read, it will clear PCM_RXBUF_FULL_FG flag. The data of each channel are placed from right to left. For example, when each channel is in 8-bit data mode and 4 channels are active, the receive data are placed at this register. Ch0 is placed at bit RXDAT0 [7:0], Ch1 is placed at bit RXDAT0 [15:8], Ch2 is placed at bit RXDAT0 [23:16], Ch3 is placed at bit RXDAT0 [31:24].	0x00000000

3.5.26 PCM Receive Data[63:32]

Short Name: PCM_RX_DATA_H
Address: 0xA4

Table 126. PCM Receive Data[63:32]

Bits	Type	Name	Description	Default
31:0	RO	PCM_RXDAT1	**PCM Receive Data[63:32]** This register contains the data received through the Receive paths. The MSB of each channel data is the first bit to be received and the LSB is the last bit. For example, when each channel is in 16-bit data mode and 4 channels are active, the receive data are placed at these two receive registers. We must read PCM_RXDAT1 first then read PCM_RXDAT0. Ch0 is placed at RXDAT0[15:0]. Ch1 is placed at RXDAT0[31:16], Ch2 is placed at RXDAT1[15:0], Ch3 is placed at RXDAT1[31:16].	0x00000000

3.5.27 PCM Interrupt Status

Short Name: PCM_INTR_STAT
Address: 0xA8

Table 127. PCM Interrupt Status

Bits	Type	Name	Description	Default
3	R/WC	PCM_TXBF_UNRN_FG	**PCM Transmit Buffer Under-Run Flag** The bit determines whether the transmit data is ready in the transmit buffer to shift-out. Write 1 to clear. 0: Transmit data is ready before shift-out, 1: It has occurred that transmit data is not ready before shift-out.	0x0
2	R/WC	PCM_RXBF_OVRN_FG	**PCM Receive Buffer Over-run Flag** The bit determines whether the data of the receive buffer is read by the host before next receive data shift-in done. Write 1 to clear. 0: Receive data is read before next receive data shift-in done. 1: It has occurred that receive data in receive buffer is not read before next receive data shift-in done.	0x0
1	RO	PCM_TXBUF_EMPTY_FG	**PCM Transmit Buffer Empty Flag** The bit is cleared after data written to TXDAT0 Register. 0: Transmit Output buffer is not empty, no interrupt, 1: Transmit Output buffer is empty, interrupt is asserted if enabled.	0x1
0	RO	PCM_RXBUF_FULL_FG	**PCM Receive Buffer Full Flag** The bit is cleared after data read from RXDAT0 Register. 0: Receive Input buffer is not full, no interrupt, 1: Receive Input buffer is full, interrupt is asserted if enabled.	0x0

3.5.28 PCM Interrupt Enable

Short Name: PCM_INTR_ENA
Address: 0xAC

Table 128. PCM Interrupt Enable

Bits	Type	Name	Description	Default
3	RW	PCM_ TXBF_UNRN_IEN	**PCM Transmit Buffer Under-Run Interrupt Enable** The bit determines whether PCM Transmit Buffer Under-run Interrupt is enabled. 0: disable, 1: enable.	0x0
2	RW	PCM_ RXBF_OVRN_IEN	**PCM Receive Buffer Over-Run Interrupt Enable** The bit determines whether PCM Receive Buffer Over-run Interrupt is enabled. 0: disable, 1: enable.	0x0

1	RW	PCM_ TXBUF_EMPTY _IEN	**PCM Transmit Buffer Empty Interrupt Enable** The bit determines whether PCM Transmit Buffer Empty Interrupt is enabled. 0: disable, 1: enable.	0x0
0	RW	PCM_ RXBUF_FULL _IEN	**PCM Receive Buffer Full Interrupt Enable** The bit determines whether PCM Receive Buffer Full Interrupt is enabled. 0: disable, 1: enable.	0x0

3.5.29 I2S Configuration

Short Name: I2S_CFG
Address: 0xC0

Table 129. I2S Configuration

Bits	Type	Name	Description	Default
31	RW	I2S_EN	**I2S Enable** This bit enables and disables the I2S interface operation. 0: I2S operation is disabled, all IO in high impedance, 1: I2S operation is enabled.	0x0
30	RW	I2S_MASTER_EN	**I2S Master Enable** The bit determines whether the I2S is a master or slave. 0: I2S configured as slave 1: I2S configured as master	0x0
29	RW	I2SSD_DIR	**I2SSD pin direction** 0: output 1: input	0x0
27:26	RW	I2S_TRANSFER_TIMING	**I2S Transfer Timing** The bits determine whether the I2S transfer timing is a I2S, Right justified(RJF), or Left justified(LJF). 0: I2S, 2: Right justified(RJF), 3: Left justified(LJF).	0x0
25	RW	I2S_CLK_MODE	**I2S Clock Transfer Mode** The bit determines whether the serial clock is a continuous or 256-S transfer mode. 0: I2S's clock configured as continuous transfer mode, 1: I2S's clock configured as 256-S transfer mode.	0x0
24	RW	I2S_SWAP_EN	**I2S Data Swap Enable** This bit determines if the byte-order of write data from APB to I2S_TXDAT_R/TXDAT_L registers or read data from I2S_RXDAT_R/RXDATA_L registers to APB are swapped. PWDATA is APB write bus and PRDATA is APB read bus. 0: no swap,	0x0

			I2S_TXDAT = PWDATA[31:0], PRDATA = I2S_RXDATA[31:0]. 1: data swapped, I2S_TXDAT = { PWDATA[7:0], PWDATA[15:8], PWDATA[23:16], PWDATA[31:24] }, PRDATA = { I2S_RXDAT[7:0], I2S_RXDAT[15:8], I2S_RXDAT[23:16], I2S_RXDAT[31:24] }.	
15	RW	I2S_CLK_INV	**I2S Clock Phase Invert Enable** The bit configures if the phase of I2S clock is inverted. 0: disable, 1: enable.	0x0
12	RW	I2S_IN_SEL	**I2S Data Input Pin Selection** 0: I2SSD 1: GPIOA[3]/I2SDR	0x0
5:4	RW	I2S_SCLK_DIV	**I2S Source Clock Divide** These bits determine the bit transfer rate if I2S is the master. There are 4 data-transfer rates that can be selected. 0: undivided I2S clock / 1 1: undivided I2S clock / 2 2: undivided I2S clock / 4 3: undivided I2S clock / 8	0x0
0	RW	I2S_DATA_WIDTH	**I2S Serial Data Width** These bits determine the number of bits to be shifted in or out as a single data during one shift sequence. 0: 16 bits, 1: 32 bits,	0x0

3.5.30 I2S Right Transmit Data

Short Name: I2S_RIGHT_TX_DATA
Address: 0xC4

Table 130. I2S Right Transmit Data

Bits	Type	Name	Description	Default
31:0	WO	I2S_TXDAT_R	**I2S Right Transmit Data** This register contains the data to be transmitted through Transmit path of the right channel. The MSB of channel data is the first bit to be transmitted and the LSB is the last bit. Width-16: the bit[31:16] of data are valid, Width-32: the bit[31:0] of data are valid.	0x00000000

3.5.31 I2S Left Transmit Data

Short Name: I2S_LEFT_TX_DATA
Address: 0xC8

Table 131. I2S Left Transmit Data

Bits	Type	Name	Description	Default
31:0	WO	I2S_TXDAT_L	**I2S Left Transmit Data** This register contains the data to be transmitted through Transmit path of the left channel. The MSB of channel data is the first bit to be transmitted and the LSB is the last bit. Width-16: the bit[31:16] of data are valid, Width-32: the bit[31:0] of data are valid.	0x00000000

3.5.32 I2S Right Receive Data

Short Name: I2S_RIGHT_RX_DATA
Address: 0xCC

Table 132. I2S Right Receive Data

Bits	Type	Name	Description	Default
31:0	RO	I2S_RXDAT_R	**I2S Right Receive Data** This register contains the data received through the Receive path of the right channel. The MSB of channel data is the first bit to be received and the LSB is the last bit. Width-16: the bit[31:16] of data are valid, Width-32: the bit[31:0] of data are valid.	0x00000000

3.5.33 I2S Left Receive Data

Short Name: I2S_LEFT_RX_DATA
Address: 0xD0

Table 133. I2S Left Receive Data

Bits	Type	Name	Description	Default
31:0	RO	I2S_RXDAT_L	**I2S Left Receive Data** This register contains the data received through the Receive path of the left channel. The MSB of channel data is the first bit to be received and the LSB is the last bit. Width-16: the bit[31:16] of data are valid, Width-32: the bit[31:0] of data are valid.	0x00000000

3.5.34 I2S Interrupt Status

Short Name: I2S_INTR_STAT
Address: 0xD4

Table 134. I2S Interrupt Status

Bits	Type	Name	Description	Default
7	R/WC	I2S_TXBF_UNRN_L	**I2S Transmit Buffer Under-Run flag for Left Channel** The bit determines whether the transmit data for left channel is ready in the transmit buffer to shift-out. Write 1 to clear. 0: Transmit data is ready before shift-out, no interrupt, 1: It has occurred that transmit data is not ready before shift-out, interrupt is asserted if enabled.	0x0
6	R/WC	I2S_RXBF_OVRN_L	**I2S Receive Buffer Over-Run flag for Left Channel** The bit determines whether the data of the receive buffer for left channel is read by the host before next receive data shift-in done. Write 1 to clear. 0: Receive data is read before next receive data shift-in done, no interrupt, 1: It has occurred that receive data in receive buffer is not read before next receive data shift-in done, interrupt is asserted if enabled.	0x0
5	R/WC	I2S_TXBF_UNRN_R	**I2S Transmit Buffer Under-Run flag for Right Channel** The bit determines whether the transmit data for right channel is ready in the transmit buffer to shift-out. Write 1 to clear. 0: Transmit data is ready before shift-out, no interrupt, 1: It has occurred that transmit data is not ready before shift-out, interrupt is asserted if enabled.	0x0
4	R/WC	I2S_RXBF_OVRN_R	**I2S Receive Buffer Over-Run flag for Right Channel** The bit determines whether the data of the receive buffer for right channel is read by the host before next receive data shift-in done. Write 1 to clear. 0: Receive data is read before next receive data shift-in done, no interrupt, 1: It has occurred that receive data in receive buffer is not read before next receive data shift-in done, interrupt is asserted if enabled.	0x0
3	RO	I2S_TXBUF_FG_L	**I2S Transmit Buffer Empty Flag for Left Channel** The bit is cleared after data written to I2S_TXDAT_L Register. 0: Transmit Output buffer is not empty, no interrupt, 1: Transmit Output buffer is empty, interrupt is asserted if enabled.	0x1
2	RO	I2S_RXBUF_FG_L	**I2S Receive Buffer Full Flag for Left Channel**	0x0

			The bit is cleared after data read from I2S_RXDAT_L Register. 0: Receive Input buffer is not full, no interrupt, 1: Receive Input buffer is full, interrupt is asserted if enabled.	
1	RO	I2S_TXBUF_FG_R	**I2S Transmit Buffer Empty Flag for Right Channel** The bit is cleared after data written to I2S_TXDAT_R Register. 0: Transmit Output buffer is not empty, no interrupt, 1: Transmit Output buffer is empty, interrupt is asserted if enabled.	0x1
0	RO	I2S_RXBUF_FG_R	**I2S Receive Buffer Full Flag for Right Channel** The bit is cleared after data read from I2S_RXDAT_R Register. 0: Receive Input buffer is not full, no interrupt, 1: Receive Input buffer is full, interrupt is asserted if enabled.	0x0

3.5.35 I2S Interrupt Enable

Short Name: I2S_INTR_ENA
Address: 0xD8

Table 135. I2S Interrupt Enable

Bits	Type	Name	Description	Default
7	RW	I2S_TXERR_INTEN_L	**I2S Transmit Buffer Under-Run Interrupt Enable for Left Channel** The bit determines whether I2S Transmit Buffer Under-Run Interrupt for left channel is enabled. 0: disable, 1: enable.	0x0
6	RW	I2S_RXERR_INTEN_L	**I2S Receive Buffer Over-Run Interrupt Enable for Left Channel** The bit determines whether I2S Receive Buffer Over-Run Interrupt for left channel is enabled. 0: disable, 1: enable.	0x0
5	RW	I2S_ TXERR_INTEN _R	**I2S Transmit Buffer Under-Run Interrupt Enable for Right Channel** The bit determines whether I2S Transmit Buffer Under-Run Interrupt for left channel is enabled. 0: disable, 1: enable.	0x0
4	RW	I2S_ RXERR_INTEN _R	**I2S Receive Buffer Over-Run Interrupt Enable for Right Channel** The bit determines whether I2S Receive Buffer Over-Run Interrupt for left channel is enabled. 0: disable, 1: enable.	0x0

3	RW	I2S_TXBF_INTEN_L	**I2S Transmit Buffer Empty Interrupt Enable for Left Channel** The bit determines whether I2S Transmit Buffer Empty Interrupt for left channel is enabled. 0: disable, 1: enable.	0x0
2	RW	I2S_RXBF_INTEN_L	**I2S Receive Buffer Full Interrupt Enable for Left Channel** The bit determines whether I2S Receive Buffer Full Interrupt for left channel is enabled. 0: disable, 1: enable.	0x0
1	RW	I2S_TXBF_INTEN_R	**I2S Transmit Buffer Empty Interrupt Enable for Right Channel** The bit determines whether I2S Transmit Buffer Empty Interrupt for right channel is enabled. 0: disable, 1: enable.	0x0
0	RW	I2S_RXBF_INTEN_R	**I2S Receive Buffer Full Interrupt Enable for Right Channel** The bit determines whether I2S Receive Buffer Full Interrupt for right channel is enabled. 0: disable, 1: enable.	0x0

3.6 DDR/SDR SDRAM Controller

3.6.1 DRAM General Configuration

3.6.1.1 Memory Interface Configure Register

Short Name: MEM_CFG
Address: 0x00

Table 136. Memory Interface Configure Register

Bits	Type	Name	Description	Default
25	RW	DDREn	**Configure Memory Bus as DDR/SDR DRAM.** 0: SDR 1: DDR	0x0
5:4	RW	MemWidth	**DDR/SDR DRAM Device Bus Width.** 0: x4 1: x8 2: x16 3: Reserved	0x2
2:0	RW	MemSize	DDR/SDRAM SDRAM device capacity. 0: 128Mb 1: 256Mb 2: 512Mb 3: 1Gb (DDR ONLY) 4: 16Mb (SDR ONLY) 5: 64Mb (SDR ONLY)	0x1

			Others: Undefined, Reserved.	

3.6.1.2 DRAM Parameter Configuration

Short Name: DRAM_CFG
Address: 0x04

Table 137. DRAM Parameter Configuration

Bits	Type	Name	Description	Default
6:4	RW	CASL	**CAS Latency** 2: 2 (DDR/SDR) 3: 3 (DDR/SDR) 6: 2.5 (DDR ONLY) Others: Undefined, Reserved.	0x2
1	RW	DS	**Output Drive Strength (DDR ONLY)** 0: Normal 1: Reduced (Device Dependent)	0x0
0	RW	DLL	**DLL Enable/Disable (DDR ONLY)** 0: Enable 1: Disable (For Low Speed Issue)	0x0

3.6.2 DRAM Initialization

3.6.2.1 Power ON Initial Control Register

Short Name: PWR_ON_INIT_CTRL
Address: 0x08

Table 138. Power ON Initial Control Register

Bits	Type	Name	Description	Default
1	RO	InitCmp	DDR/SDR DRAM Initial Complete. Active high.	0x0
0	W	InitStrt	Set High to Initialize DDR/SDR DRAM	0x0

3.6.3 DRAM Timing Parameter

3.6.3.1 DRAM Timing Parameter Register 0

Short Name: DRAM_TIMING0
Address: 0x10

Table 139. DRAM Timing Parameter Register 0

Bits	Type	Name	Description	Default
31:28	RW	t_{RCD}	Active to Read/Write Delay	0x3
27:24	RW	t_{RRD}	Active bank A to Active bank B Latency	0x2
23:20	RW	t_{WR}	Write recovery time	0x2
19:16	RW	t_{RC}	Active to Active /Auto-refresh Command Period	0x9
15:12	RW	t_{MRD}	LOAD MODE REGISTER command period	0x2
11:8	RW	t_{RFC}	AUTO-REFRESH command period	0x9

7:4	RW	t_{RAS}	Active to Precharge command period	0x6
3:0	RW	t_{RP}	Precharge command period	0x2

3.6.3.2 DRAM Timing Parameter Register 1

Short Name: DRAM_TIMING1
Address: 0x14

Table 140. DRAM Timing Parameter Register 1

Bits	Type	Name	Description	Default
31:28	RW	t_{WTR}	Internal Write to Read Command Delay (**DDR ONLY)**	0x1
27:24	RW	t_{DAL}	Data-in to Active Command Time (**SDR ONLY)**	0x4
23:20	R	Reserved.	Read as Zero. Write don't Care.	0x0
19:16	RW	t_{SREX}	Self-refresh Exit Time	0xF
15:8	RW	t_{XSR} (SDR) t_{XSNR} (DDR)	Exit Self-refresh to Active Command Delay Exit Self-refresh to Non-Read Command Delay	0x09 (SDR) 0x10 (DDR)
7:0	RW	t_{XSRD}	Exit Self-refresh to Read Command Delay	0xC8

3.6.3.3 DRAM Timing Parameter Register 2

Short Name: DRAM_TIMING2
Address: 0x18

Table 141. DRAM Timing Parameter Register 2

Bits	Type	Name	Description	Default
27:16	RW	t_{BREF}	**Maximum Burst Refresh Count** 0x000 = 1 Refresh Command in a burst 0x001 = 2 Refresh Commands in a burst .... 0xFFF = 4096 Refresh Commands in a burst Note: For DDR, the value must be less or equal to 0x007.	0x007
11:0	RW	t_{REFI}	**Average Periodic Refresh Interval Time** 0x000 = auto refresh disable 0x001 ~ 0xFFF = 16(N+1) HCLK ticks between refresh cycle	0x03C

Note: Default value is based on HCLK = 125MHz, DDR SDRAM = 256/512/1GMb.

3.6.4 Pre-Read Function

3.6.4.1 PreRead TimeOut Disable Register

Short Name: PREREAD_TIMEOUT_DIS
Address: 0x1C

Table 142. PreRead TimeOut Disable Register

Bits	Type	Name	Description	Default

7:0	RW	Pre_Read_TimeOut_Dis	**AHB Channel PreRead TimeOut Disable. (HIGH active)** When the corresponding bits were set, the Pre-Read time out mechanism were disabled. Pre_Read_ TimeOut_Dis [0]: Channel 0 (CPU) Pre_Read_ TimeOut_Dis [1]: Channel 1 (GDMA) Pre_Read_ TimeOut_Dis [2]: Channel 2 (IDE) Pre_Read_ TimeOut_Dis [3]: Channel 3 (PCI) Pre_Read_ TimeOut_Dis [4]: Channel 4 (USB Host 2.0) Pre_Read_ TimeOut_Dis [5]: Channel 5 (USB Host 1.1) Pre_Read_ TimeOut_Dis [6]: Channel 6 (From NIC) Pre_Read_ TimeOut_Dis [7]: Channel 7 (To NIC)	0x00

3.6.4.2 PreRead Enable

Short Name: PREREAD_ENA
Address: 0x20

Table 143. PreRead Enable

Bits	Type	Name	Description	Default
7:0	RW	Pre_Read_Enable	**AHB Channel PreRead Enable. (HIGH active)** PreRead is an enhanced ability to read data, suitable for bulk data transfer. Pre_Read_Enable[0]: Channel 0 Pre_Read_Enable[1]: Channel 1 Pre_Read_Enable[2]: Channel 2 Pre_Read_Enable[3]: Channel 3 Pre_Read_Enable[4]: Channel 4 Pre_Read_Enable[5]: Channel 5 Pre_Read_Enable[6]: Channel 6 Pre_Read_Enable[7]: Channel 7	0x00

3.6.4.3 PreRead TimeOut Register 0

Short Name: PREREAD_TIMEOUT0
Address: 0x24

Table 144. PreRead TimeOut Register 0

Bits	Type	Name	Description	Default
29:24	RW	Pre_Read_Time_Out3	**Channel 3 Pre-Read time out.** It sets pre-read data time out cycle time. After time out, the pre-read data at Channel 3 is invalid.	0x1F
21:16	RW	Pre_Read_Time_Out2	**Channel 2 Pre-Read time out.**	0x1F

			It sets pre-read data time out cycle time. After time out, the pre-read data at Channel 2 is invalid.	
13:8	RW	Pre_Read_Time_Out1	**Channel 1 Pre-Read time out.** It sets pre-read data time out cycle time. After time out, the pre-read data at Channel 1 is invalid.	0x1F
5:0	RW	Pre_Read_Time_Out0	**Channel 0 Pre-Read time out.** It sets pre-read data time out cycle time. After time out, the pre-read data at Channel 0 is invalid.	0x1F

3.6.4.4 PreRead TimeOut Register 1

Short Name: PREREAD_TIMEOUT1
Address: 0x28

Table 145. PreRead TimeOut Register 1

Bits	Type	Name	Description	Default
29:24	RW	Pre_Read_Time_Out7	**Channel 7 Pre-Read time out.** It sets pre-read data time out cycle time. After time out, the pre-read data at Channel 7 is invalid.	0x1F
21:16	RW	Pre_Read_Time_Out6	**Channel 6 Pre-Read time out.** It sets pre-read data time out cycle time. After time out, the pre-read data at Channel 6 is invalid.	0x1F
13:8	RW	Pre_Read_Time_Out5	**Channel 5 Pre-Read time out.** It sets pre-read data time out cycle time. After time out, the pre-read data at Channel 5 is invalid.	0x1F
5:0	RW	Pre_Read_Time_Out4	**Channel 4 Pre-Read time out.** It sets pre-read data time out cycle time. After time out, the pre-read data at Channel 4 is invalid.	0x1F

3.6.5 DDR Skew Adjust

3.6.5.1 DDQ Output Delay Control Register (DDR Only)

Short Name: DDQ_OUT_DLY_CTRL
Address: 0x30

Table 146. DDQ Output Delay Control Register (DDR Only)

Bits	Type	Name	Description	Default
6:4	RW	Ddq_Out_Dly1	**Byte-Lane-1 DDQ[7:0] output delay control** It is for adjusting delay of Byte-Lane-1 transmitting data to meet DDR write timing spec. Typically, each step is with 0.4ns variance.	0x4
2:0	RW	Ddq_Out_Dly0	**Byte-Lane-0 DDQ[7:0] output delay control** It is for adjusting delay of Byte-Lane-0	0x4

			transmitting data to meet DDR write timing spec. Typically, each step is with 0.4ns variance.	

3.6.5.2 DQS Input Delay Control Register (DDR Only)

Short Name: DQS_IN_DLY_CTRL
Address: 0x34

Table 147. DQS Input Delay Control Register (DDR Only)

Bits	Type	Name	Description	Default
22:20	RW	Dqs_In_Dly1_n	**Byte lane 1 DQS[1] input falling edge sample delay** It is for adjusting delay of DQS[1] input falling edge to sample DDQ[15:8]. Typically, each step is with 0.4ns variance. Note: Suggest re-configureing the value to 0x3 when DDC clock > 100MHz for better timing.	0x4
18:16	RW	Dqs_In_Dly0_n	**Byte lane 0 DQS[0] input falling edge sample delay** It is for adjusting delay of DQS[0] input falling edge to sample DDQ[7:0]. Typically, each step is with 0.4ns variance. Note: Suggest re-configuring the value to 0x3 when DDC clock > 100MHz for better timing.	0x4
6:4	RW	Dqs_In_Dly1	**Byte lane 0 DQS[1] input rising edge sample delay** It is for adjusting delay of DQS[0] input rising edge to sample DDQ[15:8]. Typically, each step is with 0.4ns variance. Note: Suggest re-configuring the value to 0x3 when DDC clock > 100MHz for better timing.	0x4
2:0	RW	Dqs_In_Dly0	**Byte lane 0 DQS[0] input rising edge sample delay** It is for adjusting delay of DQS[0] input rising edge to sample DDQ[7:0]. Typically, each step is with 0.4ns variance. Note: Suggest re-configuring the value to 0x3 when DDC clock > 100MHz for better timing.	0x4

3.6.6 Power Management

3.6.6.1 Pad Power Down Register

Short Name: PAD_PWR_DOWN
Address: 0x3C

Table 148. Pad Power Down Register

Bits	Type	Name	Description	Default
0	RW	PadPwrDwn	**Pad Receiver Power Down** 0: active 1: DDQ[15:0] and DQS(DDR only) pad	0x0

			receive power down Note, for saving more power, user can power down these DDR pad receiver when entering into DDR self reflresh state.	

3.7 Static Memory Controller

3.7.1 Memory Bank 0 Configuration Register

Short Name: MEM_BNK0_CFG
Address: 0x00

Table 149. Memory Bank 0 Configuration Register

Bits	Type	Name	Description	Default
4	RW	BNK_WPROT	**Bank Write Protected.** If BNK_WPROT is set to 1, the memory bank can't be written. Any write to protected bank will cause ERROR response on AHB. If BNK_WPROT is 0, the corresponding bank can be read or written.	0x1
3	RW	BURST_EN	**Burst Access Enable** When burst enable, AHB burst transaction will be transferred to external parallel interface, and assert Chip-enable till the burst transaction completion. When Read, only address bus toggle, output-enable signal keep asserted till transaction completion. When write, address bus and write-enable signal toggle for each data access till transaction completion.	0x0
1	RW	BNK_EN	**Bank Enables** 1: Enable 0: Disable	0x1
0	RW	BNK_MBW	**Memory Bus Width.** This register indicates the bus width of external memory bus. 0: Memory data width is 8. 1: Memory data width is 16.	0x0

3.7.2 Memory Bank 0 Timing Parameter Register

Short Name: MEM_BNK0_TIMING
Address: 0x04

Table 150. Memory Bank 0 Timing Parameter Register

Bits	Type	Name	Description	Default
31:28	RW	AST	**Address Setup Time.** This register specifies setup time, in turns of system clock cycle, from address assertion to Read/Write-enable.	0xF

			0x0: 1 cycle 0x1: 2 cycles ... 0xF: 16 cycles	
27:24	RW	AHT	**Address Hold Time.** This register specifies the hold time, in turns of system clock cycle, from Read/Write-disable to address change. 0x0: 1 cycle 0x1: 2 cycles ... 0xF: 16 cycles	0xF
23	RW		**Reserved**	0x1
22:16	RW	RAT	**Read Access Time.** The AT specifies the low pulse width of Read-enable for accessing external SRAM device in turns of system clock cycle. 0x00: 1 cycles 0x01: 2 cycles ... 0x7F: 128 cycles	0x7F
15	RW		**Reserved**	0x1
14:8	RW	WAT	**Write Access Time.** The AT specifies the low pulse width of Write-enable for accessing external SRAM device in turns of system clock cycle. 0x00: 1 cycles 0x01: 2 cycles ... 0x7F: 128 cycles	0x7F
7	RW		**Reserved**	0x1
6:0	RW	TRNA	**Turn-around Time.** The TRNA specifies the latency needed to re-drive data bus in turns of system clock cycle. 0x00: 1 cycles 0x01: 2 cycles ... 0x7F: 128 cycles	0x7F

3.7.3 Memory Bank 1 - 3 Configuration Register

Short Name: MEM_BNK1_CFG, MEM_BNK2_CFG, MEM_BNK3_CFG
Address: 0x08 (Bank 1)
Address: 0x10 (Bank 2)
Address: 0x18 (Bank 3)

Table 151. Memory Bank 1 - 3 Configuration Register

Bits	Type	Name	Description	Default
4	RW	BNK_WPROT	**Bank Write Protected.** If BNK_WPROT is set to 1, the memory bank can't be written. Any write to protected bank will cause ERROR response on AHB. If BNK_WPROT is 0, the corresponding bank can be read or	0x0

			written.	
3	RW	BURST_EN	**Burst Access Enable** When burst enable, AHB burst transaction will be transferred to external parallel interface, and assert Chip-enable till the burst transaction completion. When Read, only address bus toggle, output-enable signal keep asserted till transaction completion. When write, address bus and write-enable signal toggle for each data access till transaction completion.	0x0
2	RW	WAIT_EN	**Wait hand shake Enable** 0: wait hand shake disable 1: wait hand shake enable When the Bank disable or Bank enable but Wait hand shake disable, then the related SWAITn pin is functioned as a GPIO.	0x0
1	RW	BNK_EN	**Bank Enables** 1: Enable 0: Disable	0x0
0	RW	BNK_MBW	**Memory Bus Width.** This register indicates the bus width of external memory bus. 0: Memory data width is 8. 1: Memory data width is 16.	0x0

3.7.4 Memory Bank 1 - 3 Timing Parameter Register

Short Name: MEM_BNK1_TIMING, MEM_BNK2_TIMING, MEM_BNK3_TIMING
Address: 0x0C (Bank 1)
Address: 0x14 (Bank 2)
Address: 0x1C (Bank 3)

Table 152. Memory Bank 1 - 3 Timing Parameter Register

Bits	Type	Name	Description	Default
31:28	RW	AST	**Address Setup Time.** This register specifies setup time, in turns of system clock cycle, from address assertion to Read/Write-enable. 0x0: 1 cycle 0x1: 2 cycles ... 0xF: 16 cycles	0xF
27:24	RW	AHT	**Address Hold Time.** This register specifies the hold time, in turns of system clock cycle, from Read/Write-disable to address change. 0x0: 1 cycle 0x1: 2 cycles ... 0xF: 16 cycles	0xF
23	RW		**Reserved**	0x1
22:16	RW	RAT	**Read Access Time.** The AT specifies the low pulse width of	0x7F

			Read-enable for accessing external SRAM device in turns of system clock cycle. 0x00: 1 cycles 0x01: 2 cycles ... 0x7F: 128 cycles	
15	RW		**Reserved**	0x1
14:8	RW	WAT	**Write Access Time.** The AT specifies the low pulse width of Write-enable for accessing external SRAM device in turns of system clock cycle. 0x00: 1 cycles 0x01: 2 cycles ... 0x7F: 128 cycles	0x7F
7	RW		**Reserved**	0x1
6:0	RW	TRNA	**Turn-around Time.** The TRNA specifies the latency needed to re-drive data bus in turns of system clock cycle. 0x00: 1 cycles 0x01: 2 cycles ... 0x7F: 128 cycles	0x7F

3.8 IDE Controller

3.8.1 IDE PIO mode Control Register

Address: 0x00
Default: 0x0000-00F4

Table 153. IDE PIO mode Control Register

Bits	Type	Name	Description	Default
7:6	RW	PIOCLKSEL	**PIO Clock Divider Select** The setting of these bits determines the PIO mode clock (PIO_CLK) divided by 1/2/3/4 from AHB clock (125M/100M/87.5M) 0 = divide by 1 (125M/100M/87.5M) 1 = divide by 2 (62.5M/50M/43.75M) 2 = divide by 3 (41.7M/33.3M/29.1M) 3 = divide by 4 (31.25M/25M/21.875M)	0x3
5:4	RW	DMACLKSEL	**DMA Clock Divider Select** The setting of these bits determines the DMA and Ultra DMA mode clock (DMA_CLK) divided by 1/2/3/4 from AHB clock (125M/100M/87.5M) 0 = divide by 1 (125M/100M/87.5M) 1 = divide by 2 (62.5M/50M/43.75M) 2 = divide by 3 (41.7M/33.3M/29.1M)	0x3

			3 = divide by 4 (31.25M/25M/21.875M) Note: At Ultra DMA mode, user should set CLKDIVSEL = 00.	
2	RW	IDEP	**IDE Decode Enable** This bit enable/disable the Primary decode. 0 = disable 1 = Enables the controller to decode the Command and control blocks. Default value is 0x1, always enable for IDE PIO decode	0x1
1	RW	IE1	**Drive 1 IORDY Sample Point Enable** 0 = Disable IORDY sampling for this drive. 1 = Enable IORDY sampling for this drive.	0x0
0	RW	IE0	**Drive 0 IORDY Sample Point Enable** 0 = Disable IORDY sampling for this drive. 1 = Enable IORDY sampling for this drive.	0x0

3.8.2 IDE Drive0 PIO Timing Configuration Register

Address: 0x04
Default: 0x0000-03AA

Table 154. IDE Drive0 PIO Timing Configuration Register

Bits	Type	Name	Description	Default
15:12	RW	Ta_0	**IDE PIO Drive 0 IORDY Sample Point** The setting of these bits determines the number of PIO clock cycles (Ta_0+1) between DIOR_n/DIOW_n assertion and the first IORDY sample point.	0x0
11:8	RW	T2i_0	**IDE Drive 0 Recovery Time** The setting of these bits determines the number of PIO clock cycles (T2i_0+1) between the last IORDY sample point and the DIOR_n/DIOW_n strobe of the next cycle.	0x3
7:4	RW	T1_0	**IDE Drive0 Address Valid to DIORn/DIOWn setup time. The unit is One-Clock of the PIO clock.** Setup Time = T1_0 + 1	0xA
3:0	RW	T2_0	**IDE Drive0 DIORn/DIOWn asserted pulse width. The unit is One-Clock period of the PIO clock** Asserted Pulse Width = T2_0 + 1	0xA

3.8.3 IDE Drive1 PIO Timing Configuration Register

Address: 0x08
Default: 0x0000-03AA

Table 155. IDE Drive1 PIO Timing Configuration Register

Bits	Type	Name	Description	Default
15:12	RW	Ta_1	**IDE PIO Drive 1 IORDY Sample Point** The setting of these bits determines the number of PIO clock cycles (Ta_1+1) between DIOR_n/DIOW_n assertion and the first IORDY sample point.	0x0
11:8	RW	T2i_1	**IDE Drive1 Recovery Time** The setting of these bits determines the number of PIO clock cycles (T2i_1+1) between the last IORDY sample point and the DIOR_n/DIOW_n strobe of the next cycle.	0x3
7:4	RW	T1_1	**IDE Drive1 Address Valid to DIORn/DIOWn setup time. The unit is One-Clock of the PIO clock.** Setup Time = T1_1 + 1	0xA
3:0	RW	T2_1	**IDE Drive1 DIORn/DIOWn asserted pulse width. The unit is One-Clock period of the PIO clock** Asserted Pulse Width = T2_1 + 1	0xA

3.8.4 IDE Drive0 DMA Timing Configuration Register

Address: 0x0C
Default: 0x0000-03AA

Table 156. IDE Drive0 DMA Timing Configuration Register

Bits	Type	Name	Description	Default
11:8	RW	Tm_0	**IDE Drive0 Address Valid to DIORn/DIOWn setup time. The unit is One-Clock of the DMA clock.** Setup Time = Tm_0 + 1	0x3
7:4	RW	Td_0	**IDE Drive0 DIORn/DIOWn asserted pulse width. The unit is One-Clock period of the DMA clock** Asserted Pulse Width = Td_0 +1	0xA
3:0	RW	Tk_0	**IDE Drive0 DIORn/DIOWn negated pulse width. The unit is One-Clock period of the DMA clock** Negated Pulse Width = Tk_0 + 1 Note: DMA cycle time = Asserted pulse width + Negated Pulse Width. Note: Tk_0 MUST be larger than 3 for DMA mode 1. And Tk_0 MUST be larger than 2 for DMA mode 2.	0xA

3.8.5 IDE Drive1 DMA Timing Configuration Register

Address: 0x10
Default: 0x0000-03AA

Table 157. IDE Drive1 DMA Timing Configuration Register

Bits	Type	Name	Description	Default
11:8	RW	Tm_1	**IDE Drive1 Address Valid to DIORn/DIOWn setup time. The unit is One-Clock of the DMA clock.** Setup Time = Tm_1 + 1	0x3
7:4	R/W	Td_1	**IDE Drive1 DIORn/DIOWn asserted pulse width. The unit is One-Clock period of the DMA clock** Asserted Pulse Width = Td_1 +1	0xA
3:0	R/W	Tk_1	**IDE Drive1 DIORn/DIOWn negated pulse width. The unit is One-Clock period of the DMA clock** Negated Pulse Width = Tk_1 + 1 Note: DMA cycle time = Asserted pulse width + Negated Pulse Width. Note: Tk_1 MUST be larger than 3 for DMA mode 1. And Tk_1 MUST be larger than 2 for DMA mode 2.	0xA

3.8.6 IDE Ultra DMA mode Timing Configuration Register

Address: 0x14
Default: 0x0000-0000

Table 158. IDE Ultra DMA mode Timing Configuration Register

Bits	Type	Name	Description	Default
31:28	R/W	PCT1	**Primary Drive 1 Cycle Time** For Ultra DMA mode, this field determines the minimum write strobe cycle time (CT). Software will set suitable values based on different AHB clock frequency. Note: Unit is one DMA clock cycle. **For UltraDMA (DMA clock is 125MHz):** Mode0: CT = 4'd15 Mode1: CT = 4'd10 Mode2: CT = 4'd8 Mode3: CT = 4'd6 Mode4: CT = 4'd4 Mode5: CT = 4'd3 **For UltraDMA (DMA clock is 100MHz):** Mode0: CT = 4'd12 Mode1: CT = 4'd8 Mode2: CT = 4'd6 Mode3: CT = 4'd5 Mode4: CT = 4'd3 Mode5: CT = 4'd2 **For UltraDMA (DMA clock is 87.5MHz):** Mode0: CT = 4'd11	0x0

			Mode1: CT = 4'd8 Mode2: CT = 4'd6 Mode3: CT = 4'd5 Mode4: CT = 4'd3 Mode5: CT = 4'd2	
27:24	R/W	PCT0	**Primary Drive 0 Cycle Time** For Ultra DMA mode, this field determines the minimum write strobe cycle time (CT). Software will set suitable values based on different AHB clock frequency. Note: Unit is one DMA clock cycle. **For UltraDMA (DMA clock is 125MHz):** Mode0: CT = 4'd15 Mode1: CT = 4'd10 Mode2: CT = 4'd8 Mode3: CT = 4'd6 Mode4: CT = 4'd4 Mode5: CT = 4'd3 **For UltraDMA (DMA clock is 100MHz):** Mode0: CT = 4'd12 Mode1: CT = 4'd8 Mode2: CT = 4'd6 Mode3: CT = 4'd5 Mode4: CT = 4'd3 Mode5: CT = 4'd2 **For UltraDMA (DMA clock is 87.5MHz):** Mode0: CT = 4'd11 Mode1: CT = 4'd8 Mode2: CT = 4'd6 Mode3: CT = 4'd5 Mode4: CT = 4'd3 Mode5: CT = 4'd2	0x0
23:20	R/W	ENV1	**Used by IDE Ultra DMA mode.** Drive1 Envelope time = ENV1+1 DMA clock cycles (from DMACK_ to STOP and HDMARDY_ during data in burst initiation and from DMACK to STOP during data out burst initiation)	0x0
19:16	R/W	ENV0	**Used by IDE Ultra DMA mode.** Drive0 Envelope time = ENV0+1 DMA clock cycles (from DMACK_ to STOP and HDMARDY_ during data in burst initiation and from DMACK to STOP during data out burst initiation)	0x0
15:12	R/W	MLI1	**Used by IDE Ultra DMA mode. Interlock time = MLI1+1 cycles.** The unit is one-Clock of the DMA clock	0x0
11:8	R/W	MLI0	**Used by IDE Ultra DMA mode. Interlock time = MLI0+1 cycles.** The unit is one-Clock of the DMA clock	0x0

3.8.7 IDE DMA and Ultra DMA mode Control Register

Address: 0x18
Default: 0x0000-0000

Table 159. IDE DMA and Ultra DMA mode Control Register

Bits	Type	Name	Description	Default
4	R/W	FAST_DMA_EN	**USB to IDE Fast Path DMA Enable** 0 = Normal Case – IDE to Memory 1 = Fast Path - IDE to USB Note: When FAST_DMA_EN = 1, USB device hardware hand shake signals are connected to IDE fast path DMA, otherwise, they are connected to HSDMA.	0x0
3	R/W	DMAEn1	**Drive 1 DMA mode Enable** 0 = DMA mode disable. 1 = DMA mode enable.	0x0
2	R/W	DMA En0	**Drive 0 DMA mode Enable** 0 = DMA mode disable. 1 = DMA mode enable.	0x0
1	R/W	UDMAEn1	**Drive 1 Ultra DMA Mode Enable** 0 = Disable 1 = Enable Ultra DMA mode for primary channel drive 1 Note: When UDMAEn1 is set to 1, the above DMAEn1 has no effect. That is, Drive 1 is in Ultra DMA mode.	0x0
0	R/W	UDMAEn0	**Drive 0 Ultra DMA Mode Enable** 0 = Disable 1 = Enable Ultra DMA mode for primary channel drive 0 Note: When UDMAEn0 is set to 1, the above DMAEn0 has no effect. That is, Drive 0 is in Ultra DMA mode.	0x0

3.8.8 IDE Status and Control Register

Address: 0x1C
Default: 0x0000-0080

Table 160. IDE Status and Control Register

Bits	Type	Name	Description	Default
6	R/W	PRDI_Mask	**PRD Interrupt Mask** 0 = PRD interrupt is enabled 1 = PRD interrupt is disabled	0x1
5	R/WC	Error	**Error** 0 = Software clears it by writing a 1 to it. 1 = This bit is set when the controller encounters an AHB error response when transferring data on AHB.	0x0
4	RO	ACT	**IDE Active** 0 = This bit is cleared by hardware	0x0

			when the last transfer for a region is performed, where EOT for that region is set in the region descriptor. It is also cleared by controller when the Start bit below is cleared. When this bit is read as 0, all data transferred from the drive during the previous IDE command is visible in system memory, unless the IDE command is aborted. 1 = Set by the host controller when the Start bit is set.	
3	R/W	RWC	**Read / Write Control** This bit set the direction of the IDE transfer. This bit must NOT be changed when the bus master function is active. 0 = IDE Host Controller Reads 1 = IDE Host Controller Writes	0x0
2	R/WC	PRD_intr	**PRD Interrupt** 0 = When this bit is cleared by software, the interrupt is cleared. 1 = Set when the host controller completes execution of a PRD.	0x0
1	R/WC	Dev_intr	**Device Interrupt** Software can use this bit to determine if an IDE device has asserted its interrupt pin (INTRQ). 0 = Software clears the bit by writing a 1 to it. If this bit is cleared while the interrupt is still active, this bit will remain clear until another assertion edge is detected on the interrupt pin. 1 = Set by the rising edge of the interrupt pin, regardless of whether or not the interrupt is masked. When this bit is read as 1, all data transferred from the drive is visible in system memory.	0x0
0	R/W	START	**Start / Stop IDE DMA Transfer** 0 = All state information is lost when this bit is cleared. Master mode operation cannot be stopped and then resumed. If this bit is reset while IDE operation is still active and the drive has not yet finished its data transfer, the IDE command is said to be aborted and data transferred from the drive may be discarded instead of being written to system memory. 1 = Enables IDE DMA operation of the controller. IDE DMA operation begins when this bit is detected changing from 0 to 1. The controller will transfer data between the IDE device and memory only when this bit is set. DMA operation can be halted by writing a 0 to this bit.	0x0

			NOTE: This bit is intended to be cleared by software after the data transfer is completed, as indicated by either the IDE Active bit being cleared or the Interrupt bit in this register being set, or both. Hardware does not clear this bit automatically.	

Note: The bit[2] (PRD_intr) will combine bit[1] (Dev_intr) and bit[4] (Active) to explain the IDE interrupt meaning shown at the following:

PRD_intr	Dev_intr	Active	Description
0	0	1	DMA transfer is in progress. No interrupt has been generated by the IDE device.
1	1	0	The IDE device generated an interrupt. The controller exhausted the Physical Region Descriptors. This is the normal completion case where the size of the physical memory regions was equal to the IDE device transfer size.
0	1	1	The IDE device generated an interrupt. The controller has not reached the end of the physical memory regions. This is a valid completion case where the size of the physical memory regions was larger than the IDE device transfer size.
1	0	0	This bit combination signals an error condition. If the Error bit in the status register is set, then the controller has some problem transferring data to/from memory. Specifics of the error have to be determined using bus-specific information. If the Error bit is not set, then the PRD's specified a smaller size than the IDE transfer size.

3.8.9 IDE DMA Descriptor Table Pointer Register

Address: 0x20
Default: 0x0000-0000

Table 161. IDE DMA Descriptor Table Pointer Register

Bits	Type	Name	Description	Default
31:2	RW	ADDR	**Address of Descriptor Table** The descriptor table must be DWord-aligned, and must not cross 64-KB boundary in memory.	0x00000000

3.8.10 IDE to USB Fast Path Access Window Register

Address: 0x24
Default: 0x0000-0000

Table 162. IDE to USB Fast Path Access Window Register

Bits	Type	Name	Description	Default
31:2	RW	UD_ADDR	**Address of USB Device Access Window** The address is DWord-aligned.	0x00000000

3.8.11 IDE to USB Fast Path DMA Burst Size Register

Address: 0x28
Default: 0x0000-0000

Table 163. IDE to USB Fast Path DMA Burst Size Register

Bits	Type	Name	Description	Default
19:16	RW	HHS_SIZE	**Hardware Hand Shake Size Selection.** It means how many transfers in a hardware hand shake "req/ack" period. Hardware Hand Shake Size = 2^HHS_SIZE, where allowed HHS_SIZE = 2~10 (it means size = 4, …, 1K), and 0,1, and 11~15 are reserved. Unit = 1 Double Word.	0x0
14:0	RW	FAST_BURST_SIZE	**Burst Size For Fast Path.** Unit = 1 DWord The maximum number of burst size is16K DWord 64K Bytes).	0x0000

Note: The following registers exist at devices physically and memory mapped at IDE memory space offset 0x20~0x3C (Command Block) and 0x40 (Control Block).

3.8.12 Data Register

Address: 0x1800_0020
Default: 0x0000-0000

Table 164. Data Register

Bits	Type	Name	Description	Default
15:0	RW	DataReg		

3.8.13 Error Register (Read)

Address: 0x1800_0024
Default: 0x0000-0000

Table 165. Error Register (Read)

Bits	Type	Name	Description	Default
7:0	RO	ErrorReg		

3.8.14 Feature Register (Write)

Address: 0x1800_0024
Default: 0x0000-0000

Table 166. Feature Register (Write)

Bits	Type	Name	Description	Default
7:0	WO	FeatureReg		

3.8.15 Sector Count Register

Address: 0x1800_0028
Default: 0x0000-0000

Table 167. Sector Count Register

Bits	Type	Name	Description	Default
7:0	RW	SectorCountReg		

3.8.16 LBA Low Register

Address: 0x1800_002C
Default: 0x0000-0000

Table 168. LBA Low Register

Bits	Type	Name	Description	Default
7:0	RW	LBAHReg		

3.8.17 LBA MID Register

Address: 0x1800_0030
Default: 0x0000-0000

Table 169. LBA MID Register

Bits	Type	Name	Description	Default
7:0	RW	LBAMReg		

3.8.18 LBA High Register

Address: 0x1800_0034
Default: 0x0000-0000

Table 170. LBA High Register

Bits	Type	Name	Description	Default
7:0	RW	LBALReg		

3.8.19 Device Register

Address: 0x1800_0038
Default: 0x0000-0000

Table 171. Device Register

Bits	Type	Name	Description	Default
7:0	RW	DeviceReg		

3.8.20 Command Register (Write)

Address: 0x1800_003C
Default: 0x0000-0000

Table 172. Command Register (Write)

Bits	Type	Name	Description	Default
7:0	WO	CommandReg		

3.8.21 Status Register (Read)

Address: 0x1800_003C
Default: 0x0000-0000

Table 173. Status Register (Read)

Bits	Type	Name	Description	Default
7:0	RO	StatusReg		

3.8.22 Device Control Register (Write

Address: 0x1800_0040
Default: 0x0000-0000

Table 174. Device Control Register (Write)

Bits	Type	Name	Description	Default
7:0	WO	DeviceCtrlReg		

3.8.23 Alternate Status Register (Read)

Address: 0x1800_0040
Default: 0x0000-0000

Table 175. Alternate Status Register (Read)

Bits	Type	Name	Description	Default
7:0	RO	AlternateStatusReg		

3.9 Miscellaneous

3.9.1 Memory Re-map Register

Address: 0x00

Table 176. Memory Re-map Register

Bits	Type	Name	Description	Default
0	RW	RemapEn	**Remap Enable.**	0x0

			After set, it can only be cleared by reset. 0: parallel Flash boot, alias 0x1000-0000 as 0x0000-0000; SPI serial flash boot, alias 0x3000-0000 as 0x0000-0000 1: alias 0x2000-0000 as 0x0000-0000	

3.9.2 Chip Configuration Register

Address: 0x04

Table 177. Chip Configuration Register

Bits	Type	Name	Description	Default
12	RO	Test_Mode_En	**Test Mode Enable** 0: The chip is in normal mode, and the below BYPASS and Test Mode information are mean less. 1: The chip is in test mode, and the below BYPASS and Test Mode information are as their definition.	Set by external pin TESTMODE
11	RO	BYPASS	By pass PLL 0: All of internal PLL are active 1: All of internal PLL are by passed.	
10:8	RO	Test_Mode	Test Mode 0~7 for chip test.	
4	RW	SPISerialFlashEn	**SPI Serial Flash access region enable** 0: Disable access SPI bank 0 (SPICS0n active) though 0x3000-0000 region. It can only be accessed from 0x7100-0000, SPI controller's registers. 1: Enable access SPI bank 0 (SPI serial Flash memory) through 0x3000-0000 region.	0x1
2	RO	ICESEL	**ICE Select.** 0: Faraday ICE. 1: ARM Multi-ICE.	Set by external configuration pin CLKOUT.
1	RO	endian	**Endian** 0: Little Endian 1: Big Endian	Set by external configuration pin SPIDT.
0	RO	SPIBoot	**Boot from SPI Serial Flesh Memory** 0: Boot from Parallel Flash Memory 1: Boot from SPI Serial Flash Memory Note: After boot completed and the above Remap bit is set, suggest clearing SPIBankEn to 0 to disable accessing SPI serial flash memory from 0x3000-0000, but accessing it from 0x7100-0000, SPI controller's registers.	Set by external configuration pin CKE.

3.9.3 PCI Control and Broken Mask Register

Address: 0x10

Table 178. PCI Control and Broken Mask Register

Bits	Type	Name	Description	Default
9	RW	run_66MHz	**66 MHz Capability** This bit indicates to PCI devices whether or not this PCI bridge is capable of running at 66MHz. 0: running at only 33MHz 1: can support 66MHz operation	0x1
8	RW	hs_fast	This bit can be set to 1 to improve the performance of AHB-to-PCI write accesses. Set it to 1 improves the latency of PCI master write accesses. This input should be set to 1 if the AHB clock runs faster than the PCI clock. It should be set to 0 if the CPU clock does not run faster than the PCI clock. If clock speeds are undetermined, hs_fast should be set to 0.	0x1
4:0	RW	Broken_Mask	**PCI Device Broken Interrupt Mask** Bit: 0: External PCI Device #0 1: External PCI Device #1 2: Reserved 3: Reserved 4: PCI Host Bridge	0x1F

3.9.4 PCI Broken Status Register

Address: 0x14

Table 179. PCI Broken Status Register

Bits	Type	Name	Description	Default
4:0	RWC	Broken_Status	**PCI Device Broken Status** PCI Device Broken means the device has gotten the bus grant but kept the bus idle for more than 15 PCI clock cycle. Each bit is write 1 clear. Bit: 0: External PCI Device #0 1: External PCI Device #1 2: Reserved 3: Reserved 4: PCI Host Bridge	0x00

3.9.5 PCI Device ID and Vendor ID Register

Address: 0x18

Table 180. PCI Device ID and Vendor ID Register

Bits	Type	Name	Description	Default
31:16	RW	Device_id	**PCI Host Bridge Device ID**	0x8131
15:0	RW	Vendor_id	**PCI Host Bridge Vendor ID**	0xEEEE

3.9.6 USB Host PHY Control and Test Register

Address: 0x1C

Table 181. USB Host PHY Control and Test Register

Bits	Type	Name	Description	Default
18	RW	TEST_CLK30_SYNC	**CLK30 Sync Control for test purpose**	9x0
13:0	RW	USB20PHY_CTRL	**USB 2.0 Host PHY Control** [1:0] : TX amplitude control: "00" : 420mV, "01" : 440mV, "10" : 470mV, "11" : 500 mV [2]: TX PRE_AMP control [4:3]: Impendence control, "00" : < 45Ω, "01": = 45Ω, "10": >45Ω, "11": >> 45Ω [5]: RX CHIRP level threshold, "0": 510mV, "1"; 550mV [6]: RX squelch level threshold, "0": 140mV, "1": 100mV [7]: USB1.1 current control [13:8]: RX receiver algorithm control	0x000C

3.9.7 GPIO_A Pin Enable Register

Short Name: GPIOA_EN
Address: 0x20

Table 182. GPIO_A Pin Enable Register

Bits	Type	Name	Description	Default
31	RW	SPICSn[3]	**SPICSn[3] Pin Enable** 0: enable GPIOA[31] pin function 1: enable SPICSn[3] pin function	0x0
30	RW	SPICSn[2]	**SPICSn[2] Pin Enable** 0: enable GPIOA[30] pin function 1: enable SPICSn[2] pin function	0x0
29	RW	SPICSn[1]	**SPICSn[1] Pin Enable** 0: enable GPIOA[29] pin function 1: enable SPICSn[1] pin function	0x0
28	RW	SPICSn[0]	**SPICSn[0] Pin Enable** 0: enable GPIOA[28] pin function 1: enable SPICSn[0] pin function	0x0
27	RW	SPICLK	**SPICLK Pin Enable** 0: enable GPIOA[27] pin function 1: enable SPICLK pin function	0x0

26	RW	SPIDR	**SPIDR Pin Enable** 0: enable GPIOA[26] pin function 1: enable SPIDR pin function	0x0
25	RW	WDTRSTn	**WDTRSTn Pin Enable** 0: enable GPIOA[25] pin function 1: enable WDTRSTn pin function	0x0
24	RW	LED2	**LED2 Pin Enable** 0: enable GPIOA[24] pin function 1: enable LED2 pin function	0x0
23	RW	LED1	**LED1 Pin Enable** 0: enable GPIOA[23] pin function 1: enable LED1 pin function	0x0
22	RW	LED0	**LED0 Pin Enable** 0: enable GPIOA[22] pin function 1: enable LED0 pin function	0x0
21	RW	PCMCLK	**PCMCLK Pin Enable** 0: enable GPIOA[21] pin function 1: enable PCMCLK pin function	0x0
20	RW	PCMFS	**PCMFS Pin Enable** 0: enable GPIOA[20] pin function 1: enable PCMFS pin function	0x0
19	RW	PCMDT	**PCMDT Pin Enable** 0: enable GPIOA[19] pin function 1: enable PCMDT pin function	0x0
18	RW	PCMDR	**PCMDR Pin Enable** 0: enable GPIOA[18] pin function 1: enable PCMDR pin function	0x0
17	RW	I2SCLK	**I2SCLK Pin Enable** 0: enable GPIOA[17] pin function 1: enable I2SCLK pin function	0x0
16	RW	I2SWS	**I2SWS Pin Enable** 0: enable GPIOA[16] pin function 1: enable I2SWS pin function	0x0
15	RW	I2SSD	**I2SSD Pin Enable** 0: enable GPIOA[15] pin function 1: enable I2SSD pin function	0x0
14	RW	SCL	**SCL Pin Enable** 0: enable GPIOA[14] pin function 1: enable SCL pin function	0x0
13	RW	SDA	**SDA Pin Enable** 0: enable GPIOA[13] pin function 1: enable SDA pin function	0x0
3	RW	UR_ACT1	**UR_ACT1 Pin Enable** 0: enable GPIOA[3]/I2SDR pin function 1: UR_ACT1 pin function	0x0
2	RW	UR_ACT0	**UR_ACT0 Pin Enable** 0: enable GPIOA[2] pin function 1: UR_ACT0 pin function	0x0
1	RW	EXT_INT30	**EXT_INT30 Pin Enable** 0: enable GPIOA[1] pin function 1: enable EXT_INT30 pin function	0x0
0	RW	EXT_INT29	**EXT_INT29 Pin Enable** 0: enable GPIOA[0] pin function 1: enable EXT_INT29 pin function	0x0

3.9.8 GPIO_B Pin Enable Register

Short Name: GPIOB_EN
Address: 0x24

Table 183. GPIO_B Pin Enable Register

Bits	Type	Name	Description	Default
31	RW	GNT1n	**GNT1n Pin Enable** 0: enable GPIOB[31] pin function 1: enable GNT1n pin function	0x0
30	RW	REQ1n	**REQ1n Pin Enable** 0: enable GPIOB[30] pin function 1: enable REQ1n pin function	0x0
29	RW	GNT0n	**GNT0n Pin Enable** 0: enable GPIOB[29] pin function 1: enable GNT0n pin function	0x0
28	RW	REQ0n	**REQ0n Pin Enable** 0: enable GPIOB[28] pin function 1: enable REQ0n pin function	0x0
27	RW	SERRn	**SERRn Pin Enable** 0: enable GPIOB[27] pin function 1: enable SERRn pin function	0x0
26	RW	PERRn	**PERRn Pin Enable** 0: enable GPIOB[26] pin function 1: enable PERRn pin function	0x0
25	RW	STOPn	**STOPn Pin Enable** 0: enable GPIOB[25] pin function 1: enable STOPn pin function	0x0
24	RW	DEVSELn	**DEVSELn Pin Enable** 0: enable GPIOB[24] pin function 1: enable DEVSELn pin function	0x0
23	RW	TRDYn	**TRDYn Pin Enable** 0: enable GPIOB[23] pin function 1: enable TRDYn pin function	0x0
22	RW	UR_RXD[1]	**UR_RXD[1] Pin Enable** 0: enable GPIOB[22] pin function 1: enable UR_RXD[1] pin function	0x0
21	RW	UR_TXD[1]	**UR_TXD[1] Pin Enable** 0: enable GPIOB[21] pin function 1: enable UR_TXD[1] pin function	0x0
20	RW	PWAITn	Reserved	0x0
19	RW	REGn	Reserved	0x0
18	RW	CE2n	Reserved	0x0
17	RW	CE1n	Reserved	0x0
16	RW	SWAIT3n	**SWAIT3n Pin Enable** 0: enable GPIOB[16] pin function 1: enable SWAIT3n pin function	0x0
15	RW	SWAIT2n	**SWAIT2n Pin Enable** 0: enable GPIOB[15] pin function 1: enable SWAIT2n pin function	0x0
14	RW	SWAIT1n	**SWAIT1n Pin Enable** 0: enable GPIOB[14] pin function	0x0

			1: enable SWAIT1n pin function	
13	RW	SCE3n	**SCE3n Pin Enable** 0: enable GPIOB[13] pin function 1: enable SCE3n pin function	0x0
12	RW	SCE2n	**SCE2n Pin Enable** 0: enable GPIOB[12] pin function 1: enable SCE2npin function	0x0
11	RW	SCE1n	**SCE1n Pin Enable** 0: enable GPIOB[11] pin function 1: enable SCE1n pin function	0x0
10	RW	DIOWn	**DIOWn Pin Enable** 0: enable GPIOB[10] pin function 1: enable DIOWn pin function	0x0
9	RW	DIORn	**DIORn Pin Enable** 0: enable GPIOB[9] pin function 1: enable DIORn pin function	0x0
8	RW	IDECS1n	**IDECS1n Pin Enable** 0: enable GPIOB[8] pin function 1: enable IDECS1n pin function	0x0
7	RW	IDECS0n	**IDECS0n Pin Enable** 0: enable GPIOB[7] pin function 1: enable IDECS0n pin function	0x0
6	RW	INTRQ	**INTRQ Pin Enable** 0: enable GPIOB[6] pin function 1: enable INTRQ pin function	0x0
5	RW	DMACKn	**DMACKn Pin Enable** 0: enable GPIOB[5] pin function 1: enable DMACKn pin function	0x0
4	RW	DMARQ	**DMARQ Pin Enable** 0: enable GPIOB[4] pin function 1: enable DMARQ pin function	0x0
3	RW	IORDY	**IORDY Pin Enable** 0: enable GPIOB[3] pin function 1: enable IORDY pin function	0x0
2	RW	COL	**COL Pin Enable** 0: enable GPIOB[2] pin function 1: enable COL pin function	0x0
1	RW	MDIO	**MDIO Pin Enable** 0: enable GPIOB[1] pin function 1: enable MDIO pin function	0x0
0	RW	MDC	**MDC Pin Enable** 0: enable GPIOB[0] pin function 1: enable MDC pin function	0x0

3.9.9 GPIOA Pull Up/Down Resistor Configuration Register

Address: 0x28

Table 184. GPIOA Pull Up/Down Resistor Configuration Register

Bits	Type	Name	Description	Default
25:24	RW	GPIOA12_Pull	**GPIOA[12] Pull Resistor Select** 0: none 1: 75k pull down	0x2

			2: 75k pull up 3: 75k keeper	
23:22	RW	GPIOA11_Pull	**GPIOA[11] Pull Resistor Select** 0: none 1: 75k pull down 2: 75k pull up 3: 75k keeper	0x2
21:20	RW	GPIOA10_Pull	**GPIOA[10] Pull Resistor Select** 0: none 1: 75k pull down 2: 75k pull up 3: 75k keeper	0x2
19:18	RW	GPIOA9_Pull	**GPIOA[9] Pull Resistor Select** 0: none 1: 75k pull down 2: 75k pull up 3: 75k keeper	0x2
17:16	RW	GPIOA8_Pull	**GPIOA[8] Pull Resistor Select** 0: none 1: 75k pull down 2: 75k pull up 3: 75k keeper	0x2
15:14	RW	GPIOA7_Pull	**GPIOA[7] Pull Resistor Select** 0: none 1: 75k pull down 2: 75k pull up 3: 75k keeper	0x1
13:12	RW	GPIOA6_Pull	**GPIOA[6] Pull Resistor Select** 0: none 1: 75k pull down 2: 75k pull up 3: 75k keeper	0x1
11:10	RW	GPIOA5_Pull	**GPIOA[5] Pull Resistor Select** 0: none 1: 75k pull down 2: 75k pull up 3: 75k keeper	0x1
9:8	RW	GPIOA4_Pull	**GPIOA[4] Pull Resistor Select** 0: none 1: 75k pull down 2: 75k pull up 3: 75k keeper	0x1
7:6	RW	GPIOA3_Pull	**GPIOA[3] Pull Resistor Select** 0: none 1: 75k pull down 2: 75k pull up 3: 75k keeper	0x1
5:4	RW	GPIOA2_Pull	**GPIOA[2] Pull Resistor Select** 0: none 1: 75k pull down 2: 75k pull up 3: 75k keeper	0x1
3:2	RW	GPIOA1_Pull	**GPIOA[1] Pull Resistor Select** 0: none 1: 75k pull down	0x1

			2: 75k pull up 3: 75k keeper	
1:0	RW	GPIOA0_Pull	**GPIOA[0] Pull Resistor Select** 0: none 1: 75k pull down 2: 75k pull up 3: 75k keeper	0x1

3.9.10 GPIOA Drive Strength Configuration Register

Address: 0x2C

Table 185. GPIOA Drive Strength Configuration Register

Bits	Type	Name	Description	Default
12	RW	GPIOA12_Drive	**GPIOA[12] Drive Strength Select** 0: 4mA 1: 8mA	0x0
11	RW	GPIOA11_Drive	**GPIOA[11] Drive Strength Select** 0: 4mA 1: 8mA	0x0
10	RW	GPIOA10_Drive	**GPIOA[10] Drive Strength Select** 0: 4mA 1: 8mA	0x0
9	RW	GPIOA9_Drive	**GPIOA[9] Drive Strength Select** 0: 4mA 1: 8mA	0x0
8	RW	GPIOA8_Drive	**GPIOA[8] Drive Strength Select** 0: 4mA 1: 8mA	0x0
7	RW	GPIOA7_Drive	**GPIOA[7] Drive Strength Select** 0: 4mA 1: 8mA	0x0
6	RW	GPIOA6_Drive	**GPIOA[6] Drive Strength Select** 0: 4mA 1: 8mA	0x0
5	RW	GPIOA5_Drive	**GPIOA[5] Drive Strength Select** 0: 4mA 1: 8mA	0x0
4	RW	GPIOA4_Drive	**GPIOA[4] Drive Strength Select** 0: 4mA 1: 8mA	0x0
3	RW	GPIOA3_Drive	**GPIOA[3] Drive Strength Select** 0: 4mA 1: 8mA	0x0
2	RW	GPIOA2_Drive	**GPIOA[2] Drive Strength Select** 0: 4mA 1: 8mA	0x0
1	RW	GPIOA1_Drive	**GPIOA[1] Drive Strength Select** 0: 4mA 1: 8mA	0x0
0	RW	GPIOA0_Drive	**GPIOA[0] Drive Strength Select** 0: 4mA 1: 8mA	0x0

3.9.11 Fast Ethernet PHY LED Configuration Register

Address: 0x30

Table 186. Fast Ethernet PHY LED Configuration Register

Bits	Type	Name	Description	Default
1:0	RW	led_mode_pin	**LED Mode Selection** **Note:** Changing this field will not change LED mode immediately. After toggling Fast Ethernet PHY software reset register bit (assert, then de-assert FE_PHY_SWRn of Software Reset Control register in Clock and Power Management block), then the LED mode setting will take action. **0:** LED0: Link/Activity LED on: link up off: link down flash: activity LED1: Speed LED on: 100M off: l0M LED2: Duplex/Collision LED on: full off: half flash: collision **1:** LED0: Link/Activity LED on: link up off: link down flash: activity LED1: Speed LED on: 100M off: l0M LED2: Duplex LED on: full off: half **2:** LED0: Link/Activity LED if in 100M mode on: link up off: link down flash: activity LED1: Link/Activity LED if in 10M mode on: link up off: link down flash: activity LED2: Duplex/Collision LED on: full off: half flash: collision **3:** LED0: Link/Activity LED	0x2

			on: link up off: link down flash: activity LED1: Speed LED on: 100M off: I0M LED2: Collision LED flash: collision	

3.9.12 HSDMA Control and Status Register

Address: 0x40

Table 187. HSDMA Control and Status Register

Bits	Type	Name	Description	Default
12	RO	M0_Error	Master 0 AHB Error Response When Master 0 AHB transaction has an Error response, he bit will be asserted, and the DMA engine will stop and assert HSDMA interrupt. It waits for software to clear the following DMACEN bit to clear the Error status. Then software can re-issue a new DMA command.	0x0
11:8	RO	LLP_CNT	**LLP Counter** When the following DMACEN is set to 1 (while the DMA transfer is enabled), the LLP_CNT is reset to zero. For each descriptor transaction finished, the LLP_CNT increases by one. Note that when the last transaction is finished, the LLP_CNT also increases by one. When DMACEN bit is cleared to 0 manually, the chain transfer is stopped and then the LLP_CNT keep original value.	0x0
3	RW	HHS_MODE	**Mode of Operation.** 0: Normal Mode (default). 1: Hardware Handshake Mode.	0x0
2	RW	M0_AD_CTL	**DMA Master 0 Address Control.** 0: Incremental Address 1: Fixed Address Note: DMA Master 1 is always in Incremental Address mode.	0x0
1			Reserved	0x0
0	RW	DMACEN	**HSDMA Controller Enable** When this bit is set to 1, DMA will start to move data, and will auto-clear the DMACEN bit when the DMA is completed, and generate an edge triggered interrupt. When CPU clears the bit from 1 to 0 manually, the DMA will be terminated without an interrupt. 0: Disable 1: Enable	0x0

3.9.13 HSDMA Master 0 Address Register

Short Name: M0Addr
Address: 0x50

Table 188. HSDMA Master 0 Address Register

Bits	Type	Name	Description	Default
31:2	RW	M0Addr	**DMA Master 0 Address** It is DW alignment.	0x00000000

3.9.14 HSDMA Master 1 Address Register

Short Name: M1Addr
Address: 0x54

Table 189. HSDMA Master 1 Address Register

Bits	Type	Name	Description	Default
31:0	RW	M1Addr	**DMA Master 1 Address** It is DW alignment.	0x00000000

3.9.15 HSDMA Linked List Descriptor Pointer

Short Name: HSDMA_LLP
Address: 0x58

Table 190. HSDMA Linked List Descriptor Pointer

Bits	Type	Name	Description	Default
31:2	RW	LLPAddr[31:2]	**Linked List Descriptor Pointer Address** It is DW alignment. Note: When LLPAddr = 0, means this channel's Linked List function is disabled. Note: When LLPAddr is not 0, then DMA engine will move data according to registers' setting. When completed, it will read the 1st descriptor, pointed by LLPAddr, and so on. Note: The last one descriptor is indicated with its LLPAddr = 0.	0x00000000

3.9.16 HSDMA Transfer Size Register

Short Name: HSDMA_TOT_SIZE
Address: 0x5C

Table 191. HSDMA Transfer Size Register

Bits	Type	Name	Description	Default
29	RW	Data_Direction	**Direction of Data Movement** 0: DMA Master 0 to DMA Master 1	0x0

			1: DMA Master 1 to DMA Master 0	
28	RW	TC_MASK	**Terminal Count Interrupt Mask** 0: Pass Interrupt 1: Mask Interrupt Note: When TC_MASK of related Link List Descriptor is 1, then the TC Interrupt will be suppressed when the data movement, pointed by the descriptor, is completed.	0x1
27:24	RW	HHS_SIZE	**Hardware Hand Shake Size Selection.** It means how many transfers in a hardware hand shake "req/ack" period. Hardware Hand Shake Size = 2^HHS_SIZE, where allowed HHS_SIZE = 0~10 (it means size = 1, 2, 4, …, 1K), and 11~15 are reserved. Note: For USB Device data transfer, the HHS_SIZE is re-commanded to be equal to USB end point maximum packet size. And it can be changed per descriptor by the related end-point's MAX_PKT_SIZE	0x0
15:0	RW	TOT_SIZE	**Total Transfer Size**	0x0000

3.10 Clock and Power Management

3.10.1 Clock gate control register 0 for AHB and APB devices

Address: 0x00

Table 192. Clock gate control register 0 for AHB and APB devices

Bits	Type	Name	Description	Default
30	RW	PCI_CLK_EN	**PCICLK of PCI Host Bridge Enable.** 1: Enable clock. 0: Disable clock.	0x0
28	RW	HCLK_PCI_EN	**HCLK of PCI Host Bridge Enable.** 1: Enable clock. 0: Disable clock.	0x0
24	RW	HCLK_USBH_EN	**HCLK of USB Host Controller Enable.** 1: Enable clock. 0: Disable clock. Note: Only when bit 2 of e-Fuse is one, the USB Host HCLK can be enabled; otherwise, USB Host clock will be always disabled.	0x0
23	RW	MAC_CLK_EN	**MAC Clock Enable.** 1: Enable clock. 0: Disable clock.	0x0
22	RW	MDC_CLK_EN	**MDC Clock Enable.** 1: Enable clock. 0: Disable clock.	0x0
21	RW	PCLK_NIC_EN	**PCLK of NIC Enable.** 1: Enable clock. 0: Disable clock.	0x0

20	RW	HCLK_NIC_EN	**HCLK of NIC and NIC Clock Enable.** 1: Enable clock. 0: Disable clock.	0x0
16	RW	HCLK_GDMA_EN	**HCLK of Generic DMA Enable.** 1: Enable clock. 0: Disable clock.	0x0
12	RW	HCLK_VIC_EN	**HCLK of Vector Interrupt Controller Enable.** 1: Enable clock. 0: Disable clock.	0x0
9	RW	PCLK_IDE_EN	**PCLK of IDE Controller Enable.** 1: Enable clock. 0: Disable clock. Note: Only when bit 4 of e-Fuse is one, the IDE Host Controller clock can be enabled; otherwise, it will be always disabled.	0x0
8	RW	HCLK_IDE_EN	**HCLK of IDE Controller Enable.** 1: Enable clock. 0: Disable clock. Note: Only when bit 4 of e-Fuse is one, the IDE Host Controller clock can be enabled; otherwise, it will be always disabled.	0x0
5	RW	PCLK_SMC_EN	**PCLK of Static Memory Controller Enable.** 1: Enable clock. 0: Disable clock.	0x1
4	RW	HCLK_SMC_EN	**HCLK of Static Memory Controller Enable.** 1: Enable clock. 0: Disable clock.	0x1
1	RW	PCLK_SDMC_EN	**PCLK of DRAM Controller Enable.** 1: Enable clock. 0: Disable clock.	0x1
0	RW	HCLK_SDMC_EN	**HCLK of DRAM Controller Enable.** This bit will be auto-set to 1 when the AHB clock is gated off. 1: Enable clock. 0: Disable clock. When this bit is set to 0, Power Management will wait for CPU entering Idle mode or Sleep mode, and then signal DRAM controller to issue self-refresh command. When this command is issued, the DRAM controller will signal Power Management to gate off DRAM controller's AHB clock. After wake up signal asserted, Power Management will turn on DRAM controller's AHB clock and signal DRAM controller to leave self-refresh mode. And then wake up CPU.	0x1

3.10.2 Clock gate control register 1 for AHB and APB devices

Address: 0x04

Table 193. Clock gate control register 1 for AHB and APB devices

Bits	Type	Name	Description	Default

29	RW	HCLK_HSDMA_EN	**HCLK of HSDMA Enable.** 1: Enable clock. 0: Disable clock.	0x0
28	RW	HCLK_USBD_EN	**HCLK of USB Device Enable.** 1: Enable clock. 0: Disable clock. Note: Only when bit 3 of e-Fuse is one, the USB Device HCLK can be enabled; otherwise, USB Device clock will be always disabled.	0x0
25	RW	PCLK_GPIO_EN	**PCLK of GPIO Enable.** 1: Enable clock. 0: Disable clock.	0x0
23	RW	PCLK_RTC_EN	**PCLK of Real Time Clock Enable.** 1: Enable clock. 0: Disable clock.	0x0
21	RW	PCLK_WDT_EN	**PCLK of Watch Dog Timer Enable.** 1: Enable clock. 0: Disable clock.	0x1
17	RW	PCLK_Timer_EN	**PCLK of Timer Enable.** 1: Enable clock. 0: Disable clock.	0x0
13	RW	PCLK_UART1_EN	**PCLK of UART1 Enable.** 1: Enable clock. 0: Disable clock.	0x0
12	RW	PCLK_UART0_EN	**PCLK of UART0 Enable.** 1: Enable clock. 0: Disable clock.	0x0
10	RW	I2S_CLK_EN	**I2S 44.1KHz Clock Enable.** 1: Enable clock. 0: Disable clock.	0x0
6	RW	PCM_CLK_EN	**PCM Master Clock Enable.** 1: Enable clock. 0: Disable clock.	0x0
1	RW	PCLK_P2S_EN	**PCLK of SPI/PCM/I2S/TWI Controller Enable.** 1: Enable clock. 0: Disable clock.	0x1
0	RW	HCLK_SPI_EN	**HCLK of SPI Flash Boot Controller Enable.** 1: Enable clock. 0: Disable clock.	0x1

3.10.3 Software reset control

Address: 0x08

Table 194. Software reset control

Bits	Type	Name	Description	Default
16	RW	HSDMA_SWRn	**HSDMA Software Reset.** (LOW active) It is needed to program it low, then high to generate a reset low pulse.	0x0
15	RW	FE_PHY_SWRn	**Internal Fast Ethernet PHY Software Reset.** (LOW active)	0x0

			It is needed to program it low, then high to generate a reset low pulse.	
14	RW	USBD_SWRn	**USB Device Software Reset.** (LOW active) It is needed to program it low, then high to generate a reset low pulse.	0x0
13	RW	GPIO_SWRn	**GPIO Software Reset.** (LOW active) It is needed to program it low, then high to generate a reset low pulse.	0x0
12	RW	WDT_SWRn	**Watch Dog Timer Software Reset.** (LOW active) It is needed to program it low, then high to generate a reset low pulse.	0x1
11	RW	Timer_SWRn	**Timer Software Reset.** (LOW active) It is needed to program it low, then high to generate a reset low pulse.	0x0
10	RW	UART1_SWRn	**UART1 Software Reset.** (LOW active) It is needed to program it low, then high to generate a reset low pulse.	0x0
9	RW	UART0_SWRn	**UART0 Software Reset.** (LOW active) It is needed to program it low, then high to generate a reset low pulse.	0x0
8	RW	P2S_SWRn	**SPI/PCM/I2S/TWI Software Reset.** (LOW active) It is needed to program it low, then high to generate a reset low pulse.	0x1
7	RW	PCI_SWRn	**PCI Host Bridge and PCI device Software Reset.** (LOW active) It is needed to program it low, then high to generate a reset low pulse.	0x0
6	RW	USBH_SWRn	**USB Host Controller Software Reset.** (LOW active) It is needed to program it low, then high to generate a reset low pulse.	0x0
5	RW	NIC_SWRn	**NIC Software Reset. (**LOW active) It is needed to program it low, then high to generate a reset low pulse.	0x0
4	RW	GDMA_SWRn	**Generic DMA Software Reset**. (LOW active) It is needed to program it low, then high to generate a reset low pulse.	0x0
3	RW	VIC_SWRn	**Vector Interrupt Controller Software Reset**. (LOW active) It is needed to program it low, then high to generate a reset low pulse.	0x0
2	RW	IDE_SWRn	**IDE Controller Software Reset.** (LOW active) It is needed to program it low, then high to generate a reset low pulse.	0x0
1	RW	SMC_SWRn	**Static Memory Controller Software Reset.** (LOW active) It is needed to program it low, then high to generate a reset low pulse.	0x1
0	RW	Global_SWRn	**Global Software Reset.** (LOW active) When the bit is programmed to low, the whole system is reset. It will be auto cleared to high after 2 APB clocks.	0x1

3.10.4 System clock control register

Address: 0x0C

Table 195. System clock control register

Bits	Type	Name	Description	Default
21:20	RW	CLKOUT_DIV	**CLKOUT Clock Divider. The output clock of pin CLKOUT is after the divider.** Dividend: The following CLKOUT_Sel selected clock 0: divided by 1. 1: divided by 2. 2: divided by 3. 4: divided by 4.	0x0
19:16	RW	CLKOUT_Sel	**Pin CLKOUT clock source select** 0: System Xtal Clock 25MHz 1: Reference Clock 32.678KHz 2: PCLK (default 43.75MHz) 3: 125MHz (for NIC) 4: 66.666MHz (for PCI) 5: 120MHz (for USB, UART) 6: 48MHz (from USB Host PHY) 7: 30MHz (from USB Device PHY) 8: 125MHz (from FEPHY) 9: 73.728MHz (for UART/PCM/I2S) 10: 8.192MHz (for PCM) 11: I2S Clock (Default: 11.2896MHz, can be configured by the following I2S_Sel bits) 12: 32.256MHz (for I2S 44.1K Audio) 13: 12MHz (for USB Host and Device PHY) 14~15: 0 Hz (silence) Note: CLKOUT_Sel = 2, 3, 4, 5, 6, 7, 11 should be tested at PLL test mode for IC production test.	0xF
15:14	RW	MDC_DIV	**MDC Clock Divider.** Dividend: 2.5MHz clock 0: divided by 1. 1: divided by 2. 2: divided by 4. 3: divided by 8.	0x2
13:12	RW	I2S_Sel	**I2S Clock Select.** 0: 8.192MHz (for 32K audio) 1: 11.2896MHz (for 44.1K audio) 2: 12.288MHz (for 48K audio) 3: reserved	0x1
11:10	RW	PCI_DIV	**PCI Clock Divider.** Dividend: 66.66MHz clock 0: divided by 1. 1: divided by 2. 2: divided by 3. 3: divided by 4.	0x1
9:8	RW	PCLK_DIV	**PCLK Clock Divider.** Dividend: HCLK.	0x1

			1: divided by 2. Note, due to APB bridge must keep APB_CLK = AHB_CLK/2, only default value is allowed.	
7	RW	NICCLK_Sel	**NIC Clock Select.** It is used to select a NIC clock for 10/100/1000Mbps speed. 0 & Giga: HCLK (must > 62.5MHz) 0 & 100: 25MHz 0 & 10: 2.5MHz 1 & Giga: 62.5MHz 1 & 100: 12.5Mhz 1 & 10: 1.25MHz Note: If it is possible to set HCLK to be lower than 62.5MHz, then the bit must be set to 1 to make NIC work normally.	0x0
6	RW	RTC_Sel	**RTC Clock Source Select** 0: external 32.768KHz reference clock 1: external 25MHz reference clock Recommend to set it to 1.	0x0
5:4	RW	HCLK_DIV	**HCLK Clock Divider.** Dividend: FCLK. 0: divided by 1. 1: divided by 2. 2: divided by 3. 3: divided by 4.	0x1
3:2	RW	FCLK_DIV	**CPU Clock Divider.** Dividend: PLL output clock. 0: divided by 1. 1: divided by 2. 2: divided by 3. 3: divided by 4.	0x0
1:0	RW	CPU_PLL_Sel	**CPU_PLL Frequency Select.** 0: 175MHz. 1: 200MHz. 2: 225MHz 3: 250MHz	0x0

3.10.5 PLL/Hard Macro Power Down Control Register

Address: 0x10

Table 196. PLL/Hard Macro Power Down Control Register

Bits	Type	Name	Description	Default
7	RW	SYSTEM_XTAL_PWD	**Enable System XTAL (25MHz) Power Down Function** 0: Never power down 1: Power down when CPU enters Sleep mode	0x0
6	RW	PLLx7_PWD	**Power Down PLLx7 (for I2S 44.1KHz audio)** 1: Power down	0x1
5	RW	PLLx2250_PWD	**Power Down PLLx2250 (for UART 14.7456MHz clock, PCM clock, and I2S clocks)** 1: Power down	0x1
4	RW	USBD_PHY_PWD	**Power Down USB Device PHY.** 1: Power down	0x1

			Note: Only when bit 3 of e-Fuse is one, this bit USBD_PHY_PWD can be disabled; otherwise, USB Device PHY will be always disabled.	
3	RW	USBH_PHY_PWD	**Power Down USB Host PHY.** 1: Power down Note: Only when bit 2 of e-Fuse is one, this bit USBH_PHY_PWD can be disabled; otherwise, USB Host PHY will be always disabled.	0x1
2	RW	PLLx3_PWD	**Power Down PLLx3 (for USB Host, USB Device, and 24MHz UART clock).** 1: Power down	0x1
1	RW	PLLx8_PWD	**Power Down PLLx8 (for PCI, USB Host, USB Device, and 24MHz UART clock).** 1: Power down	0x1
0	RW	PLLx5_PWD	**Power Down PLLx5 (for NIC MAC).** 1: Power down	0x1

3.10.6 CPU Initialization Register

Address: 0x14

Table 197. CPU Initialization Register

Bits	Type	Name	Description	Default
21:16	RW	PAT	**PLL Acquisition Time.** This field defines the time the PLL need to be stable when power on or leaving power down mode. The PLL stable time is defined as wake_up_time = (PAT + 1) * 1024 / OSCIN, where OSCIN = 25MHz.	0x02

3.10.7 Pad Drive Strength Control Register

Address: 0x1C

Table 198. Pad Drive Strength Control Register

Bits	Type	Name	Description	Default
3	RW	MII_not_bounded	**MII signal pads are not packaged out.** When MII signals pads are not packaged out, please set this bit to 1 to prevent I/O pads in tri-state to save power.	0x0
2	RW	MII_Speed	**MII/RGMII Pad Drive Strength Select.** 0: RGMII mode. 1: MII mode.	0x0
1:0	RW	PCI_Drive_Sel	**PCI I/O Pad Drive Strength Select** 0: CARDBUS mode. 1: PCI mode 2,3: Reserved	0x1

3.10.8 USB Device Power Management Register

Address: 0x20

Table 199. USB Device Power Management Register

Bits	Type	Name	Description	Default
5	RW	OUTCLKSEL	**USB Device PHY Clock Source Select** 0: Select external 12MHz Xtal clock source 1: Select internal 12MHz clock source	0x1
4	RW	Remote_wakeup	**Remote Wake Up external USB Host**	0x0
0	RO	VBUS	**X_VBUS pin status** 0: The USB device doesn't attach to a Host 1: The USB device attach to a Host	

3.10.9 Regulator Control Register

Address: 0x24

Table 200. Regulator Control Register

Bits	Type	Name	Description	Default
23:22	RW	bg_sel	**Adjust Bandgap Voltage.**	0x0
21:20	RW	gm2_1p25	**Adjust 1.25V Regulator Loop Stability.**	0x1
19:18	RW	gm2_18	**Adjust 1.8V Regulator Loop Stability.**	0x0
17:16	RW	gm2_25	**Adjust 2.5V Regulator Loop Stability.**	0x3
15	RW	pd_1p25	**Power Down 1.25V Regulator (**active HIGH).	0x0
14	RW	pd_18	**Power Down 1.8V Regulator (**active HIGH).	0x0
13	RW	pd_25	**Power Down 2.5V Regulator** (active HIGH) Note, the setting of pd_25 and the following sel_sdr will affect the voltage of 2.5V regulator. Please refer the following table.	0x0
12	RW	sel_sdr	**Select SDR interface** (active HIGH) Note, the setting of pd_25 and sel_sdr will affect the voltage of 2.5V regulator. Please refer the following table.	0x0
11:9	RW	vdd_1p25	**Fine tune regulated 1.25 voltage under DDR**	0x3
8:6	RW	vdd_1p25_sel	**Control 1.25V Regulator Regulated vdd Output.**	0x4
5:3	RW	vdd_18_sel	**Control 1.8V Regulator Regulated vdd Output.**	0x4
2:0	RW	vdd_25_sel	**Control 2.5V Regulator Regulated vdd Output.**	0x3

Table 201. Table of 2.5V Power Scenario

DDRR/SDR Power (V)	MII/RGMII power (V)	sel_sdr	pd_25	VDD_25 (V)
2.5	2.5	0	0	2.5
2.5	3.3	0	0	2.5
3.3	2.5	1	0	2.5
3.3	3.3	1	1	*

Note: When application system uses SDR and MII interface, 2.5V power is not needed. Please power down the 2.5V regulator!

3.10.10 PLLx2250 Control Register

Address: 0x2C

Table 202. PLLx2250 Control Register

Bits	Type	Name	Description	Default
9:7	RW	LDS[3:1]	**Delay-cell performance select**	0x0
6:4	RW	LCP[3:1]	**Charge Pump Performance Select** Recommend to change to 0x0	0x1
3:2	RW	LDFV[1:0]	**Feedback Divider Select**	0x0
1:0	RW	LIC[2:1]	**Charge Pump Current Select** Recommend to change to 0x1	0x3

3.11 UART0 and UART1

Note: UART0 Registers are defined at 0x7800-0000 ~ 0x7800-001C, and UART1 Registers are defined at 0x7880-0000 ~ 0x7880-001C. And they are totally the same.

3.11.1 Receive Buffer Register/Transmit Holding Register/Baud-Rate Divisor Latch

Short Name: RBR (DLAB = 0 for read)
Short Name: THR (DLAB = 0 for write)
Short Name: DLL (DLAB = 1)
Address: 0x00

Table 203. DLAB = 0 for read (RBR)

Bits	Type	Name	Description	Default
7:0	RO	RBR	**Receive Data Port**.	0x00

Table 204. DLAB = 0 for write (THR)

Bits	Type	Name	Description	Default
7:0	WO	THR	**Transmit Data Port**	0x00

Table 205. DLAB = 1 (DLL)

Bits	Type	Name	Description	Default
7:0	RW	DLL	**Baud Rate Divisor Latch Least Significant Byte.** The Divisor Latch is a 16-bit register, whose most significant byte is hold in the following DLM, and its least significant byte is hold in DLL. The Baud Rate can be controlled by DLL, and DLM with the clock generated from Pre-scaler. Division factor from 1 to 65535 can be programmed. When {DLM, DLL} = 0, UART Baud Rate = 0.	0x01

3.11.2 Interrupt Enable Register / Baud-Rate Divisor Latch

Short Name: IER (DLAB = 0)

Short Name: DLM (DLAB = 1)
Address: 0x04

Table 206. DLAB = 0 (IER)

Bits	Type	Name	Description	Default
3	RW	UART Status	This bit enables the UART Status Interrupt when set to logic 1	0x0
2	RW	Receiver Line Status	This bit enables the Receiver Line Status Interrupt when set to logic 1	0x0
1	RW	THR Empty	This bit enables the Transmitter Holding Register Empty Interrupt when set to logic 1	0x0
0	RW	Receiver Data Available	This bit enables the Received Data Available Interrupt (and character reception timeout interrupts in the FIFO mode) when set to logic 1	0x0

Table 207. DLAB = 1 (DLM)

Bits	Type	Name	Description	Default
7:0	RW	DLM	Baud Rate Divisor Latch Most Significant Byte	0x00

3.11.3 Interrupt Identification Register / Pre-scalar Register

Short Name: IIR (DLAB = 0 for read)
Short Name: FCR (DLAB = 0 for write)
Short Name: PSR (DLAB = 1)
Address: 0x08

Table 208. DLAB = 0 for read (IIR)

Bits	Type	Name	Description	Default
7:6	RO	FIFO Mode Enable	These two bits are set when FCR[0] is set as 1.	0x0
4	RO	Tx FIFO full	This bit is set as 1 when Tx FIFO is full.	0x0
3:0	RO	Interrupt Identification Code	These bits identify the highest priority interrupt that is pending. Please view the following Table 214 for detail.	0x1

Table 209. DLAB = 0 for write (FCR)

Bits	Type	Name	Description	Default
7:6	WO	RXFIFO_TRGL	Use to set the trigger level for the Rx FIFO interrupt. 0: 1 character 1: 4 characters 2: 8 characters 3: 14 characters	0x0
2	WC	Tx FIFO Reset	Set this bit to logic 1 clears all bytes in the Tx FIFO and resets its counter logic to 0. The shift register is not cleared, so any reception active will continue. The bit will automatically return to zero.	0x0
1	WC	Rx FIFO Reset	Set this bit to logic 1 clears all bytes in the Rx FIFO and resets its counter logic to 0. The shift register is not cleared, so any reception active will continue.	0x0

			The bit will automatically return to zero.	
0	WO	FIFO Enable	Set this bit to logic 1 enables both the transmitter and receiver FIFO. Changing this bit automatically resets both FIFO.	0x0

Table 210. DLAB = 1 (PSR)

Bits	Type	Name	Description	Default
1:0	RW	PSR	**Set Pre-scalar Value of UART Baud Rate Generator.** The Pre-scalar will generate clock for Baud Rate Divisor as follows: 0: Disable clock. 1: 24MHz 2: 14.7456MHz 3: Reserved **Note** • When selecting 24 MHz as clock source, the following baud rates are available: 1500000 115200 57600 38400 19200 9600 4800 2400 1200 • When selecting 14.7456 MHz as clock source, the following baud rates are available: 921600 460800 230400 115200 57600 38400 19200 9600 4800 2400 1200	0x0

3.11.4 Line Control Register

Short Name: LCR
Address: 0x0C

Table 211. Line Control Register

Bits	Type	Name	Description	Default
7	RW	DLAB	**Divisor Latch Access Bit (DLAB).** This bit must be set in order to access the DLL, DLM and PSR registers which program the division constants for the baud rate divider and	0x0

			the pre-scalar.	
6	RW	Set Break	**Break Control** This bit causes a break condition to be transmitted to the receiving UART. When it is set to HIGH, the serial output (**UR_TXD**) is forced to the Spacing (logic 0) state. When it is set to LOW, the break is disabled. The Break Control bit acts only on **UR_TXD** and has no effect on the transmitter logic, so if several characters are stored in the transmitter's FIFO, they will be removed from this FIFO and passed sequentially to the Transmitter Shift Register, which serializes them. This can be useful to establish the break time making use of the THR Empty and Transmitter Empty flags of the LSR.	0x0
5	RW	Stick Parity	**Enable of Stick Parity** When bits 3, 4 and 5 are logic 1, the Parity bit is transmitted and checked as logic 0. If bits 3 and 5 are HIGH and bit 4 is LOW, then the Parity bit is transmitted and checked as logic 1. If bit 5 is LOW, Stick Parity is disabled.	0x0
4	RW	Even Parity	**Even Parity Select.** The bit is Even Parity Select bit. When bit 3 is logic HIGH and bit 4 is logic LOW, an odd number of logic 1s is transmitted or checked in the data word bits and Parity bit.	0x0
3	RW	Parity Enable	**Parity Enable.** This bit is the Parity Enable bit. When this bit is a logic HIGH, a Parity bit is generated (transmit data) or checked (receive data) between the last data word bit and the Stop bit of the serial data. When bit 3 is logic HIGH and bit 4 is logic HIGH, an even number of logic 1s is transmitted or checked.	0x0
2	RW	Stop Bits	**Select Number of Stop Bits** This bit selects the number of stop bits to be transmitted. If cleared, only one stop bit will be transmitted. If set, two stop bits (1.5 with 5-bit data) will be transmitted before the start bit of the next character. The receiver always checks only one stop bit.	0x0
1	RW	WL1	This bit along with WL0 defines the word length of the data being transmitted and received.	0x0
0	RW	WL0	This bit along with WL1 defines the word length of the data being transmitted and received.	0x0

3.11.5 UART Control Register

Short Name: UCR

Address: 0x10

Table 212. UART Control Register

Bits	Type	Name	Description	Default
5	RW	DMA Mode	In the FIFO mode (FCR[0] = 1), and this bit is set, DMA mode is enabled.	0x0
4	RW	Loop	**Loop-back Mode.** This is the loop back mode control bit. Loop back mode is intended for the UART communication testing.	0x0

3.11.6 UART Line Status /Test Control Register

Short Name: LSR
Address: 0x14

Table 213. For Read

Bits	Type	Name	Description	Default
7	RO	FIFO Data Error	**FIFO Data Error Flag.** If the FIFO is disabled (16450 mode), this bit is always zero. If the FIFO is active, this bit will be set as soon as any data character in the receiver's FIFO has parity or framing error or the break indication active. Note that this bit is cleared when the CPU reads the LSR and the rest of the data in the receiver's FIFO do not have any of these three associated flags on.	0x0
6	RO	Transmitter Empty	**Transmitter Empty Flag.** It is "1" when both the THR (or Tx FIFO) and the TSR (Transmitter Shift Register) are empty. Reading this bit as "1" means that no transmission is currently taking place in the UR_TXD output pin, the transmission line is idle. Note that as soon as new data is written in the THR, this bit will be cleared.	0x0
5	RO	THR Empty	**THR Empty Flag.** It indicates that the UART is ready to accept a new character for transmission. In addition, this bit causes the UART to issue an interrupt to the CPU when the Transmit Holding Register Empty Interrupt enable bit (IER[1]) is set high. In non-FIFO mode, this bit is set whenever the 1-byte THR is empty. If the THR holds data to be transmitted, this bit is immediately set when this data is passed to the TSR. In FIFO mode, this bit is set when the transmitter's FIFO is completely empty, being 0 if there is at least one byte in the FIFO waiting to be passed to the TSR for transmission.	0x0
4	RO	Break Interrupt	It is set to "1" if the receiver's line input UR_RXD	0x0

			was held at zero for a complete character time. It is to say the positions corresponding to the start bit, the data, the parity bit (if any) and the (first) stop bit were all detected as zeroes. Note that a FramingError flag always accompanies this flag. This bit is queued in the receiver's FIFO in the same way as the Parity Error bit. When break occurs, only one zero character is loaded into the FIFO. The next character transfer is enabled after UR_RXD goes to the marking state and receives the next valid start bit. Note that this bit is cleared as soon as the LSR is read.	
3	RO	Framing Error	**Frame Error Flag.** It indicates that the received character did not have a valid stop bit (i.e., a 0 was detected in the (first) stop bit position instead of a 1). This bit is queued in the receiver's FIFO in the same way as the Parity Error bit. When a frame error is detected, the receiver tries to resynchronize: if the next sample is again a zero it will be taken as the beginning of a possible new start bit. Note that this bit is cleared as soon as the LSR is read.	0x0
2	RO	Parity Error	**Parity Error Flag.** When it is set, it indicates that the parity of the received characters wrong according to the current setting in LCR. This bit is queued in the receiver's FIFO, so it is associated to the particular character that had the error. Therefore, LSR must be read before RBR: each time a character is read from RBR, the next character passes to the top of the FIFO and LSR is loaded with the queued error flags corresponding to this top-of-the-FIFO character. Note that this bit is cleared as soon as the LSR is read.	0x0
1	RO	Overrun Error	**Overrun Error Flag.** When it is set, a character has been completely assembled in the Receiver Shift Register without having free space to put it in the receiver's FIFO or holding register. When an overrun condition appears, the result is different depending on whether the 16-byte FIFO is active or not: If the FIFO is not active, so that only a 1-character Receiver Holing Register is available, the unread data in this RBR will not be overwritten with the new character just received. If the FIFO is active, the character just received in the Receiver Shift Register will be overwritten, but the data already present in the FIFO is not changed. The Overrun Error flag is set as soon as the overrun condition appears. It is not queued in the FIFO if this is active. Note that this bit is cleared as soon as the LSR is	0x0

			read.	
0	RO	Data Ready	**Data Ready Flag.** It is set if one or more characters have been received and are waiting in the receiver's FIFO for the user to read them. It is cleared to logic 0 by reading all of the data in the Receiver Buffer Register or the FIFO.	0x0

For Write: Reserved

3.11.7 Scratch Pad Register

Short Name: SPR
Address: 0x1C

Table 214. Scratch Pad Register

Bits	Type	Name	Description	Default
7:0	RW	User Data	This 8-bit read/write register has no effect on the operation of the Serial Port. It is intended as a scratchpad register to be used by the programmer to hold data temporarily.	0x00

Table 215. UART Interrupt Identification

Interrupt Identification Code				Interrupt Set and Reset Functions			
Bits 3	**Bits 2**	**Bits 1**	**Bits 0**	**Priority Level**	**Interrupt Type**	**Interrupt Source Description**	**Interrupt Reset Method**
0	0	0	1	----	None	There is no interrupt pending.	None
0	1	1	0	Highest	Receiver Line Status	There is an overrun error, parity error, framing error or break interrupt indication corresponding to the received data on top of the receive FIFO. Note that the FIFO error flag in LSR does not influence this interrupt, which is related only to the data on top of the Rx FIFO. This is directly related to the presence of a 1 in any of the LSR bits 1 to 4.	Read the Line Status Register (LSR)
0	1	0	0	Second	Received Data Ready	In non-FIFO mode, there is received data available in the RHR register. In FIFO mode, the number of characters in the receive FIFO is equal to or greater than the trigger level programmed in FCR. The interrupt signal will stay active while the number of words in the FIFO stays higher than that value and will be cleared when the microprocessor reads the necessary words to make the number of words in the FIFO less than the trigger level. Note that this is not directly related to LSR bit 0, which always indicates that there is at least one word ready.	Read the Receiver Buffer Register (RBR)

1	1	0	0	Second	Character Reception Timeout	There is at least one character in the receive FIFO and during a time corresponding to four characters at the selected baud rate, no new character has been received. A FIFO timeout interrupt will occur, if the following conditions exist: 1. At least one character is in the FIFO. 2. The most recent serial character received were longer than 4 continuous character times ago (if 2 stop bits are programmed, the second one is included in this time delay).	Read the Receiver Buffer Register (RBR)
0	0	1	0	Third	Transmitter Holding Register Empty	In non-FIFO mode, the 1-byte THR is empty. In FIFO mode, the complete 16-byte transmit FIFO is empty, so 1 to 16 characters can be written to THR. That is to say, THR Empty bit in LSR is one.	Write the Transmitter Holding Register (THR). Alternatively, reading the Interrupt Identification Register (IIR) will also clear the interrupt if this is the interrupt type being currently indicated (this will not clear the flag in the LSR).

3.12 Timer

3.12.1 Timer 1 Counter Register

Address: 0x00

Table 216. Timer 1 Counter Register

Bits	Type	Name	Description	Default
31:0	RW	Tm1Counter	**Timer1 Counter.** If the timer is disabled, Tm1Counter will hold current value. And if the counter up/down counts to 0, an overflow event occurs.	0x00000000

3.12.2 Timer 1 Auto Reload Value Register

Address: 0x04

Table 217. Timer 1 Auto Reload Value Register

Bits	Type	Name	Description	Default
31:0	RW	Tm1Load	**Timer1 Auto Reload Value.** If Timer1 overflow occurs, the value of Tm1Load is loaded into Tm1Counter.	0x00000000

3.12.3 Timer 1 Match Value 1 Register

Address: 0x08

Table 218. Timer 1 Match Value 1 Register

Bits	Type	Name	Description	Default

31:0	RW	Tm1Match1	**Timer1 Match Value1.** When Tm1Counter's value is equal to Tm1Match1 and Timer1 is enabled, the Tm1Match1 interrupt is asserted.	0x00000000

3.12.4 Timer 1 Match Value 2 Register

Address: 0x0C

Table 219. Timer 1 Match Value 2 Register

Bits	Type	Name	Description	Default
31:0	RW	Tm1Match2	**Timer1 Match Value2.** When Tm1Counter's value is equal to Tm1Match2 and Timer1 is enabled, the Tm1Match2 interrupt is asserted.	0x00000000

3.12.5 Timer 2 Counter Register

Address: 0x10

Table 220. Timer 2 Counter Register

Bits	Type	Name	Description	Default
31:0	RW	Tm2Counter	**Timer2 Counter.** If the timer is disabled, Tm2Counter will hold current value. And if the counter up/down counts to 0, an overflow event occurs.	0x00000000

3.12.6 Timer 2 Auto Reload Value Register

Address: 0x14

Table 221. Timer 2 Auto Reload Value Register

Bits	Type	Name	Description	Default
31:0	RW	Tm2Load	**Timer2 Auto Reload Value.** If Timer2 overflow occurs, the value of Tm2Load is loaded into Tm2Counter.	0x00000000

3.12.7 Timer 2 Match Value 1 Register

Address: 0x18

Table 222. Timer 2 Match Value 1 Register

Bits	Type	Name	Description	Default
31:0	RW	Tm2Match1	**Timer2 Match Value1.** When Tm2Counter's value is equal to Tm2Match1 and Timer2 is enabled, the Tm2Match1 interrupt is asserted.	0x00000000

3.12.8 Timer 2 Match Value 2 Register

Address: 0x1C

Table 223. Timer 2 Match Value 2 Register

Bits	Type	Name	Description	Default
31:0	RW	Tm12Match2	**Timer2 Match Value2.** When Tm2Counter's value is equal to Tm2Match2 and Timer2 is enabled, the Tm2Match2 interrupt is asserted.	0x00000000

3.12.9 Timer 1 and 2 Control Register

Address: 0x30

Table 224. Timer 1 and 2 Control Register

Bits	Type	Name	Description	Default
10	RW	TmCR[10]	**Timer2 Up/Down Counter Control.** 0: Up. 1: Down.	0x0
9	RW	TmCR[9]	**Timer1 Up/Down Counter Control.** 0: Up. 1: Down.	0x0
8:6	RO		Reserved	0x0
5	RW	Tm2OFEnable	**Timer2 Overflow Interrupt Enable.** 0: Disable. 1: Enable.	0x0
4	RW	Tm2Clock	**Timer2 Clock Source Select.** 0: PCLK. 1: 1KHz Clock	0x0
3	RW	Tm2Enable	**Timer2 Enable.** 0: Disable. 1: Enable.	0x0
2	RW	Tm1OFEnable	**Timer1 Overflow Interrupt Enable.** 0: Disable. 1: Enable.	0x0
1	RW	Tm1Clock	**Timer1 Clock Source Select.** 0: PCLK. 1: 1KHz Clock	0x0
0	RW	Tm1Enable	**Timer1 Enable.** 0: Disable. 1: Enable.	0x0

3.12.10 Interrupt Status Register

Address: 0x34

Table 225. Interrupt Status Register

Bits	Type	Name	Description	Default
5	RWC	Tm2overflow	Timer2 Overflow Interrupt.	0x0
4	RWC	Tm2Match2	Timer2 Match Value2 Interrupt.	0x0
3	RWC	Tm2Match1	Timer2 Match Value1 Interrupt	0x0
2	RWC	Tm1overflow	Timer1 Overflow Interrupt	0x0
1	RWC	Tm1Match2	Timer1 Match Value2 Interrupt	0x0
0	RWC	Tm1Match1	Timer1 Match Value1 Interrupt	0x0

3.12.11 Interrupt Mask Register

Address: 0x38

Table 226. Interrupt Mask Register

Bits	Type	Name	Description	Default
5	RW	Tm2overflow mask	**Timer2 Overflow Interrupt Mask.** 0: non-mask 1: mask	0x0
4	RW	Tm2Match2 mask	**Timer2 Match Value2 Interrupt Mask.** 0: non-mask 1: mask	0x0
3	RW	Tm2Match1 mask	**Timer2 Match Value1 Interrupt Mask.** 0: non-mask 1: mask	0x0
2	RW	Tm1overflow mask	**Timer1 Overflow Interrupt Mask.** 0: non-mask. 1: mask.	0x0
1	RW	Tm1Match2 mask	**Timer1 Match Value2 Interrupt Mask.** 0: non-mask 1: mask	0x0
0	RW	Tm1Match1 mask	**Timer1 Match Value1 Interrupt Mask.** 0: non-mask 1: mask	0x0

3.12.12 Free Running Timer

Address: 0x40

Table 227. Free Running Timer

Bits	Type	Name	Description	Default
31:0	RO	Tm3CounterL	Tm3Counter[31:0] It is a fine resolution counter. The counter's clock is 100KHz. After reset, it is free running.	0x00000000

3.12.13 Free Running Timer Control Register

Address: 0x44

Table 228. Free Running Timer Control Register

Bits	Type	Name	Description	Default
17	RW	Tm3Run	**Timer3 counter Run** Writing 1 to the bit will start to count from current value of Tm3Counter[47:0]. Write 0 to the bit will stop to count, and keep current value.	0x1
16	WC	Tm3Reset	**Timer3 counter Reset** Writing 1 to the bit will clear Tm3Counter[47:0] to 0, and then restart to count immediately and the above Tm3Run bit will be set to 1. Write 0 is no effect.	0x0
15:0	RO	Tm3CounterH	Tm3Counter[47:32]	0x0000

3.13 Watch Dog Timer

3.13.1 Watch Dog Timer Counter Register

Address: 0x00

Table 229. Watch Dog Timer Counter Register

Bits	Type	Name	Description	Default
31:0	RO	WdCounter	**Watch Dog Timer Counter Register.** WdCounter is a down counter and contains the counter's current value.	0x03EF_1480

3.13.2 Watch Dog Timer Counter Auto-reload Register

Address: 0x04

Table 230. Watch Dog Timer Counter Auto-reload Register

Bits	Type	Name	Description	Default
31:0	RW	WdLoad	**Watch Dog Timer Counter Auto Reload Register.** When reset or restart, the value of WdLoad will be loaded into WdCounter.	0x03EF_1480

3.13.3 Watch Dog Timer Counter Restart Register

Address: 0x08

Table 231. Watch Dog Timer Counter Restart Register

Bits	Type	Name	Description	Default
15:0	WO	WdRestart	**Watch Dog Timer Counter Restart Register.** Writing 0x5AB9 to this register, Watch Dog Timer will automatically reload WdLoad to WDcounter	0x00000000

			and restart to counting.	

3.13.4 Watch Dog Timer Control Register

Address: 0x0C

Table 232. Watch Dog Timer Control Register

Bits	Type	Name	Description	Default
4	RW	WdClock	**Watch Dog Timer Clock Source Select.** 0: PCLK 1: 10Hz Clock	0x0
3	RW	WdExt	**Watch Dog Timer External Signal Enable**. 0: Disable 1: Enable	0x0
2	RW	WdIntr	**Watch Dog Timer System Interrupt Enable**. 0: Disable 1: Enable	0x0
1	RW	WdRst	**Watch Dog Timer System Reset Enable.** 0: Disable. 1: Enable.	0x0
0	RW	WdEnable	**Watch Dog Timer Enable.** 0: Disable. 1: Enable.	0x0

3.13.5 Watch Dog Timer Status Register

Address: 0x10

Table 233. Watch Dog Timer Status Register

Bits	Type	Name	Description	Default
0	RO	WdStatus	**Watch Dog Timer Status.** This bit is set when the counter reaches zero. 0: Does not reach zero. 1: Watch Dog reaches zero.	0x0

3.13.6 Watch Dog Timer Clear Register

Address: 0x14

Table 234. Watch Dog Timer Clear Register

Bits	Type	Name	Description	Default
0	WC	WdClear	**Watch Dog Timer Clear.** Writing 1 to this register will clear WdStatus.	0x0

3.13.7 Watch Dog Timer Interrupt Length Register

Address: 0x18

Table 235. Watch Dog Timer Interrupt Length Register

Bits	Type	Name	Description	Default
7:0	RW	WdIntrlen	**Watch Dog Timer Interrupt Length.** This register controls the length of reset and interrupt.	0xFF

3.14 Real Time Counter

3.14.1 RTC Second Register

Address: 0x00

Table 236. RTC Second Register

Bits	Type	Name	Description	Default
5:0	RO	RtcSecond	**RTC Second Counter Register.** Its range is 0~59.	0x00

3.14.2 RTC Minute Register

Address: 0x04

Table 237. RTC Minute Register

Bits	Type	Name	Description	Default
5:0	RO	RtcMinute	**RTC Minute Counter Register.** Its range is 0~59.	0x00

3.14.3 RTC Hour Register

Address: 0x08

Table 238. RTC Hour Register

Bits	Type	Name	Description	Default
4:0	RO	RtcHour	**RTC Hour Counter Register.** Its range is 0~23.	0x00

3.14.4 RTC Day Register

Address: 0x0C

Table 239. RTC Day Register

Bits	Type	Name	Description	Default
15:0	RO	RtcDays	**RTC Day Counter Register.**	0x0000

3.14.5 RTC Second Alarm Register

Address: 0x10

Table 240. RTC Second Alarm Register

Bits	Type	Name	Description	Default
5:0	RW	AlarmSecond	**RTC Second Alarm Register.** For example, if user wants to set alarm at 12:10:10, the AlarmSecond needs to be set 0xA.	0x3F

3.14.6 RTC Minute Alarm Register

Address: 0x14

Table 241. RTC Minute Alarm Register

Bits	Type	Name	Description	Default
5:0	RW	AlarmMinute	**RTC Minute Alarm Register.** For example, if user wants to set alarm at 12:10:10, the AlarmMinute needs to be set 0xA.	0x3F

3.14.7 RTC Hour Alarm Register

Address: 0x18

Table 242. RTC Hour Alarm Register

Bits	Type	Name	Description	Default
4:0	RW	AlarmHour	**RTC Hour Alarm Register.** For example, if user wants to set alarm at 12:10:10, the AlarmHour needs to be set 0xC.	0x1F

3.14.8 RTC Record Register

Address: 0x1C

Table 243. RTC Record Register

Bits	Type	Name	Description	Default
31:0	RW	RtcRecord	**RTC Record Register.** It is used to adjust the difference of current time and RTC counter time. The following expression can determine the value of RtcRecord. *RtcDays*86400 + RtcHour*3600 + RtcMinute*60 + RtcSecond + RtcRecord = seconds of (Current time – Base time)* *"Base time" is defined by programmer. For example, it can be defined at 2000/01/01/00:00:00. And "Current time" is input by user when system initialization.*	0x00000000

3.14.9 RTC control Register

Address: 0x20

Table 244. RTC control Register

Bits	Type	Name	Description	Default
5	RW	RTC match alarm interrupt	**RTC Match Alarm Interrupt Enable.** When enabled, the RTC rtc_alarm interrupt occurrs every match alarm. 0: Disable. 1: Enable.	0x0
4	RW	RTC interrupt every day	**RTC Auto Alarm Every Day Enable.** When enabled, the RTC rtc_day interrupt occurs every day 0: Disable. 1: Enable.	0x0
3	RW	RTC interrupt every hour	**RTC Auto Alarm Every Hour Enable.** When enabled, the RTC rtc_hour interrupt occurs every hour 0: Disable. 1: Enable.	0x0
2	RW	RTC interrupt every minute	**RTC Auto Alarm Every Minute Enable.** When enabled, the RTC rtc_min interrupt occurs every minute 0: Disable. 1: Enable.	0x0
1	RW	RTC interrupt every second	**RTC Auto Alarm Every Second Enable.** When enabled, the RTC rtc_sec interrupt occurs every second. 0: Disable. 1: Enable.	0x0
0	RW	RTC enable	**RTC Enable.** This bit can enable Real Time Clock or show if the RTC is enabled. When the bit is 0, the RTC is off. 0: Disable 1: Enable Note: CPU can set this bit to 1 to enable RTC. Since this bit will be sync. by 32.768KHz clock, user must wait for a little while to read correct value. After writing 1, write 0 has no effect. It can be reset only by RTC dedicated Power On Reset. System reset has no effect on it.	0x0

3.14.10 Interrupt Status Register

Address: 0x34

Table 245. Interrupt Status Register

Bits	Type	Name	Description	Default
4	RWC	Rtc_alarm	Indicate the rtc_alarm interrupt occurs, write 1	0x0

			clear.	
3	RWC	Rtc_day	Indicate the rtc_day interrupt occurs, write 1 clear.	0x0
2	RWC	Rtc_hour	Indicate the rtc_hour interrupt occurs, write 1 clear.	0x0
1	RWC	Rtc_min	Indicate the rtc_min interrupt occurs, write 1 clear.	0x0
0	RWC	Rtc_sec	Indicate the rtc_sec interrupt occurs, write 1 clear.	0x0

3.15 GPIOA and GPIOB Controller

3.15.1 GPIO Data Output Register

Short Name: GpioDataOut
Address: 0x00

Table 246. GPIO Data Output Register

Bits	Type	Name	Description	Default
31:0	RW	GpioDataOut	**GPIO Data Output Register.** It is Double Word operation logic as usual, comparing to the following **GPIO Data Bit Set Register** and **GPIO Data Bit Clear Register**. Writing to the register will affect every register bits.	0x00000000

3.15.2 GPIO Data Input Register

Short Name: GpioDataIn
Address: 0x04

Table 247. GPIO Data Input Register

Bits	Type	Name	Description	Default
31:0	RO	GpioDataIn	**GPIO Data Input Register.**	0x00000000

3.15.3 GPIO Direction Register

Short Name: PinDir
Address: 0x08

Table 248. GPIO Direction Register

Bits	Type	Name	Description	Default
31:0	RW	PinDir	**GPIO Direction Register.** 0: Pin input. 1: Pin output.	0x00000000

3.15.4 GPIO Data Bit Set Register

Short Name: GpioDataSet
Address: 0x10

Table 249. GPIO Data Bit Set Register

Bits	Type	Name	Description	Default
31:0	W	GpioDataSet	**GPIO Data Bit Set Register.** It is bit operation logic. When write to this register and if some bits of GpioDataSet are 1, the corresponding bits in GpioDataOut register will be set to 1, and the others will not be changed.	0x00000000

3.15.5 GPIO Data Bit Clear Register

Short Name: GpioDataClear
Address: 0x14

Table 250. GPIO Data Bit Clear Register

Bits	Type	Name	Description	Default
31:0	W	GpioDataClear	**GPIO Data Bit Clear Register.** It is bit operation logic. When write to this register and if some bits of GpioDataClear are 1, the corresponding bits in GpioDataOut register will be cleard, and the others will not be changed.	0x00000000

3.15.6 GPIO Interrupt Enable Register

Short Name: IntrEnable
Address: 0x20

Table 251. GPIO Interrupt Enable Register

Bits	Type	Name	Description	Default
31:0	RW	IntrEnable	**GPIO Interrupt Enable Register.** 0: Pin interrupt is disabled 1: Pin interrupt is enabled	0x00000000

3.15.7 GPIO Interrupt Raw Status Register

Short Name: IntrRawState
Address: 0x24

Table 252. GPIO Interrupt Raw Status Register

Bits	Type	Name	Description	Default
31:0	RO	IntrRawState	**GPIO Interrupt Raw Status Register.** 0: Interrupt is not detected. 1: Interrupt is detected.	0x00000000

3.15.8 GPIO Interrupt Masked Status Register

Short Name: IntrMaskedState
Address: 0x28

Table 253. GPIO Interrupt Masked Status Register

Bits	Type	Name	Description	Default
31:0	RO	IntrMaskedState	**GPIO Interrupt Masked Status Register.** 0: Interrupt is not detected or masked. 1: Interrupt is detected and not masked.	0x00000000

3.15.9 GPIO Interrupt Mask Register

Short Name: IntrMask
Address: 0x2C

Table 254. GPIO Interrupt Mask Register

Bits	Type	Name	Description	Default
31:0	RW	IntrMask	**GPIO Interrupt Mask Register.** 0: Mask is disabled. 1: Mask is enabled.	0x00000000

3.15.10 GPIO Interrupt Clear Register

Short Name: IntrClear
Address: 0x30

Table 255. GPIO Interrupt Clear Register

Bits	Type	Name	Description	Default
31:0	WO	IntrClear	**GPIO Interrupt Clear.** Write 0: No effect Write 1: Clear interrupt	0x00000000

3.15.11 GPIO Interrupt Trigger Method Register

Short Name: IntrTrigger
Address: 0x34

Table 256. GPIO Interrupt Trigger Method Register

Bits	Type	Name	Description	Default
31:0	RW	IntrTrigger	**GPIO Interrupt Trigger Method Register.** 0: Edge trigger. 1: Level trigger.	0x00000000

3.15.12 GPIO Interrupt Trigger by Both Edges Register

Short Name: IntrBoth
Address: 0x38

Table 257. GPIO Interrupt Trigger by Both Edges Register

Bits	Type	Name	Description	Default
31:0	RW	IntrBoth	**GPIO Interrupt Edge Trigger by Both.** 0: Single edge. 1: Both edges.	0x00000000

3.15.13 GPIO Interrupt Trigger by Rising-/Falling-Edge or High/Low level Register

Short Name: IntrRiseNeg
Address: 0x3C

Table 258. GPIO Interrupt Trigger by Rising-/Falling-Edge or High/Low level Register

Bits	Type	Name	Description	Default
31:0	RW	IntrRiseNeg	**GPIO Interrupt Triggered by Rising or Failing Edge.** 0: Rising-edge. 1: Falling-edge. **GPIO Interrupt Triggered by High or Low Level.** 0: High-level. 1: Low-level.	0x00000000

3.15.14 GPIO Bounce Enable Register

Short Name: BounceEnable
Address: 0x40

Table 259. GPIO Bounce Enable Register

Bits	Type	Name	Description	Default
31:0	RW	BounceEnable	**GPIO Pre-scale Clock Enable.** When enable, the interrupt pin is sampled by extended clock of PCLK, instead of PCLK. 0: Disable. 1: Enable.	0x00000000

3.15.15 GPIO Bounce clock pre-scale Register

Short Name: BouncePreScale
Address: 0x44

Table 260. GPIO Bounce clock pre-scale Register

Bits	Type	Name	Description	Default
23:0	RW	BouncePreScale	**GPIO Pre-scale.** It is used to adjust interrupt sampling clock	0x00_07D0

			period as the following expression: *Extended Clock Frequency = PCLK/(BouncePreScale+1)* The allowed range is 0x0001 ~ 0xFFFF.	

3.16 PCI Configuration Data

3.16.1 PCI Configuration Data

Short Name: CONFIG_DATA
Address: 0xA000_0000

Table 261. CONFIG_DATA

Bits	Type	Name	Description	Default
31:0	RW	CONFIG_DATA	PCI configuration data access window.	

3.17 PCI Configuration Address

3.17.1 PCI Configuration Address

Short Name: CONFIG_ADDR
Address: 0xA400_0000

Table 262. CONFIG_ADDR

Bits	Type	Name	Description	Default
31	RW	Enable	Configuration Access Enable 0: disable 1: enable	0x0
23:16	RW	Bus number	PCI Bus Number It is used to select 1 of 256 buses in a system. If the Bus Number is zero, Type 0 configuration translation is used. If the Bus Number is non-zero, Type 1 configuration translation is used.	0x00
15:11	RW	Device number	PCI Device Number It is used to select 1 of 21 PCI devices on a given bus. Device 0's IDSEL is connected to AD[11], and Device 1's IDSEL is connected to AD[12], and so on.	0x00
10:8	RW	Function number	Function Number It is used to select 1 of 8 possible functions on a multifunction device.	0x0
7:2	RW	Register number	Register Number It is used to select a 32-bit data in the Configuration Space of the intended target.	0x00

3.18 USB Host 1.1 Configuration

3.18.1 Command Register

Address: 0x04-05

Table 263. Command Register

Bits	Type	Description	Default
2	RW	**Master Enable.** If set to 1, Host Controller(HC) is enabled to run master cycles.	0x0 (Recom. to 0x1)
1	RW	**Operation Register Access Enable.** If set to 1, USB1.1 Operation Registers can be accessed.	0x0 (Recom. to 0x1)

3.18.2 Operational Mode Enable Register

Address: 0x44
Default: 0x22

Table 264. Operational Mode Enable Register

Bits	Type	Description	Default
6	RW	**BuferUnderOrphan.** The buffer point will reset while data under-run occurs.	0x0 (Recom. to 0x1)
5	RW	**NoResp3HS. (LOW Active)** When clear, the Error interrupt is reported after three consecutive USB bus transfer error.	0x1 (Recom. to 0x0)
4	RW	**HcControl bit 9 and bit10 Function Enable.** When this bit is set, the function of RemoteWakeupConnected and RemoteWakeupEnable in OHCI will be enabled. Otherwise, these two bits are always cleared.	0x0
2	RW	**BUFUNDEREOF.** Setting this bit will stop the Data Buffer Module function until next access.	0x0
1	RW	**S3_WAKEUP.** HC will accept wait-state write command after leaving suspend mode if there are no device attached.	0x1
0	RW	**Data Buffer Region16.** When set, the size of the region for the data buffer is 16 bytes. Otherwise, the size is 32 bytes.	0x0

3.19 USB Host 1.1 Operation

3.19.1 Control and Status Partition

3.19.1.1 HcRevision Register

Short Name: HcRevision
Address: 0x00

Table 265. HcRevision Register

Bits	Type	Description	Default
8	RO	**Legacy.** This read-only field is 1 to indicate that the legacy support registers are present in this HC.	0x1
7:0	RO	**Revision.** This read-only field contains the BCD representation of the version of the HCI specification that is implemented by this HC. For example, a value of 11h corresponds to version 1.1. All of the HC implementations that are compliant with current OpenHCI 1.0 specification will have a value of 10h.	0x10

3.19.1.2 HcControl Register

Short Name: HcControl
Address: 0x04

Table 266. HcControl Register

Bits	Type	Description	Default
10	RO	**RemoteWakeupEnable.** This bit is used by HCD to enable or disable the remote wakeup feature upon the detection of upstream resume signaling. When this bit is set and the Resume Detected bit in HC Interrupt Status is set, a remote wakeup is signaled to the host system. Setting this bit has no impact on the generation of hardware interrupt.	0x0
9	RO	**Remote Wakeup Connected.** This bit indicates whether HC supports remote wakeup signaling or not. If remote wakeup is supported and used by the system, it is the responsibility of system firmware to set this bit during POST. HC clears the bit upon a hardware reset but does not alter it upon software reset.	0x0
8	RW	**Interrupt Routing.** This bit determines the routing of interrupts generated by events registered in HC Interrupt Status. If clear, all interrupts are routed to the normal host bus interrupt mechanism. If set, interrupts are routed to the System Management Interrupt. HCD clears this bit upon hardware reset, but it does not alter this bit upon a software reset. HCD uses this bit as a tag to indicate the ownership of HC.	0x0

7:6	RW	**HostControllerFunctionalState for USB.** 0: UsbReset. 1: UsbResume. 2: UsbOperational. 3: UsbSuspend. A transition to UsbOperational from another state causes SOF generation to begin 1 ms later. HCD may determine whether HC has begun sending SOFs by reading the StartofFrame field of HcInterruptStatus. This field may be changed by HC only in the UsbSuspend state. HC may move from the UsbSuspend state to the UsbResume state after detecting the resume signal from a downstream port. HC enters UsbSuspend after a software reset, whereas it enters UsbReset after a hardware reset. The latter also resets the Root Hub and asserts subsequent reset signal to downstream ports.	0x0
5	RW	**BulkListEnable.** This bit is set to enable the processing of the Bulk list in the next Frame. If cleared by HCD, processing of the Bulk list does not occur after the next SOF. HC checks this bit whenever it determines to process the list. When disabled, HCD may modify the list. If HcBulkCurrentED is pointing to a ED to be removed, HCD must advance the pointer by updating HcBulkCurrentED before re-enabling the processing of the list.	0x0
4	RW	**ControlListEnable.** This bit is set to enable the processing of the Control list in the next Frame. If cleared by HCD, the processing of the Control list does not occur after the next SOF. HC must check this bit whenever it determines to process the list. When disabled, HCD may modify the list. If HcControlCurrentED is pointing to a ED to be removed, HCD must advance the pointer by updating HcControlCurrentED before re-enabling the processing of the list.	0x0
3	RW	**IsochronousEnable.** This bit is used by HCD to enable/disable the processing of isochronous EDs. While processing the periodic list in a Frame, HC checks the status of this bit when it finds an Isochronous ED (F=1). If set (enabled), HC continues processing the EDs. If cleared (disabled), HC halts processing of the periodic list (which now contains only isochronous EDs) and begins processing the Bulk/Control lists. Setting this bit is guaranteed to take effect in the next Frame (not the current Frame).	0x0
2	RW	**PeriodicListEnable.** This bit is set to enable the processing of the periodic list in the next Frame. If cleared by HCD, the processing of the periodic list does not occur after the next SOF. HC must check this bit before it starts processing the list.	0x0
1:0	RW	**ControlBulkServiceRatio.** This specifies the service ratio between Control and Bulk EDs. Before processing any of the non-periodic lists, HC must compare the ratio specified with its internal count on how many non-empty Control EDs have been processed, in determining whether to continue serving another Control ED or switching to Bulk EDs.	0x0

		The internal count will be retained when crossing the frame boundary. In case of reset, HCD is responsible for restoring this value. **CBSR** **No. of Control EDs Over Bulk EDs Served** 0 1:1 1 2:1 2 3:1 3 4:1	

3.19.1.3 HcCommandStatus Register

Short Name: HcCommandStatus
Address: 0x08

Table 267. HcCommandStatus Register

Bits	Type	Description	Default
17:16	RO	**SchedulingOverrunCount.** These bits are incremented on each scheduling overrun error. It is initialized to 00b and wraps around at 11b. This will be incremented when a scheduling overrun is detected even if Scheduling Overrun in Hc Interrupt Status has already been set. This is used by HCD to monitor any persistent scheduling problems.	0x0
3	RW	**OwnershipChangeRequest.** This bit is set by an OS HCD to request a change of control of the HC. When set HC will set the Ownership Change field in HcInterrupt Status. After the changeover, this bit is cleared and remains so until the next request from OS HCD.	0x0
2	RW	**BulkListFilled.** This bit is used to indicate whether there are any TDs on the Bulk list. It is set by HCD whenever it adds a TD to an ED in the Bulk list. When HC begins to process the head of the Bulk list, it checks BF. As long as BulkListFilled is 0, HC will not start processing the Bulk list. If BulkListFilled is 1, HC will start processing the Bulk list and will set BF to 0. If HC finds a TD on the list, then HC will set BulkListFilled to 1 causing the Bulk list processing to continue. If no TD is found on the Bulk list, and if HCD does not set BulkListFilled, then BulkListFilled will still be 0 when HC completes processing the Bulk list and Bulk list processing will stop.	0x0
1	RW	**ControlListFilled.** This bit is used to indicate whether there are any TDs on the Control list. It is set by HCD whenever it adds a TD to an ED in the Control list. When HC begins to process the head of the Control list, it checks CLF. As long as ControlListFilled is 0, HC will not start processing the Control list. If CF is 1, HC will start processing the Control list and will set ControlListFilled to 0. If HC finds a TD on the list, then HC will set ControlListFilled to 1 causing the Control list processing to continue. If no TD is found on the Control list, and	0x0

		if the HCD does not set ControlListFilled, then ControlListFilled will still be 0 when HC completes processing the Control list and Control list processing will stop	
0	RW	**HostControllerReset.** This bit is set by HCD to initiate a software reset of HC. Regardless of the functional state of HC, it moves to the UsbSuspend state in which most of the operational registers are reset except those stated otherwise; e.g., the InterruptRouting field of HcControl, and no Host bus accesses are allowed. This bit is cleared by HC upon the completion of the reset operation. The reset operation must be completed within 10 μs. This bit, when set, should not cause a reset to the Root Hub and no subsequent reset signal should be asserted to its downstream ports.	0x0

3.19.1.4 HcInterruptStatus Register

Short Name: HcInterruptStatus
Address: 0x0C

Table 268. HcInterruptStatus Register

Bits	Type	Description	Default
30	RW	**Ownership Change Status.** This bit is set by HC when HCD sets the Ownership Change Request field in HcCommandStatus. This event, when unmasked, will always generate an System Management Interrupt (SMI#) immediately.	0x0
6	RW	**RootHubStatusChange Status.** This bit is set when the content of HcRhStatus or the content of any of HcRhPortStatus[NumberofDownstreamPort] has changed.	0x0
5	RW	**FrameNumberOverflow Status.** This bit is set when the MSb of HcFmNumber (bit 15) changes value, from 0 to 1 or from 1 to 0, and after HccaFrameNumber has been updated.	0x0
4	RO	**UnrecoverableError Status.** This bit is set when HC detects a system error not related to USB. HC should not proceed with any processing nor signaling before the system error has been corrected. HCD clears this bit after HC has been reset. This event is not implemented and is hard-coded to '0'.	0x0
3	RW	**ResumeDetected Status.** This bit is set when HC detects that a device on the USB is asserting resume signaling. It is the transition from no resume signaling to resume signaling causing this bit to be set. This bit is not set when HCD sets the UsbResume state.	0x0
2	RW	**StartofFrame Status.** This bit is set by HC at each start of a frame and after the update of HccaFrameNumber. HC also generates a SOF token at the same time.	0x0
1	RW	**WritebackDoneHead Status.**	0x0

		This bit is set immediately after HC has written HcDoneHead to HccaDoneHead. Further updates of the HccaDoneHead will not occur until this bit has been cleared. HCD should only clear this bit after it has saved the content of HccaDoneHead.	
0	RW	**SchedulingOverrun Status.** This bit is set when the USB schedule for the current Frame overruns and after the update of HccaFrameNumber. A scheduling overrun will also cause the SchedulingOverrunCount of HcCommandStatus to be incremented.	0x0

3.19.1.5 HcInterruptEnable Register

Short Name: HcInterruptEnable
Address: 0x10

Table 269. HcInterruptEnable Register

Bits	Type	Description	Default
31	RW	**MasterInterrupt Enable.** A '0' written to this field is ignored by HC. A '1' written to this field enables interrupt generation due to events specified in the other bits of this register. This is used by HCD as a Master Interrupt Enable.	0x0
30	RW	**OwnershipChange Enable.** 0 : Ignore. 1 : Enable interrupt generation due to Ownership Change.	0x0
6	RW	**RootHubStatusChange Enable.** 0 : Ignore. 1 : Enable interrupt generation due to Root Hub Status Change.	0x0
5	RW	**FrameNumberOverflow Enable.** 0 : Ignore. 1 : Enable interrupt generation due to Frame Number Overflow.	0x0
4	RW	**UnrecoverableError Enable.** This event is not implemented. All writes to this bit will be ignored.	0x0
3	RW	**ResumeDetected Enable.** 0 : Ignore 1 : Enable interrupt generation due to Resume Detect.	0x0
2	RW	**StartofFrame Enable.** 0 : Ignore 1 : Enable interrupt generation due to Start of Frame.	0x0
1	RW	**WritebackDoneHead Enable.** 0 : Ignore 1 : Enable interrupt generation due to HcDoneHead Writeback	0x0
0	RW	**SchedulingOverrun Enable.** 0 : Ignore 1 : Enable interrupt generation due to Scheduling Overrun.	0x0

3.19.1.6 HcInterruptDisable Register

Short Name: HcInterruptDisable
Address: 0x14

Table 270. HcInterruptDisable Register

Bits	Type	Description	Default
31	RW	**MasterInterrupt Disable.** A '0' written to this field is ignored by HC. A '1' written to this field disables interrupt generation due to events specified in the other bits of this register. Note that this field is set after a hardware or software reset.	0x0
30	RW	**OwnershipChange Disable.** 0 : Ignore. 1 : Disable interrupt generation due to Ownership Change.	0x0
6	RW	**RootHubStatusChange Disable.** 0 : Ignore. 1 : Disable interrupt generation due to Root Hub Status Change.	0x0
5	RW	F**rameNumberOverflow Disable.** 0 : Ignore. 1 : Disable interrupt generation due to Frame Number Overflow.	0x0
4	RW	**UnrecoverableError Disable.** This event is not implemented. All writes to this bit will be ignored.	0x0
3	RW	**ResumeDetected Disable.** 0 : Ignore 1 : Disable interrupt generation due to Resume Detect.	0x0
2	RW	**StartofFrame Disable.** 0 : Ignore. 1 : Disable interrupt generation due to Start of Frame.	0x0
1	RW	**WritebackDoneHead Disable.** 0 : Ignore. 1 : Disable interrupt generation due to HcDoneHead Writeback.	0x0
0	RW	Scheduling Overrun Disable 0 : Ignore. 1 : Disable interrupt generation due to Scheduling Overrun.	0x0

3.19.2 Memory Pointer Partition

3.19.2.1 HcHCCA Register

Short Name: HcHCCA
Address: 0x18

Table 271. HcHCCA Register

Bits	Type	Description	Default
31:8	RW	This is the base address of the Host Controller Communication	0x000000

		Area.	

3.19.2.2 HcPeriodCurrentED Register

Short Name: HcPeriodCurrentED
Address: 0x1Ch

Table 272. HcPeriodCurrentED Register

Bits	Type	Description	Default
31:4	RW	**PeriodCurrentED.** This is used by HC to point to the head of one of the Periodic lists that will be processed in the current Frame. The content of this register is updated by HC after a periodic ED has been processed. HCD may read the content in determining which ED is currently being processed at the time of reading.	0x0000000

3.19.2.3 HcControlHeadED Register

Short Name: HcControlHeadED
Address: 0x20h

Table 273. HcControlHeadED Register

Bits	Type	Description	Default
31:4	RW	**ControlHeadED.** HC traverses the Control list starting with the HcControlHeadED pointer. The content is loaded from HCCA during the initialization of HC.	0x0000000

3.19.2.4 HcControlCurrentED Register

Short Name: HcControlCurrentED
Address: 0x24

Table 274. HcControlCurrentED Register

Bits	Type	Description	Default
31:4	RW	**ControlCurrentED.** This pointer is advanced to the next ED after serving the present one. HC will continue processing the list from where it left off in the last Frame. When it reaches the end of the Control list, HC checks the ControlListFilled in HcCommandStatus. If set, it copies the content of HcControlHeadED to HcControlCurrentED and clears the bit. If not set, it does nothing. HCD is allowed to modify this register only when the ControlListEnable of HcControl is cleared. When set, HCD only reads the instantaneous value of this register. Initially, this is set to zero to indicate the end of the Control list.	0x0000000

3.19.2.5 HcBulkHeadED Register

Short Name: HcBulkHeadED
Address: 0x28

Table 275. HcBulkHeadED Register

Bits	Type	Description	Default
31:4	RW	**BulkHeadED.** HC traverses the Bulk list starting with the HcBulkHeadED pointer. The content is loaded from HCCA during the initialization of HC.	0x0000000

3.19.2.6 HcBulkCurrentED Register

Short Name: HcBulkCurrentED
Address: 0x2C

Table 276. HcBulkCurrentED Register

Bits	Type	Description	Default
31:4	**RW**	**BulkCurrentED.** This is advanced to the next ED after the HC has served the present one. HC continues processing the list from where it left off in the last Frame. When it reaches the end of the Bulk list, HC checks the ControlListFilled of HcControl. If set, it copies the content of HcBulkHeadED to HcBulkCurrentED and clears the bit. If it is not set, it does nothing. HCD is only allowed to modify this register when the BulkListEnable of HcControl is cleared. When set, the HCD only reads the instantaneous value of this register. This is initially set to zero to indicate the end of the Bulk list.	0x0000000

3.19.2.7 HcDoneHead Register

Short Name: HcDoneHead
Address: 0x30

Table 277. HcDoneHead Register

Bits	Type	Description	Default
31:4	RW	**DoneHead.** When a TD is completed, HC writes the content of HcDoneHead to the NextTD field of the TD. HC then overwrites the content of HcDoneHead with the address of this TD. This is set to zero whenever HC writes the content of this register to HCCA. It also sets the WritebackDoneHead of HcInterruptStatus.	0x0000000

3.19.3 Frame Counter Partition

3.19.3.1 HcFmInterval Register

Short Name: HcFmInterval
Address: 0x34

Table 278. HcFmInterval Register

Bits	Type	Description	Default
31	RW	**FrameIntervalToggle.** HCD toggles this bit whenever it loads a new value to FrameInterval.	0x0
30:16	RW	**FSLargestDataPacket.** This field specifies a value that is loaded into the Largest Data Packet Counter at the beginning of each frame. The counter value represents the largest amount of data in bits which can be sent or received by the HC in a single transaction at any given time without causing scheduling overrun. The field value is calculated by the HCD.	0x0000
13:0	RW	**FrameInterval.** This specifies the interval between two consecutive SOFs in bit times. The nominal value is set to be 11,999. HCD should store the current value of this field before resetting HC. By setting the HostControllerReset field of HcCommandStatus as this will cause the HC to reset this field to its nominal value. HCD may choose to restore the stored value upon the completion of the Reset sequence.	0x2EDF

3.19.3.2 HcFmRemaining Register

Short Name: HcFmRemaining
Address: 0x38

Table 279. HcFmRemaining Register

Bits	Type	Description	Default
31	RO	**FrameRemainingToggle.** This bit is loaded from the FrameIntervalToggle field of HcFmInterval whenever FrameRemaining reaches 0. This bit is used by HCD for the synchronization between FrameInterval and FrameRemaining.	0x0
13:0	RO	**FrameRemaining.** This counter is decremented at each bit time. When it reaches zero, it is reset by loading the FrameIntervalvalue specified in HcFmInterval at the next bit time boundary. When entering the UsbOperational state, HC re-loads the content with the FrameInterval of HcFmInterval and uses the updated value from the next SOF.	0x0628

3.19.3.3 HcFmNumber Register

Short Name: HcFmNumber
Address: 0x3C

Table 280. HcFmNumber Register

Bits	Type	Description	Default
15:0	RO	**FrameNumber.** This is incremented when HcFmRemaining is re-loaded. It will be rolled over to 0h after FFFFh. When entering the UsbOperational state, this will be incremented automatically. The content will be written to HCCA after HC has incremented the FrameNumber at each frame boundary and sent a SOF but before HC reads the first ED in that Frame. After writing to HCCA, HC will set the StartofFrame in HcInterruptStatus.	0x0000

3.19.3.4 HcPeriodicStart Register

Short Name: HcPeriodicStart
Address: 0x40

Table 281. HcPeriodicStart Register

Bits	Type	Description	Default
13:0	RW	**PeriodicStart.** After a hardware reset, this field is cleared. This is then set by HCD during the HC initialization. The value is calculated roughly as 10% off from HcFmInterval. A typical value will be 3E67h. When HcFmRemaining reaches the value specified, processing of the periodic lists will have priority over Control/Bulk processing. HC will therefore start processing the Interrupt list after completing the current Control or Bulk transaction that is in progress.	0x0000

3.19.3.5 HcLSThreshold Register

Short Name: HcLSThreshold
Address: 0x44

Table 282. HcLSThreshold Register

Bits	Type	Description	Default
11:0	RW	**LSThreshold.** This field contains a value that is compared to the FrameRemaining field prior to initiating a Low Speed transaction. The transaction is started only if FrameRemaining ≥ this field. The value is calculated by HCD with the consideration of transmission and set-up overhead.	0x000

3.19.4 Root Hub Partition

3.19.4.1 HcRhDescriptorA Register

Short Name: HcRhDescriptorA
Address: 0x48
Default: 0x0100_0002

Table 283. HcRhDescriptorA Register

Bits	Type	Description	Default
31:24	RW	**PowerOnToPowerGoodTime.** This byte specifies the duration HCD has to wait before accessing a powered-on port of the Root Hub. It is implementation-specific. The unit of time is 2 ms. The duration is calculated as POTPGT * 2 ms.	0x01
12	RW	**NoOverCurrentProtection.** This bit describes how the overcurrent status for the Root Hub ports is reported. When this bit is cleared, the OverCurrentProtectionMode field specifies global or per-port reporting. 0 : Over-current status is reported collectively for all downstream ports. 1 : No over-current protection supported.	0x0
11	RW	**OverCurrentProtectionMode.** This bit describes how the overcurrent status for the Root Hub ports is reported. At reset, this field should reflect the same mode as PowerSwitchingMode. This field is valid only if the NoOverCurrentProtection field is cleared. 0 : Over-current status is reported collectively for all downstream ports. 1 : Over-current status is reported on a per-port basis.	0x0
10	RO	**DeviceType.** This bit specifies that the Root Hub is not a compound device. The Root Hub is not permitted to be a compound device. This field should always read/write 0.	0x0
9	RW	**NoPowerSwitching.** These bits are used to specify whether power switching is supported or ports are always powered. USB HC supports global power switching mode. When this bit is cleared, the PowerSwitchingMode specifies global or per-port switching. 0 : Ports are power switched. 1 : Ports are always powered on when the HC is powered on.	0x0
8	RW	**PowerSwitchingMode.** This bit is used to specify how the power switching of the Root Hub ports is controlled. USB HC supports global power switching mode. This field is only valid if the NoPowerSwitching field is cleared. 0: all ports are powered at the same time. 1: Each port is powered individually. This mode allows port power to be controlled by either the global	0x0

<table>
<tr><td></td><td></td><td>switch or per-port switching.
If the PortPowerControlMask bit is set, the port responds only to port power commands (Set/ClearPortPower).
If the port mask is cleared, then the port is controlled only by the global power switch (Set/ Clear Global Power).</td><td></td></tr>
<tr><td>7:0</td><td>RO</td><td>NumberDownstreamPorts.
These bits specify the number of downstream ports supported by the Root Hub.
Both of the HC support three downstream ports.

In A version it can be programmed to 02h or 01h, HC0 is controlled by SB CFG 6C bit 0,1 and HC1 is controlled by SB CFG 6C bit 2,3. The setting is tabulated below
<table>
<tr><th>NDP's value</th><th>SB CFG 6C bit0 or bit2</th><th>SB CFG 6C bit1 or bit3</th></tr>
<tr><td>01h</td><td>1</td><td>X</td></tr>
<tr><td>02h</td><td>0</td><td>1</td></tr>
<tr><td>03h</td><td>0</td><td>0</td></tr>
</table></td><td>0x02</td></tr>
</table>

3.19.4.2 HcRhDescriptorB Register

Short Name: HcRhDescriptorB
Address: 0x4C

Table 284. HcRhDescriptorB Register

Bits	Type	Description	Default
31:16	RW	**PortPowerControlMask.** Each bit indicates if a port is affected by a global power control command when PowerSwitchingMode is set. When set, the port's power state is only affected by per-port power control (Set/ClearPortPower). When cleared, the port is controlled by the global power switch (Set/ClearGlobalPower). If the device is configured to global switching mode (PowerSwitchingMode=0), this field is not valid. USB HC implements global power switching. bit 0: Reserved bit 1: Ganged-power mask on Port #1 bit 2: Ganged-power mask on Port #2 bit 3-15: reserved	0x0000
15:0	RW	**DeviceRemovable.** Each bit is dedicated to a port of the Root Hub. When cleared, the attached device is removable. When set, the attached device is not removable. bit 0: Reserved bit 1: Device attached to Port #1 bit 2: Device attached to Port #2 bit 3-15: reserved	0x0000

3.19.4.3 HcRhStatus Register

Short Name: HcRhStatus
Address: 0x50

Table 285. HcRhStatus Register

Bits	Type	Description	Default
31	WO	**ClearRemoteWakeupEnable(Write)** Writing a '1' clears DeviceRemoveWakeupEnable. Writing a '0' has no effect.	0x0
17	RW	**OverCurrentIndicatorChange.** This bit is set by hardware when a change has occurred to the OCI field of this register. The HCD clears this bit by writing a '1'. Writing a '0' has no effect.	0x0
16	RW	**LocalPowerStatusChange(Read).** The Root Hub does not support the local power status feature; thus, this bit is always read as '0'. **SetGlobalPower(Write).** In global power mode (PowerSwitchingMode=0), this bit is written to '1' to turn on power to all ports (clear PortPowerStatus). In per-port power mode, it sets PortPowerStatus only on ports whose PortPowerControlMask bit is not set. Writing a '0' has no effect.	0x0
15	RW	**DeviceRemoteWakeupEnable(Read).** This bit enables a ConnectStatusChange bit as a resume event, causing a UsbSuspend to UsbResume state transition and setting the ResumeDetected interrupt. 0 : **ConnectStatusChange** is not a remote wakeup event. 1 : **ConnectStatusChange** is a remote wakeup event. **SetRemoteWakeupEnable(Write).** Writing a '1' sets DeviceRemoveWakeupEnable. Writing a '0' has no effect.	0x0
1	RO	**OverCurrentIndicator.** This bit reports overcurrent conditions when the global reporting is implemented. When set, an over-current condition exists. When cleared, all power operations are normal. If per-port over-current protection is implemented this bit is always '0'	0x0
0	RW	**LocalPowerStatus(Read).** The Root Hub does not support the local power status feature; thus, this bit is always read as '0'. **ClearGlobalPower(Write).** In global power mode (PowerSwitchingMode=0), This bit is written to '1' to turn off power to all ports (clear PortPowerStatus). In per-port power mode, it clears PortPowerStatus only on ports whose PortPowerControlMask bit is not set. Writing a '0' has no effect.	0x0

3.19.4.4 HcRhPortStatus Register

Short Name: HcRhPortStatus
Address: 0x54 (USB Port 0)
Address: 0x58 (USB Port 1)

Table 286. HcRhPortStatus Register

Bits	Type	Description	Default
20	RW	**PortResetStatusChange.** This bit is set at the end of the 10-ms port reset signal. The HCD writes a '1' to clear this bit. Writing a '0' has no effect. 0 : port reset is not complete. 1 : port reset is complete.	0x0
19	RW	**PortOverCurrentIndicatorChange.** This bit is valid only if over-current conditions are reported on a per-port basis. This bit is set when Root Hub changes the PortOverCurrentIndicator bit. The HCD writes a '1' to clear this bit. Writing a '0' has no effect. 0 : no change in PortOverCurrentIndicator. 1 : PortOverCurrentIndicator has changed.	0x0
18	RW	**PortSuspendStatusChange.** This bit is set when the full resume sequence has been completed. This sequence includes the 20-s resume pulse, LS EOP, and 3-ms resynchronization delay. The HCD writes a '1' to clear this bit. Writing a '0' has no effect. This bit is also cleared when ResetStatusChange is set. 0 : resume is not completed. 1 : resume completed.	0x0
17	RW	**PortEnableStatusChange.** This bit is set when hardware events cause the PortEnableStatus bit to be cleared. Changes from HCD writes do not set this bit. The HCD writes a '1' to clear this bit. Writing a '0' has no effect. 0 : no change in PortEnableStatus. 1 : change in PortEnableStatus.	0x0
16	RW	**ConnectStatusChange.** This bit is set whenever a connect or disconnect event occurs. The HCD writes a '1' to clear this bit. Writing a '0' has no effect. If CurrentConnectStatus is cleared when a SetPortReset, SetPortEnable, or SetPortSuspend write occurs, this bit is set to force the driver to re-evaluate the connection status since these writes should not occur if the port is disconnected. 0 : no change in CurrentConnectStatus. 1 : change in CurrentConnectStatus. *Note: If the DeviceRemovable[NDP] bit is set, this bit is set only after a Root Hub reset to inform the system that the device is attached.*	0x0
9	RW	**LowSpeedDeviceAttached(Read).**	0x0

		This bit indicates the speed of the device attached to this port. When set, a Low Speed device is attached to this port. When clear, a Full Speed device is attached to this port. This field is valid only when the CurrentConnectStatus is set. 0 : full speed device attached. 1 : low speed device attached.	
8	RW	**Port Power Status(Read).** This bit reflects the port's power status, regardless of the type of power switching implemented. This bit is cleared if an overcurrent condition is detected. HCD sets this bit by writing Set Port Power or Set Global Power. HCD clears this bit by writing Clear Port Power or Clear Global Power. Which power control switches will be enabled is determined by Power Switching Mode and Port Power Control Mask[NDP]. In global switching mode (Power Switching Mode=0), only Set/ClearGlobalPower controls this bit. In per-port power switching (Power Switching Mode=1), if the Port Power Control Mask[NDP] bit for the port is set, only Set/ClearPortPower commands are enabled. If the mask is not set, only Set/ Clear Global Power commands are enabled. When port power is disabled, Current Connect Status, Port Enable Status, Port Suspend Status, and Port Reset Status should be reset. 0 : port power is OFF. 1 : port power is ON. **SetPortPower(Write).** The HCD writes a '1' to set the PortPowerStatus bit. Writing a '0' has no effect. *Note: This bit is always reads '1b' if power switching is not supported.*	0x0
4	RW	**PortResetStatus(Read).** When this bit is set by a write to SetPortReset, port reset signaling is asserted. When reset is completed, this bit is cleared when PortResetStatusChange is set. This bit cannot be set if CurrentConnectStatus is cleared. 0 : port reset signal is not active. 1 : port reset signal is active. **SetPortReset(Write).** The HCD sets the port reset signaling by writing a '1' to this bit. Writing a '0' has no effect. If CurrentConnectStatus is cleared, this write does not set PortResetStatus, but instead sets ConnectStatusChange. This informs the driver that it attempted to reset a disconnected port.	0x0
3	RW	**PortOverCurrentIndicator(Read).** This bit is only valid when the Root Hub is configured in such a way that over-current conditions are reported on a per-port basis. If per-port over-current reporting is not supported, this bit is set to 0. If cleared, all power operations are normal for this port. If set, an over-current condition exists on this port. This bit always reflects the over-current input signal 0 : no over-current condition.	0x0

		1 : over-current condition detected. **ClearSuspendStatus(Write).** The HCD writes a '1' to initiate a resume. Writing a '0' has no effect. A resume is initiated only if PortSuspendStatus is set.	
2	RW	**PortSuspendStatus(Read).** This bit indicates the port is suspended or in the resume sequence. It is set by a SetSuspendState write and cleared when PortSuspendStatusChange is set at the end of the resume interval. This bit cannot be set if CurrentConnectStatus is cleared. This bit is also cleared when PortResetStatusChange is set at the end of the port reset or when the HC is placed in the UsbResume state. If an upstream resume is in progress, it should propagate to the HC. 0 : port is not suspended. 1 : port is suspended. **SetPortSuspend(Write).** The HCD sets the PortSuspendStatus bit by writing a '1' to this bit. Writing a '0' has no effect. If CurrentConnectStatus is cleared, this write does not set PortSuspendStatus; instead it sets ConnectStatusChange. This informs the driver that it attempted to suspend a disconnected port.	0x0
1	RW	**PortEnableStatus(Read).** This bit indicates whether the port is enabled or disabled. The Root Hub may clear this bit when an over-current condition, disconnect event, switched-off power, or operational bus error such as babble is detected. This change also causes PortEnabledStatusChange to be set. HCD sets this bit by writing SetPortEnable and clears it by writing ClearPortEnable. This bit cannot be set when CurrentConnectStatus is cleared. This bit is also set, if not already, at the completion of a port reset when ResetStatusChange is set or port suspend when SuspendStatusChange is set. 0 : port is disabled. 1 : port is enabled. **SetPortEnable(Write).** The HCD sets PortEnableStatus by writing a '1'.Writing a '0' has no effect. If CurrentConnectStatus is cleared, this write does not set PortEnableStatus, but instead sets ConnectStatusChange. This informs the driver that it attempted to enable a disconnected port.	0x0
0	RW	**CurrentConnectStatus(Read).** This bit reflects the current state of the downstream port. 0 : no device connected. 1 : device connected. **ClearPortEnable(Write).** The HCD writes a '1' to this bit to clear the PortEnableStatus bit. Writing a '0' has no effect. The CurrentConnectStatus is not affected by any write. *Note: This bit is always read '1b' when the attached device is*	0x0

		nonremovable (DeviceRemoveable[NDP]).	

3.20 USB Host 2.0 Configuration

3.20.1 Command Register

Address: 0x04-05
Default: 0x0000

Table 287. Command Register

Bit	Type	Description	Default
2	RW	**Master Enable.** If set to 1, EHC is enabled to run master cycles.	0x0 (Recom. to 0x1)
1	RW	**Operation Register Access Enable.** If set to 1, USB2.0 Operation Registers can be accessed.	0x0 (Recom. to 0x1)

3.20.2 Operational Mode Enable Register

Address: 0x40-43
Default: 0x0000_0080

Table 288. Operational Mode Enable Register

Bit	Type	Description	Default
31:8	RW	Reserved These bits are only used for test-mode. Changes to these bits will cause undefined behavior.	0x000000
7:5	RW	**FIFO Threshold Control,** These bits are used to control the FIFO threshold level. When FIFO threshold is reached, OUT cycle will be driven by EHC. 000: FIFO threshold is set to 128 bytes 001: FIFO threshold is set to 256 bytes 010: FIFO threshold is set to 384 bytes 011: FIFO threshold is set to 512 bytes 100: FIFO threshold is set to 640 bytes 101: FIFO threshold is set to 768 bytes 110: FIFO threshold is set to 896 bytes 111: FIFO threshold is set to 1024 bytes	0x4
4:3	RW	Reserved These bits are only used for test-mode. Changes to these bits will cause undefined behavior.	
2	RW	**Debug Port Enable.** This register enables the debug port for EHC. 1: Debug port is enabled 0: Debug port is disabled	0x0
1:0	RW	**Write Special Registers Protect.** These registers protect write-special registers.	0x0

		10: Registers can be written Others: Register cannot be written	

3.21 USB Host 2.0 Operation

3.21.1 Capability Registers Length

Short Name: CAPLENGTH
Address: 0x00

Table 289. Capability Registers Length

Bit	Type	Description	Default
7:0	RO	**Capability Register Length.** This register indicates to the length of the host controller capability registers.	0x20

3.21.2 Host Controller Interface Version Number

Short Name: HCIVERSION
Address: 0x02-03

Table 290. Host Controller Interface Version Number

Bit	Type	Description	Default
15:0	RO	**Host Controller Interface Version Number.** This register indicates the EHC support the EHCI Spec Revision 1.0.	0x0100

3.21.3 Structure Parameters

Short Name: HCSPARAMS
Address: 0x04-07
Default: 0x0010_1202

Table 291. Structure Parameters

Bit	Type	Description	Default
23:20	RO	**Debug Port Number.** This register identifies the first port as the debug port.	0x1
15:12	RO	**Number of Companion Controller (N_CC).** This field indicates the number of companion controllers associated with this USB2.0 host controller.	0x1
11:8	RO	**Number of Ports per Companion Controller (N_PCC).** This field indicates the number of ports supported per companion host controller.	0x2
3:0	RO	**Number of Ports (N_PORTS).**	0x2

		This field indicates the number of ports supported on this host controller.	

3.21.4 Capability Parameters

Short Name: HCCPARAMS
Address: 0x08-0B
Default: 0x0000_7070

Table 292. Capability Parameters

Bit	Type	Description	Default
15:8	RO	**EHCI Extend Capabilities Pointer (EECP).** This field indicates the existence of a capability list.	0x70
7:4	RW	**Isochronous Scheduling Threshold.** This field indicates, relative to the current position of the executing host controller, where software can reliable update the isochronous schedule.	0x7
2	RW	**Asynchronous Schedule Park Capability.** If this bit is set to a one, then the host controller supports the park feature for high-speed queue heads in the Asynchronous Schedule.	0x0

3.21.5 USB2.0 Command Register

Short Name: USB2CMD
Address: 0x20-23
Default: 0x0008_0000

Table 293. USB2.0 Command Register

Bit	Type	Description	Default
23:16	RW	**Interrupt Threshold Control.** This field is used by system software to select the maximum rate at which the host controller will issue interrupt.	0x08
11	RW	**Asynchronous Schedule Park Mode Enable.** If the Asynchronous Park Capability bit in the HCCPARAMS register is a one, then this bit defaults to a 1h and is R/W. Software uses this bit to enable or disable Park mode.	0x0
9:8	RW	**Asynchronous Schedule Park Mode Count.** If the Asynchronous Park Capability bit in the HCCPARAMS register is a one, then this bit defaults to a 3h and is R/W. This field contains a count to the number of successive transactions the host controller is allowed to execute from a high-speed queue head on the asynchronous schedule before continuing traversal of the asynchronous schedule.	0x0
7	RW	**Light Host Controller Reset.** It allows the driver to reset the EHCI controller without affecting the state of the ports or the relationship to the companion host controllers.	0x0
6	RW	**Interrupt on Async Advance Doorbell.**	0x0

		This bit is used as a doorbell by software to tell the host controller to issue an interrupt the next time it advances asynchronous schedule.	
5	RW	Asynchronous Schedule Enable This bit controls whether the host controller skips processing the Asynchronous Schedule.	0x0
4	RW	Periodic Schedule Enable This bit controls whether the host controller skips processing the Periodic Schedule.	0x0
1	RW	Host Controller Reset (HCRESET) This control bit is used by software to reset the host controller.	0x0
0	RW	Run/Stop (RS) When set to a 1, the host controller proceeds with execution of the schedule.	0x0

3.21.6 USB2.0 Status Register

Short Name: USB2STS
Address: 0x24-27
Default: 0x0000_1000

Table 294. USB2.0 Status Register

Bit	Type	Description	Default
15	RO	**Asynchronous Schedule Status.** This bit reports the current real status of the Asynchronous Schedule. If this bit is a zero then the status of the Asynchronous Schedule is disable.	0x0
14	RO	**Periodic Schedule Status.** This bit reports the current real status of the Periodic Schedule. If this bit is a zero then the status of the Periodic Schedule is disable.	0x0
13	RO	**Reclamation.** This bit is used to detect an empty asynchronous schedule.	0x0
12	RO	**Host Controller Halted (HCHalted).** This bit is a zero whenever the Run/Stop bit is a one. The Host Controller sets this bit to one after it has stopped executing as a result of the Run/Stop bit being set to 0, either by software or by the Host Controller hardware.	0x1
5	R/WC	**Interrupt on Async Advance.** System software can force the host controller to issue an interrupt the next time the host controller advances the asynchronous schedule by writing a one to the Interrupt on Async Advance Doorbell bit in the USB2CMD register.	0x0
4	R/WC	**Host System Error.** The Host Controller sets this bit to 1 when a serious error occurs during a host system access involving the Host Controller module.	0x0
3	R/WC	**Frame List Rollover.** The Host Controller sets this bit to a one when the Frame List	0x0

		Index rolls over from its maximum value to zero.	
2	R/WC	**Port Change Detect.** The Host Controller sets this bit to a one when any port for which the Port Owner bit is set to zero has a change bit transition from a zero to a one or a Force Port Resume bit transition from a zero to a one as a result of a J-K transaction detected on a suspended port. This bit will also be set as a result of the Connect Status Change being set to a one after system software has relinquished ownership of a connected port by writing a one to a port's Port Owner bit.	0x0
1	R/WC	**USB Error Interrupt (USBERRINT).** The Host Controller sets this bit to 1 when completion of a USB transaction results in an error condition.	0x0
0	R/WC	**USB Interrupt (USBINT).** The Host Controller sets this bit to 1 one the completion of a USB transaction, which result in the retirement of a Transfer Descriptor that had its IOC bit set.	0x0

3.21.7 USB2.0 Interrupt Enable Register

Short Name: USB2INTR
Address: 0x28-2C
Default: 0x0000_0000

Table 295. USB2.0 Interrupt Enable Register

Bit	Type	Description	Default
5	RW	**Interrupt on Async Advance Enable.**	0x0
4	RW	**Host System Error Enable.**	0x0
3	RW	**Frame List Rollover Enable.**	0x0
2	RW	**Port Change Detect Enable.**	0x0
1	RW	**USB Error Interrupt Enable.**	0x0
0	RW	**USB Interrupt Enable.**	0x0

3.21.8 Frame Index Register

Short Name: FRINDEX
Address: 0x2C-2F

Table 296. Frame Index Register

Bit	Type	Description	Default
13:0	RW	**Frame Index.** The value in this register increment at the end of each time frame.	0x0000

3.21.9 Periodic Frame List Base Address Register

Short Name: PERIODICLISTBASE
Address: 0x34-37

Default: undefined

Table 297. Periodic Frame List Base Address Register

Bit	Type	Description	Default
31:12	RW	**Base Address.** These bits correspond to memory address [31:12].	

3.21.10 Current Asynchronous List Address Register

Short Name: ASYNCLISTBASE
Address: 0x38-3B
Default: undefined

Table 298. Current Asynchronous List Address Register

Bit	Type	Description	Default
31:5	RW	**Link Pointer.** These bits correspond to memory address [31:5].	

3.21.11 Configure Flag Register

Short Name: CONFIGFLAG
Address: 0x60-63

Table 299. Configure Flag Register

Bit	Type	Description	Default
0	RW	**Configure Flag (CF).** Host software sets this bit as the last action in its process of configuring the Host Controller. Writing a one to this register will route all port to this host controller.	0x0

3.21.12 Port Status and Control Register

Short Name: PORTSC0
Short Name: PORTSC1
Address: 0x64-67 (Port 0)
Address: 0x68-6B (Port 1)
Default: 0x0000_3000

Table 300. Port Status and Control Register

Bit	Type	Description	Default
22	RW	**Wake on Over-current Enable (WKOC_E).** Writing this bit to a one enables the port to be sensitive to over-current conditions as wake-up events.	0x0
21	RW	**Wake on Disconnect Enable (WKDSCNNT_E).** Writing this bit to a one enables the port to be sensitive to device disconnects as wake-up events.	0x0
20	RW	**Wake on Connect Enable (WKCNNT_E).**	0x0

		Writing this bit to a one enables the port to be sensitive to device connects as wake-up events.	
19:16	RW	**Port Test Control.** When this field is zero, the port is NOT operation in a test mode.	0x0
13	RW	**Port Owner.** This bit unconditionally goes to a 0b when the Configured bit in the CONFIGFLAG register makes a 0b to 1b transition. This bit unconditionally goes to 1b whenever the Configured bit is zero. Software writes a one to this bit when the attached device is not a high-speed device.	0x1
12	RO	**Port Power (PP).** The Host Controller does not have port power control switches. Each port is hard-wired to power.	0x1
11:10	RO	**Line Status.** These bits reflect the current logical levels of the D+ (bit 11) and D- (bit 10) signal lines.	0x0
8	RW	**Port Reset.** When software writes a one to this bit, the bus reset sequence as defined in the USB Spec Revision 2.0 is started. Software writes a zero to this bit to terminate the bus reset sequence.	0x0
7	RW	**Suspend.** Software writes a one to this bit to suspend the downstream port. A write of zero to this bit is ignored by the host controller. The host controller will unconditionally set this bit to a zero when software sets the Force Port Resume from 1 to 0 or sets the Port Reset bit to 1.	0x0
6	RW	**Force Port Resume.** Software sets this bit to a 1 to driver resume signaling. The Host Controller sets this bit to a 1 if a J-to-K transition is detected while the port is in the Suspend state. A write of zero to this bit will force the downstream port follows the resume sequence follows the defined sequence documented in the USB Spec Revision 2.0.	0x0
5	R/WC	**Over-current Change.** This bit gets set to a one when there is a change to Over-current Active.	0x0
4	RO	**Over-current Active.** 0: This port does not have an over-current condition. 1: This port has an over-current condition.	0x0
3	R/WC	**Port Enable / Disable Change.** For the root hub, this bit gets set to a one only when a port is disabled due to the appropriate conditions existing at the EOF2 pointer.	0x0
2	RW	**Port Enable / Disabled.** Ports can only be enabled by the host controller as a part of the reset and enable. Software cannot enable a port by writing a one to this field. Ports can be disabled by either a fault condition or by host software.	0x0
1	R/WC	**Connect Status Change.** 1: Change in Current Connect Status.	0x0

		0: No change.	
0	RW	**Current Connect Status.** This value reflects the current connect status of the port.	0x0

3.22 USB 1.1/2.0 Device Controller

3.22.1 General Register

3.22.1.1 Main Control Register

Address: 0x00

The main control register sets the general controls for the whole device, which includes wake-up behaviors, global interrupt activity, and software reset. The bit allocation is given in the following table.

Table 301. Main Control Register

Bits	Type	Name	Description	Default
7	-	AHB_RST	**AHB Software Reset**: Write a '1' to reset HBS, HBF and BFC. This bit is automatically cleared by hard-ware reset.	-
6	R	HS_EN	**High Speed Enable:** 1: Device is in High-Speed mode. 0: Device is in Full-Speed mode.	0x0
5	R/W	CHIP_EN	**Chip Enable**: Write a '1' to enable the PAM FIFO's write cycle.	0x0
4	R/W	SFRST	**Software Reset**: Write a '1' to set a software-initiated reset to the controller. This bit cannot be set when controller is in the suspend mode, since the u_clk is stopped. Setting of this bit will cause the de-assertion of pw_save output if it is asserted. • The chirp sequence will be terminated while this bit is set. • The command FIFO will also be cleared by setting this bit. • The Frame Number Register and SOF Timer Mask Register will be cleared too. • The micro-frame number in the RGF will be cleared. Note: the data FIFO status will not be cleared	0x0

			by this reset.	
3	R/W	GOSUSP	**Go Suspend**: Writing a '1' will activate the suspend mode.	0x0
2	R/W	GLINT_EN	**Global Interrupt Enable**: A '1' enables all interrupts. Individual interrupts can be masked by setting the corresponding bits in the interrupt mask register (index 0x11 ~ 0x17).	0x0
1	R/W	FLUSH_HBF	**Flush HBF**: Write a '1' to flush HBF data to FIFO. This bit is automatically cleared by hard-ware.	0x0
0	R/W	CAP_RMWKUP	**Capability of Remote Wake-up**: A '1' indicates the controller has the capability of being wakened up by a 'wakeup' signal.	0x0

3.22.1.2 Device Address Register

Address: 0x01

The device address register stores a USB device's assigned address, and enables the USB device after executing the set-configuration command.

Table 302. Device Address Register

Bits	Type	Name	Description	Default
7	R/W	AFT_CONF	**After Set-Configuration:** A '1' indicates the device has successfully executed SET_CONFIGURATION command. Note: Before this bit is set, controller will not respond to (i.e. timeout) any non-control transfer sent by the host. It is the responsibility of AP to set this bit when the AP receives a SET_CONFIGURATION command. Otherwise, the controller will timeout non-control transfers following this command.	0x0
6:0	R/W	DEVADR [6:0]	**Device Address:** Records the latest USB device address for each SET_ADDRESS.	0x00

3.22.1.3 Test Register

Address: 0x02

The CXF loop back test procedure is described as follows:

1. Clear "TST_CLREA" and set "TST_LPCX" at the same time.

2. Write data to control FIFO via C_BUS.

3. Set the CX_DONE bit of CX Configuration and Status Register (Address = 0x0B, bit0).

4. Poll the CX_DONE bit until its value is 1.

5. Set "TST_CLREA" and clear "TST_LPCX" at the same time.

6. Clear the CX_DONE bit.

7. Read data from control FIFO via C_BUS.

8. Compare data.

Table 303. Test Register

Bits	Type	Name	Description	Default
7	R/W	TST_HALF_SPEED	**Set Half-Speed:** A '1' turns on the half-speed mode. When this bit is set to 1, the controller allows AP to access the FIFO only at each even cycle of u_clk.	0x0
6	R/W	TST_MOD	**Test Mode:** A '1' turns on the test_mode. When this bit is set to 1, the controller will enter the test mode. In the normal mode, the controller uses a counter for 10ms detection of the USB reset. The count is a large number. In the test mode, the controller will use a smaller count for the USB reset detection to save the test cycles on test machine.	0x0
5	R/W	TST_DISTOG	**Disable Toggle Sequence:** A '1' disables the toggle sequence.	0x0
4	R/W	TST_DISCRC	**Disable CRC:** When setting this bit as '1', the controller will not append CRC for upstream packets.	0x0
3	R/W	TST_DISGENSOF	**Disable Self Generation of SOF:** It uses the SOF sent from the host instead of generating a SOF by controller itself.	0x0
2	R/W	TST_CLREA	**Clear External Side Address:** Writing a '1' then a '0' to clear external side address for loop back test.	0x0
1	R/W	TST_LPCX	**Loop Back Test for CX:** A '1' indicates the loop-back test is active for the control transfer (endpoint 0).	0x0
0	R/W	TST_CLRFF	**Clear FIFO**: Writing a '1' will clear all the PAM's FIFO	0x0

			counters and location counters.	

3.22.1.4 SOF Frame Number Register Byte 0

Address: 0x04

The frame number records the last successfully received Start of Frame (SOF) number. The register contains two bytes and the bit allocation is given in the following table.

Table 304. SOF Frame Number Register Byte 0

Bits	Type	Name	Description	Default
7:0	R	SOFN [7:0]	**SOF Frame Number Bits [7:0]** Record the frame number for high speed and full speed.	0x00

3.22.1.5 SOF Frame Number Register Byte 1

Address: 0x05

Table 305. SOF Frame Number Register Byte 1

Bits	Type	Name	Description	Default
5:3	R	USOFN [2:0]	**SOF Microframe Number Bits [2:0]** Record the microframe number during high speed.	0x0
2:0	R	SOFN [10:8]	**SOF Frame Number Bits [10:8]** Record the frame number for high speed and full speed.	0x0

3.22.1.6 SOF Mask Timer Register Byte 0

Address: 0x06

This two-byte register is used to mask the last SOF.

Table 306. SOF Mask Timer Register Byte 0

Bits	Type	Name	Description	Default
7:0	R/W	SOFTM [7:0]	**SOF Mask Timer:** Time since the last SOF in the 30 MHz clock bit.	0x00

3.22.1.7 SOF Mask Timer Register Byte 1

Address: 0x07

Table 307. SOF Mask Timer Register Byte 1

Bits	Type	Name	Description	Default
7:0	R/W	SOFTM [15:8]	**SOF Mask Timer:** Time since the last SOF in the 30MHz clock bit.	0x00

3.22.2 Test Register

3.22.2.1 PHY Test Mode Selector Register

This one-byte register allows the firmware to set the D+, D- lines to predetermined states for testing purpose. The bit allocation is given in the following table. Please note that only one bit can be set at a time.

Address: 0x08

Table 308. Test Register

Bits	Type	Name	Description	Default
6	R/W	TST_FAIL	**Test Interface Fail:** This bit will be asserted to inform users that there is a crc16_err or pid_err error during receiving data with 'TST_INF' being active. This bit must be cleared by user manually.	0x0
5	R/W	TST_INF	**Test Interface between PHY and Controller:** Upon writing a '1' to this bit, the controller starts to send random patterns to PHY automatically at high-speed mode. At the same time, controller also receives the data bypassing PHY. Controller can verify if there is any timing violation at inter-face between PHY and Controller by checking whether errors occurred or not. This bit will be cleared automatically when EOP is received. Note: Before writing a '1' to this bit, the "Vendctrol" pins of PHY must be set to 3.	0x0
4	R/W	TST_PKT	**Test Mode for Packet:** Upon writing a '1' to this bit, controller would repeatedly send the packet defined in the UTMI specification to the transceiver. After the set_feature command shows the test mode and the index Test_Packet is decoded, this bit will be asserted.	0x0
3	R/W	TST_SE0NAK	Upon writing a '1', the D+ / D- lines are set to HS, quiescent state. The device only responds	0x0

			to a valid HS IN token and always responds to the IN token with NAK.	
2	R/W	TST_KSTA	Upon writing a '1', the D+ / D- is set to the high-speed K state.	0x0
1	R/W	TST_JSTA	Upon writing a '1', the D+ / D- is set to the high-speed J state.	0x0
0	R/W	UNPLUG	When the UNPLUG is set to logic '1', the device controller will set the PHY to Non-Driving mode to emulate the detachment of a device even if it is really plugged. The USB host will not detect a device's plug. Such event is called "soft-detachment." After a hardware reset, the UNPLUG will be in logic '1'. The device is now soft-detached. To enable the USB host in order to detect a device's attachment, the PHY must drive D+ and D- in the manner defined in the USB specifications. In order to enable PHY to drive D+ and D-, the AP should clear the UNPLUG bit after a hardware reset. If the AP does not clear the UNPLUG bit, the device will always be soft-detached and the USB host will never detect the device's attachment.	0x1

3.22.2.2 Vendor Specific IO Control Register

Address: 0x09

This register is provided for vendor defined test control and status for PHY.

Table 309. Vendor Specific IO Control Register

Bits	Type	Name	Description	Default
4		VCTLOAD_N	**Vendor-Specific Test Mode Control Load.** This bit controls the active low output u_vctload_n to PHY. A '1' in this bit makes u_vctload_n output a '1'. When the bit is cleared, u_vctload_n outputs a '0'.	0x0
3:0		VCTL [3:0]	**Vendor-Specific Test Mode Control.** The programmed value is delivered to PHY via the output "u_vctl".	0x0

3.22.2.3 Vendor Specific IO Status Register

Address: 0x0A

The reset value depends on the PHY.

Table 310. Vendor Specific IO Status Register

Bits	Type	Name	Description	Default
7:0	R	VSTA [7:0]	**Vendor-Specific Test Mode Status.**	The value is derived directly from PHY

3.22.2.4 CX Configuration and Status Register

Address: 0x0B

The CX Configuration and the Status Register are used to control the FIFO management for endpoint 0. The maximum packet size for endpoint 0 is 64 bytes.

Table 311. CX Configuration and Status Register

Bits	Type	Name	Description	Default
5	-	CX_EMP	**CX FIFO is Empty:** A '1' indicates that the endpoint 0 FIFO is empty.	-
4	R	CX_FUL	**CX FIFO is Full:** A '1' indicates that the endpoint 0 FIFO is full.	0x1
3	R	CX_CLR	**Clear CX FIFO Data**: Write a '1' to clear the data in endpoint 0 FIFO. Note: for endpoint 0, all the data in FIFO will be cleared no matter if the previous SETUP or IN or OUT transaction has completed or not.	0x0
2	R	CX_STL	**Stall CX**: Writing a '1' to this bit can stall Endpoint 0, and the endpoint 0 FIFO will be cleared at the same time. Once CX_STL is set, AP cannot access endpoint 0 FIFO until this bit is cleared. The stall status will be cleared by the next setup transac-tion. This bit will be cleared automatically when the endpoint 0 transaction has ended. Upon detection of a bus reset, the firmware should clear this bit. Please note that when AP wants to set CX_STL, it should set CX_DONE bit at the same time, within one write operation to CX Configuration and Status Register. In other words, the AP cannot set CX_STL with one write operation and then set CX_DONE with another write operation; it must set both bits within one write operation.	0x0
1	R/W	TST_PKDONE	**Data Transfer is Done for Test Packet**: Firmware has completed sending the whole	0x0

			test patterns to the endpoint 0 FIFO for a PHY test by writing a '1' to this bit. This bit is cleared by a hardware reset.	
0	R/W	CX_DONE	**Data Transfer is Done for CX**: Firmware has finished the whole packet transaction for endpoint 0 by writing a '1' to this bit. This bit is cleared by a hardware reset. This bit is cleared by the in-ternal signal "p_endcx" or "p_comfail".	0x0

3.22.2.5 Endpoint 0 Data Port Register Byte 0

Address: 0x0C

The address 0x0C provides direct access for a micro-controller to the FIFO for endpoint 0. In order to access the endpoint 0 FIFO, AP should only use the address 0x0C.

For example, in order to read a 31-byte packet from CXF FIFO, the AHB master should issue the following cycles:

Table 312. Endpoint 0 Data Port Register Byte 0

Cycle Number	HADDR	HTRANS	HSIZE
1	0x0C	NONSEQUENTIAL	word
2	0x0C	NONSEQUENTIAL	word
3	0x0C	NONSEQUENTIAL	word
4	0x0C	NONSEQUENTIAL	word
5	0x0C	NONSEQUENTIAL	word
6	0x0C	NONSEQUENTIAL	word
7	0x0C	NONSEQUENTIAL	word
8	0x0C	NONSEQUENTIAL	halfword
9	0x0C	NONSEQUENTIAL	byte

Alternatively, the AHB master may issue eight (8) NONSEQUENTIAL read cycles with a word size and discard the invalid byte.

3.22.3 Interrupt Mask Register

This register masks the individual interrupt sources. The interrupts for each source group and source byte can be individually controlled via corresponding mask bits. All interrupts can be globally disabled by setting bit GLINT_EN as 0 in the Main Control Register.

3.22.3.1 Interrupt Group Mask Register

Address: 0x10

To disable an interrupt, the micro-controller should set the corresponding bit as 1.

Table 313. Interrupt Group Mask Register

Bits	Type	Name	Description	Default
7	R/W	MINT_SCR7	**Mask all the interrupt bits of Interrupt Source Register Byte 7.** 0: Enable corresponding interrupt 1: Disable corresponding interrupt	0x0
6	R/W	MINT_SCR6	**Mask all the interrupt bits of Interrupt Source Register Byte 6.** 0: Enable corresponding interrupt 1: Disable corresponding interrupt	0x0
5	R/W	MINT_SCR5	**Mask all the interrupt bits of Interrupt Source Register Byte 5.** 0: Enable corresponding interrupt 1: Disable corresponding interrupt	0x0
4	R/W	MINT_SCR4	**Mask all the interrupt bits of Interrupt Source Register Byte 4.** 0: Enable corresponding interrupt 1: Disable corresponding interrupt	0x0
3	R/W	MINT_SCR3	**Mask all the interrupt bits of Interrupt Source Register Byte 3.** 0: Enable corresponding interrupt 1: Disable corresponding interrupt	0x0
2	R/W	MINT_SCR2	**Mask all the interrupt bits of Interrupt Source Register Byte 2.** 0: Enable corresponding interrupt 1: Disable corresponding interrupt	0x0
1	R/W	MINT_SCR1	**Mask all the interrupt bits of Interrupt Source Register Byte 1.** 0: Enable corresponding interrupt 1: Disable corresponding interrupt	0x0
0	R/W	MINT_SCR0	**Mask all the interrupt bits of Interrupt Source Register Byte 0.** 0: Enable corresponding interrupt 1: Disable corresponding interrupt	0x0

3.22.3.2 Interrupt Mask Register Byte 0

Address: 0x11

Table 314. Interrupt Mask Register Byte 0

Bits	Type	Name	Description	Default
7	R/W	MR_COM_ABORT	**Mask command abort interrupt.** 0: Enable corresponding interrupt 1: Disable corresponding interrupt	0x0
5	R/W	MRBUF_ERR	**Mask the read HBF error interrupt bit.**	0x1

			0: Enable corresponding interrupt 1: Disable corresponding interrupt	
4	R/W	MCX_COMFAIL_INT	**Mask the interrupt caused by the host emitting extra IN or OUT data.** 0: Enable corresponding interrupt 1: Disable corresponding interrupt	0x0
3	R/W	MCX_COMEND_INT	**Mask the host end of command (entering status stage) interrupt bit.** 0: Enable corresponding interrupt 1: Disable corresponding interrupt	0x0
2	R/W	MCX_OUT_INT	**Mask the interrupt bits of endpoint 0 for OUT.** 0: Enable corresponding interrupt 1: Disable corresponding interrupt	0x0
1	R/W	MCX_IN_INT	**Mask the interrupt bits of endpoint 0 for IN.** 0: Enable corresponding interrupt 1: Disable corresponding interrupt	0x0
0	R/W	MCX_SETUP_INT	**Mask endpoint 0 setup data received interrupt bit.** 0: Enable corresponding interrupt 1: Disable corresponding interrupt	0x0

3.22.3.3 Interrupt Mask Register Byte 1

Address: 0x12

Table 315. Interrupt Mask Register Byte 1

Bits	Type	Name	Description	Default
7	R/W	MF3_SPK_INT	**Mask the Short Packet Interrupt of FIFO 3.** 0: Enable corresponding interrupt 1: Disable corresponding interrupt	0x1
6	R/W	MF3_ OUT_INT	**Mask the OUT interrupt of FIFO 3.** 0: Enable corresponding interrupt 1: Disable corresponding interrupt	0x1
5	R/W	MF2_SPK_INT	**Mask the Short Packet Interrupt of FIFO 2.** 0: Enable corresponding interrupt 1: Disable corresponding interrupt	0x1
4	R/W	MF2_ OUT_INT	**Mask the OUT interrupt of FIFO 2.** 0: Enable corresponding interrupt 1: Disable corresponding interrupt	0x1
3	R/W	MF1_SPK_INT	**Mask the Short Packet Interrupt of FIFO 1.** 0: Enable corresponding interrupt 1: Disable corresponding interrupt	0x1
2	R/W	MF1_ OUT_INT	**Mask the OUT interrupt of FIFO 1.** 0: Enable corresponding interrupt 1: Disable corresponding interrupt	0x1
1	R/W	MF0_SPK_INT	**Mask the Short Packet Interrupt of FIFO 0** 0: Enable corresponding interrupt	0x1

			1: Disable corresponding interrupt	
0	R/W	MF0_ OUT_INT	**Mask the OUT interrupt of FIFO 0.** 0: Enable corresponding interrupt 1: Disable corresponding interrupt	0x1

3.22.3.4 Interrupt Mask Register Byte 2

Address: 0x13

Table 316. Interrupt Mask Register Byte 2

Bits	Type	Name	Description	Default
7	R/W	MF7_SPK_INT	**Mask the Short Packet Interrupt of FIFO 7.** 0: Enable corresponding interrupt 1: Disable corresponding interrupt	0x1
6	R/W	MF7_OUT_INT	**Mask the OUT interrupt of FIFO 7.** 0: Enable corresponding interrupt 1: Disable corresponding interrupt	0x1
5	R/W	MF6_SPK_INT	**Mask the Short Packet Interrupt of FIFO 6.** 0: Enable corresponding interrupt 1: Disable corresponding interrupt	0x1
4	R/W	MF6_OUT_INT	**Mask the OUT interrupt of FIFO 6.** 0: Enable corresponding interrupt 1: Disable corresponding interrupt	0x1
3	R/W	MF5_SPK_INT	**Mask the Short Packet Interrupt of FIFO 5.** 0: Enable corresponding interrupt 1: Disable corresponding interrupt	0x1
2	R/W	MF5_OUT_INT	**Mask the OUT interrupt of FIFO 5.** 0: Enable corresponding interrupt 1: Disable corresponding interrupt	0x1
1	R/W	MF4_SPK_INT	**Mask the Short Packet Interrupt of FIFO 4.** 0: Enable corresponding interrupt 1: Disable corresponding interrupt	0x1
0	R/W	MF4_OUT_INT	**Mask the OUT interrupt of FIFO 4.** 0: Enable corresponding interrupt 1: Disable corresponding interrupt	0x1

3.22.3.5 Interrupt Mask Register Byte 4

Address: 0x15

Table 317. Interrupt Mask Register Byte 4

Bits	Type	Name	Description	Default
7	R/W	MF15_SPK_INT	**Mask the Short Packet Interrupt of FIFO 15.** 0: Enable corresponding interrupt 1: Disable corresponding interrupt	0x1

6	R/W	MF15_ OUT_INT	**Mask the OUT interrupt of FIFO 15.** 0: Enable corresponding interrupt 1: Disable corresponding interrupt	0x1
5	R/W	MF14_SPK_INT	**Mask the Short Packet Interrupt of FIFO 14.** 0: Enable corresponding interrupt 1: Disable corresponding interrupt	0x1
4	R/W	MF14_ OUT_INT	**Mask the OUT interrupt of FIFO 14.** 0: Enable corresponding interrupt 1: Disable corresponding interrupt	0x1

3.22.3.6 Interrupt Mask Register Byte 5

Address: 0x16

Table 318. Interrupt Mask Register Byte 5

Bits	Type	Name	Description	Default
7	R/W	MF7_ IN_INT	**Mask the IN interrupt bits of FIFO 7.** 0: Enable corresponding interrupt 1: Disable corresponding interrupt	0x1
6	R/W	MF6_ IN_INT	**Mask the IN interrupt bits of FIFO 6.** 0: Enable corresponding interrupt 1: Disable corresponding interrupt	0x1
5	R/W	MF5_ IN_INT	**Mask the IN interrupt bits of FIFO 5.** 0: Enable corresponding interrupt 1: Disable corresponding interrupt	0x1
4	R/W	MF4_ IN_INT	**Mask the IN interrupt bits of FIFO 4.** 0: Enable corresponding interrupt 1: Disable corresponding interrupt	0x1
3	R/W	MF3_ IN_INT	**Mask the IN interrupt bits of FIFO 3.** 0: Enable corresponding interrupt 1: Disable corresponding interrupt	0x1
2	R/W	MF2_ IN_INT	**Mask the IN interrupt bits of FIFO 2.** 0: Enable corresponding interrupt 1: Disable corresponding interrupt	0x1
1	R/W	MF1_ IN_INT	**Mask the IN interrupt bits of FIFO 1.** 0: Enable corresponding interrupt 1: Disable corresponding interrupt	0x1
0	R/W	MF0_ IN_INT	**Mask the IN interrupt bits of FIFO 0.** 0: Enable corresponding interrupt 1: Disable corresponding interrupt	0x1

3.22.3.7 Interrupt Mask Register Byte 6

Address: 0x17

Table 319. Interrupt Mask Register Byte 6

Bits	Type	Name	Description	Default

7	R/W	MF15_ IN_INT	**Mask the IN interrupt bits of FIFO 15.** 0: Enable corresponding interrupt 1: Disable corresponding interrupt	0x1
6	R/W	MF14_ IN_INT	**Mask the IN interrupt bits of FIFO 14.** 0: Enable corresponding interrupt 1: Disable corresponding interrupt	0x1

3.22.3.8 Interrupt Mask Register Byte 7

Address: 0x18

Table 320. Interrupt Mask Register Byte 7

Bits	Type	Name	Description	Default
7	R/W	MRX0BTYE_INT	**Mask Received Zero-length Data Packet Interrupt** Masks the active to the received zero-length data packet interrupt bit. 0: Enable corresponding interrupt 1: Disable corresponding interrupt	0x0
6	R/W	MTX0BYTE_INT	**Mask Transferred Zero-length Data Packet Interrupt** Masks the active to transferred zero-length data packet interrupt bit. 0: Enable corresponding interrupt 1: Disable corresponding interrupt	0x0
5	R/W	MISO_SEQ_ABORT_INT	**Mask ISO Sequential Abort Interrupt** Masks the active to the ISO sequential abort interrupt bit. 0: Enable corresponding interrupt 1: Disable corresponding interrupt	0x0
4	R/W	MISO_SEQ_ERR_INT	**Mask ISO Sequential Error Interrupt** Masks the active to received ISO sequential error interrupt bit. 0: Enable corresponding interrupt 1: Disable corresponding interrupt	0x0
3	R/W	MRESM_INT	**Mask Resume Interrupt** Mask the active to the resume state change interrupt bit. 0: Enable corresponding interrupt 1: Disable corresponding interrupt	0x0
2	R/W	MSUSP_INT	**Mask Suspend Interrupt** Mask the active to suspend state change interrupt bit. 0: Enable corresponding interrupt 1: Disable corresponding interrupt	0x0
1	R/W	MUSBRST_INT	**Mask USB Reset Interrupt** Mask the bus reset interrupt bit. 0: Enable corresponding interrupt 1: Disable corresponding interrupt	0x0

0	R/W	MHBF_EMPTY_INT	**Mask HBF Empty interrupt** Mask the MBF empty interrupt bit. 0: Enable corresponding interrupt 1: Disable corresponding interrupt	0x0

3.22.3.9 Receive Zero-length Data Packet Register Byte 0

Address: 0x19

When a "Receive Zero-length Data Packet Interrupt" occurs (register 0x28 bit 7), the firmware will further check registers 0x19 and 0x1A to determine which endpoint receives a zero-length data packet. This register will then be cleared by firmware.

Table 321. Receive Zero-length Data Packet Register Byte 0

Bits	Type	Name	Description	Default
7	R/W	rx0byte_ep7	Endpoint 7 receives a zero-length data packet.	0x0
6	R/W	rx0byte_ep6	Endpoint 6 receives a zero-length data packet.	0x0
5	R/W	rx0byte_ep5	Endpoint 5 receives a zero-length data packet.	0x0
4	R/W	rx0byte_ep4	Endpoint 4 receives a zero-length data packet.	0x0
3	R/W	rx0byte_ep3	Endpoint 3 receives a zero-length data packet.	0x0
2	R/W	rx0byte_ep2	Endpoint 2 receives a zero-length data packet.	0x0
1	R/W	rx0byte_ep1	Endpoint 1 receives a zero-length data packet.	0x0

3.22.3.10 Receive Zero-length Data Packet Register Byte 1

Address: 0x1A

Table 322. Receive Zero-length Data Packet Register Byte 1

Bits	Type	Name	Description	Default
0	R/W	rx0byte_ep8	Endpoint 8 receives a zero-length data packet.	0x0

3.22.3.11 FIFO Empty Byte 0

Address: 0x1C

By polling this register, firmware can know whether the FIFO is fully empty.

Table 323. FIFO Empty Byte 0

Bits	Type	Name	Description	Default
7	R	fempt_f7	1: FIFO 7 is fully empty 0: FIFO 7 is not fully empty	0x1
6	R	fempt_f6	1: FIFO 6 is fully empty 0: FIFO 6 is not fully empty	0x1

5	R	fempt_f5	1: FIFO 5 is fully empty 0: FIFO 5 is not fully empty	0x1
4	R	fempt_f4	1: FIFO 4 is fully empty 0: FIFO 4 is not fully empty	0x1
3	R	fempt_f3	1: FIFO 3 is fully empty 0: FIFO 3 is not fully empty	0x1
2	R	fempt_f2	1: FIFO 2 is fully empty 0: FIFO 2 is not fully empty	0x1
1	R	fempt_f1	1: FIFO 1 is fully empty 0: FIFO 1 is not fully empty	0x1
0	R	fempt_f0	1: FIFO 0 is fully empty 0: FIFO 0 is not fully empty	0x1

3.22.3.12 FIFO Empty Byte 1

Address: 0x1D

Table 324. FIFO Empty Byte 1

Bits	Type	Name	Description	Default
7	R	fempt_f15	1: FIFO 15 is fully empty 0: FIFO 15 is not fully empty	0x1
6	R	fempt_f14	1: FIFO 14 is fully empty 0: FIFO 14 is not fully empty	0x1

3.22.3.13 Initial Value of Random Pattern

Address: 0x1E

Table 325. Initial Value of Random Pattern

Bits	Type	Name	Description	Default
7:0	R/W	TST_INF_INI	The controller can generate different test-pattern, of random sequence depending on the different initial value, to test interface between PHY and controller when the bit 'TST_INF' is active.	0x00

3.22.3.14 Byte Count of Random Pattern

Address: 0x1F

Table 326. Byte Count of Random Pattern

Bits	Type	Name	Description	Default
7:0	R/W	TST_CONT	The byte count of the random test-patterns which test interface between PHY and controller	0xFF

			is dependent on the value of TST_CONT.	

3.22.4 Interrupt Source Register

3.22.4.1 Interrupt Group Register

Address: 0x20

The returned value is the raw status when micro-controller reads this register.

Table 327. Interrupt Group Register

Bits	Type	Name	Description	Default
7	R	INT_SCR7	Indicate the occurrence of some interrupts in “Interrupt Source Register Byte 7”.	0x1
6	R	INT_SCR6	Indicate the occurrence of some interrupts in “Interrupt Source Register Byte 6”.	0x0
5	R	INT_SCR5	Indicate the occurrence of some interrupts in “Interrupt Source Register Byte 5”.	0x0
4	R	INT_SCR4	Indicate the occurrence of some interrupts in “Interrupt Source Register Byte 4”.	0x0
3	R	INT_SCR3	Indicate the occurrence of some interrupts in “Interrupt Source Register Byte 3”.	0x0
2	R	INT_SCR2	Indicate the occurrence of some interrupts in “Interrupt Source Register Byte 2”.	0x0
1	R	INT_SCR1	Indicate the occurrence of some interrupts in “Interrupt Source Register Byte 1”.	0x0
0	R	INT_SCR0	Indicate the occurrence of some interrupts in “Interrupt Source Register Byte 0”.	0x0

3.22.4.2 Interrupt Source Register Byte 0

Address: 0x21

The returned value is the raw status when micro-controller reads this register.

Table 328. Interrupt Source Register Byte 0

Bits	Type	Name	Description	Default
7	R/W	CX_COMABT_INT	It indicates a command abort event has occurred. For interrupts recoded in this source register, a command abort interrupt receives the highest priority. For a command abort interrupt, the AP should only clear the CX_COMABT_INT bit, and all other operations are unnecessary and should be	0x0

			avoided. In general, the command abort interrupt will be accompanied by a CX_SETUP_INT. The AP should service the command abort interrupt first in order to clear the CX_COMABT_INT. This is done because the CXF FIFO is frozen for AP access when the CX_COMABT_INT remains at '1'. In order to get the 8-byte for SETUP which leads to command an abort, the AP should clear CX_COMABT_INT first.	
5	R/W	RBUF_ERR	Indicate the AP reads an empty FIFO. This bit should be cleared by firmware.	0x0
4	R	CX_COMFAIL_INT	It indicates the control transfer has terminated abnormally. This bit will be asserted when controller receives an extra IN / OUT token at the control transfer's data stage. Once this bit is asserted, it will be kept at '1' before the AP sets the CX_Config_Status register's CX_STL bit. After setting the register's CX_STL bit, the AP should set the register's CX_DONE bit.	0x0
3	R	CX_COMEND_INT	It indicates the control transfer has entered the status stage. This bit will remain asserted before the firmware sets the CX Configuration and Status Register's (Address 0B, bit 0) CX_DONE bit. This bit will remain unchanged after the AP sets the CX_Config_Status register's CX_STL bit.	0x0
2	R	CX_OUT_INT	It indicates the control transfer contains valid data for control-write transfers. This bit will remain asserted until the firmware starts to read the data from the controller's control transfer FIFO (CXF). This bit will be cleared after the AP sets the CX_Config_Status register's CX_STL bit.	0x0
1	R	CX_IN_INT	It indicates the firmware should write data for the control-read transfer to the control transfer FIFO. For a control-read with length less than, or equal to, 64-byte, this bit will never be asserted. The firmware will decode the 8-byte data sent in control transfer's SETUP stage. The firmware should write the first data payload into the control transfer FIFO if the 8-byte represents the control-read transfer without assertion of this bit. This bit will be asserted only when the length of control-read transfer is longer than 64-byte and USB host has successfully received the data of previous packet.	0x0

			For example: for a 65-byte control-read transfer, the firmware should automatically write the first 64-byte after it decodes the 8 bytes of SETUP data. The firmware will be interrupted to write the 65th byte when the USB host ACKs to the first 64-byte. This bit will remain asserted until firmware starts to write data into the controller's control transfer FIFO (CXF). This bit will be cleared after the AP sets the CX_STL bit of CX_Config_Status register.	
0	R	CX_SETUP_INT	This bit will remain asserted until the firmware starts to read data from the controller's control transfer FIFO (CXF). This bit will remain unchanged after the AP sets the CX_Config_Status register's CX_STL bit.	0x0

3.22.4.3 Interrupt Source Register Byte 1

Address: 0x22

The returned value is the raw status when micro-controller reads this register. In the case of a ping-pong FIFO, the firmware has only to take care of the status of the "first" FIFO. For example, if FIFO 0, FIFO 1 and FIFO 2 are ping-ponging for OUT in endpoint 1, the firmware only needs to take care of the status for FIFO 0.

Table 329. Interrupt Source Register Byte 1

Bits	Type	Name	Description	Default
7	R	F3_SPK_INT	This bit becomes 1 when short packet data are received in FIFO 3. This bit is cleared once AHB master reads FIFO 3.	0x0
6	R	F3_OUT_INT	This bit becomes 1 when FIFO3 is ready to be read. This bit is cleared when all data in FIFO 3 are read out.	0x0
5	R	F2_SPK_INT	This bit becomes 1 when short packet data are received in FIFO 2. This bit is cleared once AHB master reads FIFO 2.	0x0
4	R	F2_OUT_INT	This bit becomes 1 when FIFO 2 is ready to be read. This bit is cleared when all data in FIFO 2 are read out.	0x0
3	R	F1_SPK_INT	This bit becomes 1 when short packet data are received in FIFO 1. This bit is cleared once AHB master reads FIFO 1.	0x0
2	R	F1_OUT_INT	This bit is cleared when all data in FIFO1 are	0x0

			read out. This bit becomes 1 when FIFO 1 is ready to be read.	
1	R	F0_SPK_INT	This bit becomes 1 when short packet data are received in FIFO 0. This bit is cleared once AHB master reads FIFO 0.	0x0
0	R	F0_OUT_IN	This bit becomes 1 when FIFO 0 is ready to be read. This bit is cleared when all data in FIFO 0 are read out.	0x0

3.22.4.4 Interrupt Source Register Byte 2

Address: 0x23

The returned value is the raw status when micro-controller reads this register.

Table 330. Interrupt Source Register Byte 2

Bits	Type	Name	Description	Default
7	R	F7_SPK_INT	This bit becomes 1 when short packet data are received in FIFO 7. This bit is cleared once AHB master reads FIFO 7.	0x0
6	R	F7_OUT_INT	This bit becomes 1 when FIFO 7 is ready to be read. This bit is cleared when all data in FIFO 7 are read out.	0x0
5	R	F6_SPK_INT	This bit becomes 1 when short packet data are received in FIFO 6. This bit is cleared once AHB master reads FIFO 6.	0x0
4	R	F6_OUT_INT	This bit becomes 1 when FIFO 6 is ready to be read. This bit is cleared when all data in FIFO 6 are read out.	0x0
3	R	F5_SPK_INT	This bit becomes 1 when short packet data are received in FIFO 5. This bit is cleared once AHB master reads FIFO 5.	0x0
2	R	F5_OUT_INT	This bit is cleared when all data in FIFO5 are read out. This bit becomes 1 when FIFO 5 is ready to be read.	0x0
1	R	F4_SPK_INT	This bit becomes 1 when short packet data are received in FIFO 4. This bit is cleared once AHB master reads FIFO	0x0

			4.	
0	R	F4_OUT_IN	This bit becomes 1 when FIFO 4 is ready to be read. This bit is cleared when all data in FIFO 4 are read out.	0x0

3.22.4.5 Interrupt Source Register Byte 4

Address: 0x25

The returned value is the raw status when micro-controller reads this register.

Table 331. Interrupt Source Register Byte 4

Bits	Type	Name	Description	Default
7	R	F15_SPK_INT	This bit becomes 1 when short packet data are received in FIFO 15. This bit is cleared once AHB master reads FIFO 15.	0x0
6	R	F15_OUT_INT	This bit becomes 1 when FIFO 15 is ready to be read. This bit is cleared when all data in FIFO 15 are read out.	0x0
5	R	F14_SPK_INT	This bit becomes 1 when short packet data are received in FIFO 14. This bit is cleared once AHB master reads FIFO 14.	0x0
4	R	F14_OUT_INT	This bit becomes 1 when FIFO 14 is ready to be read. This bit is cleared when all data in FIFO 14 are read out.	0x0

3.22.4.6 Interrupt Source Register Byte 5

Address: 0x26

The returned value is the raw status instead of the masked status when micro-controller reads this register. In the case of a ping-pong FIFO, the firmware has only to take care of the status of the “first” FIFO. For example, if FIFO 0, FIFO 1 and FIFO 2 are ping-ponging for the IN at endpoint 1, the firmware only needs to take care of the status for FIFO 0.

Table 332. Interrupt Source Register Byte 5

Bits	Type	Name	Description	Default
7	R	F7_ IN_INT	**This bit becomes 1 to indicate FIFO 7 is ready to be written.** This bit can be cleared under the following three conditions:	0x0

			1. A maximum-size packet is received in FIFO 7. 2. AHB master ends the transfer with the signal dma_termwr. 3. CPU sets the Done bit of register 0xA7.	
6	R	F6_ IN_INT	**This bit becomes 1 to indicate FIFO 6 is ready to be written.** This bit can be cleared under the following three conditions: 1. A maximum-size packet is received in FIFO 6. 2. AHB master ends the transfer with the signal dma_termwr. 3. CPU sets the Done bit of register 0xA6.	0x0
5	R	F5_ IN_INT	**This bit becomes 1 to indicate FIFO 5 is ready to be written.** This bit can be cleared under the following three conditions: 1. A maximum-size packet is received in FIFO 5. 2. AHB master ends the transfer with the signal dma_termwr. 3. CPU sets the Done bit of register 0xA5.	0x0
4	R	F4_ IN_INT	**This bit becomes 1 to indicate FIFO 4 is ready to be written.** This bit can be cleared under the following three conditions: 1. A maximum-size packet is received in FIFO 4. 2. AHB master ends the transfer with the signal dma_termwr. 3. CPU sets the Done bit of register 0xA4.	0x0
3	R	F3_ IN_INT	**This bit becomes 1 to indicate FIFO 3 is ready to be written.** This bit can be cleared under the following three conditions: 1. A maximum-size packet is received in FIFO 3. 2. AHB master ends the transfer with the signal dma_termwr. 3. CPU sets the Done bit of register 0xA3.	0x0
2	R	F2_ IN_INT	**This bit becomes 1 to indicate FIFO 2 is ready to be written.** This bit can be cleared under the following three conditions: 1. A maximum-size packet is received in FIFO 2.	0x0

			2. AHB master ends the transfer with the signal dma_termwr. 3. CPU sets the Done bit of register 0xA2.	
1	R	F1_ IN_INT	**This bit becomes 1 to indicate FIFO 1 is ready to be written.** This bit can be cleared under the following three conditions: 1. A maximum-size packet is received in FIFO 1. 2. AHB master ends the transfer with the signal dma_termwr. 3. CPU sets the Done bit of register 0xA1.	0x0
0	R	F0_ IN_INT	**This bit becomes 1 to indicate FIFO 0 is ready to be written.** This bit can be cleared under the following three conditions: 1. A maximum-size packet is received in FIFO 0. 2. AHB master ends the transfer with the signal dma_termwr. 3. CPU sets the Done bit of register 0xA0.	0x0

3.22.4.7 Interrupt Source Register Byte 6

Address: 0x27

The returned value is the raw status when micro-controller reads this register.

Table 333. Interrupt Source Register Byte

Bits	Type	Name	Description	Default
7	R	F15_ IN_INT	**This bit becomes 1 to indicate FIFO 15 is ready to be written.** This bit can be cleared under the following three conditions: 1. A maximum-size packet is received in FIFO 15. 2. AHB master ends the transfer with the signal dma_termwr. 3. CPU sets the Done bit of register 0xAF.	0x0
6	R	F14_ IN_INT	**This bit becomes 1 to indicate FIFO 14 is ready to be written.** This bit can be cleared under the following three conditions: 1. A maximum-size packet is received in FIFO 14. 2. AHB master ends the transfer with the	0x0

			signal dma_termwr. 3. CPU sets the Done bit of register 0xAE.	

3.22.4.8 Interrupt Source Register Byte 7

Address: 0x28

When a bus reset occurs, the bits [3:1] will be reset to 0x1.

Please note that when the TX0BYTE_IEPx bit is applied to an IN endpoint, the AP should not send the next packet to the endpoint until the TX0BYTE_INT interrupt occurs. The returned value is the raw status when micro-controller reads this register.

Table 334. Interrupt Source Register Byte 7

Bits	Type	Name	Description	Default
7	R/W	RX0BYTE_INT	**Received Zero-length Data Packet Interrupt** The controller receives a zero-length data packet from the USB host. When the controller receives a zero-length data packet from the USB host, this bit will be set. The firmware may further check registers 0x19 and 0x1A to determine which endpoint received a zero-length data packet from the USB host. When the interrupt occurs, the controller will NAK the next OUT transaction to the same endpoint until the corresponding bit (in 0x19 or 0x1A) is cleared by firmware. This bit is not affected by a USB bus reset.	0x0
6	R/W	TX0BYTE_INT	**Transferred Zero-length Data Packet Interrupt** The controller returned a zero-length data packet to the USB host. This bit will be set under the following two cases: 1. When the USB host issues an IN transaction to an isochronous end-point while the controller is not ready to return data, the controller will transfer a zero-length data packet to the USB host. In such a case, this bit will be used. 2. When the TX0BYTE_IEPx bit is set, and after the endpoint's data in the FIFO are transferred, the controller will re-turn a zero-length data packet to the next IN transaction to the same endpoint. The firmware may further check the registers	0x0

			2DH and 2EH to determine which endpoint returns a zero-length data packet to the USB host. After the AP has serviced the interrupt request, this bit must be cleared by firmware. This bit is not affected by the USB bus reset.	
5	R/W	ISO_SEQ_ABORT_INT	**ISO Sequential Abort Interrupt** A high bandwidth isochronous sequential abort. When the controller detects an incomplete DATA PID sequence during a micro frame, this bit will be set. For example, if the controller detects an MDATA followed by an SOF, this is taken as a sequential abort. The firmware should further check the registers 2BH and 2CH to determine which endpoint received an isochronous sequential abort. After the AP has serviced the interrupt request, this bit must be cleared by firmware. This bit is not affected by a USB bus reset.	0x0
4	R/W	ISO_SEQ_ERR_INT	**ISO Sequential Error Interrupt** High bandwidth isochronous sequential error. When the controller detects a DATA PID sequence error in an isochronous transaction in high bandwidth, this bit will be set. Any out of order se-quence will be taken as a sequence error. The firmware should further check the registers 29H and 2AH to determine which endpoint received an isochronous sequential error. After the AP has serviced the interrupt re-quest, this bit must be cleared by firmware. This bit is not affected by a USB bus reset.	0x0
3	R/W	RESM_INT	**Resume Interrupt** Resume-state-change interrupt bit. When the controller detects a resume event from the host, this bit will be set. After the AP has serviced the interrupt request, this bit must be cleared by firmware. When a USB bus reset occurs, it will also be cleared.	0x0
2	R/W	SUSP_INT	**Suspend Interrupt** Suspend-state-change interrupt bit. When the USB bus remains in an idle state for over 3ms, this bit will be set. This bit must be cleared before firmware sets the “GOSUSP” in register 00H. This bit will also be cleared when the USB bus resets or a resume occurs.	0x0

1	R/W	USBRST_INT	**USB Reset Interrupt** Bus reset interrupt bit. When the controller detects a USB bus reset from the host, the bit will be set. When the AP has serviced the interrupt request, this bit must be cleared by firmware.	0x0
0	R	HBF_EMPTY_INT	**HBF Empty Interrupt** Before sending data to the controller, AP should make sure that the HBF is empty and the FIFO is not full. Please note that the value of this bit is not masked during CPU reads.	0x1

3.22.4.9 Isochronous Sequential Error Register Byte 0

Address: 0x29

When an Isochronous Sequential Error Interrupt occurs (bit 4 of register 0x28), the firmware should further check registers 0x29 and 0x2A to determine which endpoint received an isochronous sequential error. This bit should be cleared by firmware

Table 335. Isochronous Sequential Error Register Byte 0

Bits	Type	Name	Description	Default
7	R/W	iso_seq_err_ep7	Endpoint 7 receives an isochronous sequential error.	0x0
6	R/W	iso_seq_err_ep6	Endpoint 6 receives an isochronous sequential error.	0x0
5	R/W	iso_seq_err_ep5	Endpoint 5 receives an isochronous sequential error.	0x0
4	R/W	iso_seq_err_ep4	Endpoint 4 receives an isochronous sequential error.	0x0
3	R/W	iso_seq_err_ep3	Endpoint 3 receives an isochronous sequential error.	0x0
2	R/W	iso_seq_err_ep2	Endpoint 2 receives an isochronous sequential error.	0x0
1	R/W	iso_seq_err_ep1	Endpoint 1 receives an isochronous sequential error.	0x0

3.22.4.10 Isochronous Sequential Error Register Byte 1

Address: 0x2A

Table 336. Isochronous Sequential Error Register Byte 1

Bits	Type	Name	Description	Default
0	R/W	iso_seq_err_ep8	Endpoint 8 receives an isochronous sequential error.	0x0

3.22.4.11 Isochronous Sequential Abort Register Byte 0

Address: 0x2B

When an Isochronous Sequential Abort Interrupt occurs (bit 5 of register 0x28), the firmware should further check registers 2BH and 2CH to determine which endpoint received an isochronous sequential abort. This register should be cleared by firmware.

Table 337. Isochronous Sequential Abort Register Byte 0

Bits	Type	Name	Description	Default
7	R/W	iso_seq_abt_ep7	Endpoint 7 receives an isochronous sequential abort.	0x0
6	R/W	iso_seq_abt_ep6	Endpoint 6 receives an isochronous sequential abort.	0x0
5	R/W	iso_seq_abt_ep5	Endpoint 5 receives an isochronous sequential abort.	0x0
4	R/W	iso_seq_abt_ep4	Endpoint 4 receives an isochronous sequential abort.	0x0
3	R/W	iso_seq_abt_ep3	Endpoint 3 receives an isochronous sequential abort.	0x0
2	R/W	iso_seq_abt_ep2	Endpoint 2 receives an isochronous sequential abort.	0x0
1	R/W	iso_seq_abt_ep1	Endpoint 1 receives an isochronous sequential abort.	0x0

3.22.4.12 Isochronous Sequential abort Register Byte 1

Address: 0x2C

Table 338. Isochronous Sequential abort Register Byte 1

Bits	Type	Name	Description	Default
0	R/W	iso_seq_abt_ep8	Endpoint 8 receives an isochronous sequential abort.	0x0

3.22.4.13 Transferred Zero-length Register Byte 0

Address: 0x2D

When a Transferred Zero-length Data Packet Interrupt occurs (bit 6 of register 0x28), the firmware may further check registers 0x2D and 0x2E to determine which endpoint returns a zero-length data packet to the USB host.

This register should be cleared by firmware.

Table 339. Transferred Zero-length Register Byte 0

Bits	Type	Name	Description	Default
7	R/W	tx0byte_ep7	Endpoint 7 transfers a zero-length data packet.	0x0
6	R/W	tx0byte_ep6	Endpoint 6 transfers a zero-length data packet.	0x0
5	R/W	tx0byte_ep5	Endpoint 5 transfers a zero-length data packet.	0x0
4	R/W	tx0byte_ep4	Endpoint 4 transfers a zero-length data packet.	0x0
3	R/W	tx0byte_ep3	Endpoint 3 transfers a zero-length data packet.	0x0
2	R/W	tx0byte_ep2	Endpoint 2 transfers a zero-length data packet.	0x0
1	R/W	tx0byte_ep1	Endpoint 1 transfers a zero-length data packet.	0x0

3.22.4.14 Transferred Zero-length Register Byte 1

Address: 0x2E

Table 340. Transferred Zero-length Register Byte 1

Bits	Type	Name	Description	Default
0	R/W	tx0byte_ep8	Endpoint 8 transfers a zero-length data packet.	0x0

3.22.5 Miscellaneous Register

3.22.5.1 Idle Counter

Address: 0x2F

Table 341. Idle Counter

Bits	Type	Name	Description	Default
2:0	R/W	IDLE_CNT	It controls the timing delay from the time indicated in the GOSUSP bit of the main control register to the time the controller enters a suspend mode. The delay is de-noted as tsusp_delay in the following figure for reference. 0: tsusp_delay = 0ms. 1: tsusp_delay = 1ms. 2: tsusp_delay = 2ms. 4: tsusp_delay = 3ms. 4: tsusp_delay = 4ms. 5: tsusp_delay = 5ms 6: tsusp_delay = 6ms. 7: tsusp_delay = 7ms. Note: The USB 2.0 specifications define TSUSP to mandate that the device should enter the suspend mode no later than 10ms after the D+/D- is continuously in an idle state. The firmware programmer should be cautious in programming the value of tsusp_delay.	0x0

3.22.6 Endpoint Configuration and Status Register

3.22.6.1 Endpoint x Map Register

Address: 0x30-0x37 (One per Endpoint, x = 1~8)

This register records the mapped FIFO number of each non-control-transfer 0 endpoint.

Table 342. Endpoint x Map Register

Bits	Type	Name	Description	Default
7:4	R/W	FNO_OEPx [3:0]	**FIFO Number for OUT Endpoint x:** Record the physical FIFO number for logical out endpoint x.	0xF
3:0	R/W	FNO_IEPx [3:0]	**FIFO Number for IN Endpoint x:** Record the physical FIFO number for logical in endpoint x.	0xF

The table below shows the Endpoint x Map Register offset for each endpoint.

Table 343. Endpoint x Map Register offset for each endpoint

Endpoint Number	Endpoint x Map Register Offset (Hex)
Endpoint 1	30
Endpoint 2	31
Endpoint 3	32
Endpoint 4	33
Endpoint 5	34
Endpoint 6	35
Endpoint 7	36
Endpoint 8	37

3.22.6.2 HBF Data Byte Count

Address: 0x3F

This register contains byte count information for HBF.

Table 344. HBF Data Byte Count

Bits	Type	Name	Description	Default
4:0	R	HBF_CNT[4:0]	**HBF Data Byte Count:** Record the data byte count in HBF.	0x0

3.22.6.3 IN Endpoint x MaxPacketSize Register Low Byte

Address: (40+2(x-1)) (One per Endpoint, x = 1~8)

Table 345. IN Endpoint x MaxPacketSize Register Low Byte

Bits	Type	Name	Description	Default
7:0	R/W	MAXPS_IEPx [7:0]	**Max Packet Size of IN Endpoint x:** Maximum packet size bits [7:0] of endpoint x which is capable of sending or receiving data smaller than or equal to this size.	0x00

3.22.6.4 IN Endpoint x MaxPacketSize Register High Byte

Address: (41+2(x-1)) (One per Endpoint, x = 1~8)

Table 346. IN Endpoint x MaxPacketSize Register High Byte

Bits	Type	Name	Description	Default
7	R/W	TX0BYTE_IEPx	Transfer a Zero-length Data Packet from Endpoint x to USB Host This bit should be set after the last transaction packet is sent to the FIFO by AHB master. After the FIFO's endpoint's data are transferred, the controller will re-turn a zero-length data packet to the next IN transaction to the same endpoint. The AP should not send the next packet to the same endpoint until TX0BYTE_INT interrupt for the endpoint occurs. This bit is cleared by hardware automatically when TX0BYTE_INT occurs.	0x0
6:5	R/W	TX_NUM_HBW_IEPx	Transaction Number for High bandwidth Endpoint x TX_NUM_HBW[1:0]: While using three transactions per micro frame, the mapped FIFO must be set as triple blocks (See FIFO configuration register for details). While using two transactions per micro frame, the mapped FIFO must be set as double blocks.	0x0
5	R/W	RSTG_IEPx	Reset Toggle Sequence for IN Endpoint x: Firmware resets the toggle bit of indexed endpoint x by writing a '1' to this bit. This bit should also be cleared by firmware.	0x0
4	R/W	STL_IEPx	Stall IN Endpoint x: The indexed endpoint x can be stalled by writing a '1' to this bit. The stall status of the indexed endpoint x can be cleared by writing a '0' to this bit.	0x0
2:0	R/W	MAXPS_IEPx [10:8]	**Max Packet Size of IN Endpoint x:** Maximum packet size bits [10:8] of endpoint x	0x2

			which is capable of sending or receiving data smaller than or equal to this size.	

The table below shows the IN Endpoint x MaxPacketSize Register Low and High Byte offset for each endpoint.

Table 347. IN Endpoint x MaxPacketSize Register Low and High Byte offset for each endpoint

Endpoint Number	High Byte Offset (Hex)	Low Byte Offset (Hex)
Endpoint 1	41	40
Endpoint 2	43	42
Endpoint 3	45	44
Endpoint 4	47	46
Endpoint 5	49	48
Endpoint 6	4B	4A
Endpoint 7	4D	4C
Endpoint 8	4F	4E

3.22.6.5 OUT Endpoint x MaxPacketSize Register Low Byte

Address: (60+2(x-1)) (One per Endpoint, x = 1~8)

Table 348. OUT Endpoint x MaxPacketSize Register Low Byte

Bits	Type	Name	Description	Default
7:0	R/W	MAXPS_OEPx [7:0]	**Max Packet Size of OUT Endpoint x:** Maximum packet size bits [7:0] of endpoint x which is capable of sending or receiving data smaller than or equal to this size.	0x00

3.22.6.6 OUT Endpoint x MaxPacketSize Register High Byte

Address: (61+2(x-1)) (One per Endpoint, x = 1~8)

Table 349. OUT Endpoint x MaxPacketSize Register High Byte

Bits	Type	Name	Description	Default
4	R/W	RSTG_OEPx	**Reset Toggle Sequence for OUT Endpoint x**: Firmware resets the indexed endpoint x's toggle bit by writing a '1' to this bit. This bit should also be cleared by firmware.	0x0
3	R/W	STL_OEPx	**Stall OUT Endpoint x**: Writing a '1' to STL_OEP will stall endpoint x. The stall status of the indexed endpoint x can be cleared by writing a '0' to this bit. (Note: after setting this bit to 1, AP still needs to monitor OUT interrupt of this endpoint. If OUT interrupt as-serts, AP must read out data in the OUT FIFO even AP doesn't need this data)	0x0

2:0	R/W	MAXPS_OEPx [10:8]	**Max Packet Size of OUT Endpoint x**: Maximum packet size bits [10:8] of endpoint x that is capable of sending or receiving data smaller than or equal to this size.	0x2

The table below shows the OUT Endpoint x MaxPacketSize Register Low and High Byte offset for each endpoint.

Table 350. OUT Endpoint x MaxPacketSize Register Low / High Byte offset for each endpoint

Endpoint Number	High Byte Offset (Hex)	Low Byte Offset (Hex)
Endpoint 1	61	60
Endpoint 2	63	62
Endpoint 3	65	66
Endpoint 4	67	66
Endpoint 5	6B	68
Endpoint 6	6D	6A
Endpoint 7	71	6C
Endpoint 8	69	6E

3.22.6.7 DMA Mode Enable Register Low Byte

Address: 0x7E

This register allows the controller to assert fusb220_dma_req_r according to the enabled FIFO. Only one bit should be set as '1' between 0x7E and 0x7F at any time. This setting informs controller that FIFO is currently being serviced by the DMA controller.

For example, assume the FIFO 0 and FIFO 1 are ping-ponging for IN endpoint 1. Only bit 0 of 0x7E should be set by firmware. The fusb220_dma_req_r is asserted when either the FIFO 0 or FIFO 1 is not full. If endpoint 1 is for OUT, the fusb220_dma_req_r is asserted when either the FIFO 0 or FIFO 1 is not empty.

Table 351. DMA Mode Enable Register Low Byte

Bits	Type	Name	Description	Default
7	R/W	fifo7_dma_en	FIFO 7 DMA Enable	0x0
6	R/W	fifo6_dma_en	FIFO 6 DMA Enable	0x0
5	R/W	fifo5_dma_en	FIFO 5 DMA Enable	0x0
4	R/W	fifo4_dma_en	FIFO 4 DMA Enable	0x0
3	R/W	fifo3_dma_en	FIFO 3 DMA Enable	0x0
2	R/W	fifo2_dma_en	FIFO 2 DMA Enable	0x0
1	R/W	fifo1_dma_en	FIFO 1 DMA Enable	0x0
0	R/W	fifo0_dma_en	FIFO 0 DMA Enable	0x0

3.22.6.8 DMA Mode Enable Register High Byte

Address: 0x7F

Please refer to Section above for detailed information.

Table 352. DMA Mode Enable Register High Byte

Bits	Type	Name	Description	Default
7	R/W	Fifo15_dma_en	FIFO 15 DMA Enable	0x0
6	R/W	Fifo14_dma_en	FIFO 14 DMA Enable	0x0

3.22.7 FIFO Configuration and Status Register

3.22.7.1 FIFOx Map Register

Address: 0x80-0x8F (One per FIFO, x = 0-7, 14-15)

Table 353. FIFOx Map Register

Bits	Type	Name	Description	Default
4	R/W	Dir_Fx	**FIFO Direction:** Data transfer direction. 0 = OUT, 1= IN.	0x0
3:0	R/W	EP_FIFOx [3:0]	**Endpoint Number for FIFO x:** Record the physical endpoint number for physical FIFO x.	0xf

The table below shows the FIFOx Map Register offset for each FIFO.

Table 354. FIFOx Map Register offset for each FIFO

FIFO Number	Offset (Hex)	FIFO Number	Offset (Hex)
FIFO 0	80		
FIFO 1	81		
FIFO 2	83		
FIFO 3	82		
FIFO 4	84		
FIFO 5	85		
FIFO 6	86	FIFO 14	8E
FIFO 7	87	FIFO 15	8F

3.22.7.2 FIFOx Configuration Register

Address: 0x90-0x9F (One per FIFO, x = 0-7, 14-15)

The PAM is configured to ten (10) FIFOs that are numbered from 0 to 7, 14, and 15. FIFO 0 to FIFO 7 are 512-byte, while FIFO 14 and FIFO 15 are 64-byte each.

The ping-pong FIFO mechanism is block-based. For example, if bit 4 of the 0x90 is set as '1' and bits [3:2] are set as 0x2, when FIFO0 is accessed, three 1024-byte blocks (FIFO 0 combined with FIFO 1, FIFO 2 combined with FIFO 3, and FIFO 4 combined with FIFO 5) would be ping-pong in turn.

A “block” can be composed of one FIFO or two FIFOs, depending on the setting of the bit BLKSZ_Fx. When this bit is set as ‘1’, the block size is 1024-byte; when this bit is set as ‘0’, the block size is 512-byte. For example, if 0x90 bit 4 is set as ‘1’, it means FIFO0 and FIFO1 are combined just like a 1024-byte FIFO, otherwise known as a “block”.

Table 355. FIFOx Configuration Register

Bits	Type	Name	Description	Default
7	R/W	EN_Fx	**Enable FIFO x:** A ‘1’ indicates that the FIFO is enabled.	0x0
4	R/W	BLKSZ_Fx	**Block Size of FIFO x:** BLKSIZE_Fx = 0 FIFO 0 ~ FIFO 13 : For transferring packets whose maximum packet size is smaller than or equal to 512 bytes FIFO 14 & FIFO 15 : For transferring packets whose maximum packet size is smaller than or equal to 64 bytes BLKSIZE_Fx = 1 FIFO 0 ~ FIFO 7 : For transferring packets whose maximum packet size is smaller than or equal to 1024 bytes and greater than 512 bytes FIFO 14 and FIFO 15 : For transferring packets whose maximum packet size is smaller than or equal to 128 bytes and greater than 64 bytes	0x0
3:2	R/W	BLKNO_Fx [1:0]	**Block Number of FIFO x:** BLKNUM_Fx = 0 : Single block BLKNUM_Fx = 1 : Double blocks BLKNUM_Fx = 2 : Triple blocks BLKNUM_Fx = 3 : Reserved	0x0
1:0	R/W	BLK_TYP_Fx [1:0]	**Transfer Type of FIFO x:** Indicates the transfer type used for FIFOx transfer. TYP_Fx = 0 : Reserved TYP_Fx = 1 : Isochronous type TYP_Fx = 2 : Bulk type TYP_Fx = 3 : Interrupt type	0x0

The table below shows the FIFOx Configuration Register offset for each FIFO.

Table 356. FIFOx Configuration Register offset for each FIFO

FIFO Number	Offset (Hex)	FIFO Number	Offset (Hex)
FIFO 0	90		
FIFO 1	91		
FIFO 2	92		
FIFO 3	93		
FIFO 4	94		
FIFO 5	95		
FIFO 6	97	FIFO 14	9E
FIFO 7	96	FIFO 15	9F

3.22.7.3 FIFOx Instruction Register

Address: (Address = 0xA0-0xAF) (One per FIFO, x = 0-7, 14-15)

Table 357. FIFOx Instruction Register

Bits	Type	Name	Description	Default
4	R/W	FFRST	**FIFO x Reset:** FIFO can be reset by firmware through setting this bit. This bit must be cleared by firmware after resetting FIFO.	0x0
3	R/W	DONE_Fx	**Data Transfer is Done for IN FIFO x:** The firmware must set this bit to inform the HBS the whole transaction is completed.	0x0
2:0	R	BC_Fx [10:8]	**OUT FIFO x Byte Count:** BC_Fx [10:0] indicates the byte number of data stored in the FIFO for OUT EPx.	0x0

The table below shows the FIFOx Instruction Register offset for each FIFO.

Table 358. FIFOx Instruction Register offset for each FIFO

FIFO Number	Offset (Hex)	FIFO Number	Offset (Hex)
FIFO 0	A0		
FIFO 1	A1		
FIFO 2	A2		
FIFO 3	A4		
FIFO 4	A3		
FIFO 5	A5		
FIFO 6	A6	FIFO 14	AE
FIFO 7	A7	FIFO 15	AF

3.22.7.4 FIFOx Byte-Count Register Low Byte

Address: 0xB0-0xBF (One per FIFO, x = 0-7, 14-15)

The value of the byte-count register is not valid after hardware reset. The AP should check it only when OUT interrupts are issued.

Table 359. FIFOx Byte-Count Register Low Byte

Bits	Type	Name	Description	Default
7:0	R	BC_Fx [7:0]	**OUT FIFO x Byte Count:** BC_Fx [10:0] indicates the byte number of data stored in the FIFO for OUT EPx. BC_Fx[10:8] is deposited at above registers, 0xA0~0xAF.	0xff

The table below shows the FIFOx Byte-Count Register Low Byte offset for each FIFO.

Table 360. FIFOx Byte-Count Register Low Byte offset for each FIFO

FIFO Number	Offset (Hex)	FIFO Number	Offset (Hex)
FIFO 0	B1		
FIFO 1	B2		
FIFO 2	B3		
FIFO 3	B0		
FIFO 4	B4		
FIFO 5	B5		
FIFO 6	B6	FIFO 14	BE
FIFO 7	B7	FIFO 15	BF

3.22.8 Data Port Register

Address: 0xC0 – 0xFC

This address provides a port for an AHB master to access the PAM's FIFO. For example, if an AHB master intends to read data from FIFO 0, it should always issue read cycle with the address C0H. In the case of a ping-pong FIFO, the AHB master should access data via the address of the "first" FIFO. For example, if FIFO 2 and FIFO 3 are ping-ponging for endpoint 2, the AHB master should always access the data of endpoint 2 via address 0xC8 only. It is invalid for AHB master to access endpoint 2 via address 0xC8 in the example.

Table 361. PAM Data Port Register offset for each FIFO

The table below shows the PAM Data Port Register offset for each FIFO.

FIFO Number	Offset (Hex)	FIFO Number	Offset (Hex)
FIFO 0	C0		
FIFO 1	C4		
FIFO 2	C8		
FIFO 3	CC		
FIFO 4	D0		
FIFO 5	D4		
FIFO 6	D8	FIFO 14	F8
FIFO 7	DC	FIFO 15	FC

3.23 Vector Interrupt Controller

3.23.1 Interrupt Raw Status Register

Address: 0x00

Table 362. Interrupt Raw Status Register

Bits	Type	Name	Description	Default
31:0	RO	IntSrc	**Shows the status of the interrupts before masking.**	

			0: Interrupt is non-active. 1: Interrupt is active. **Interrupt Bit Definition.** Bit[0]: Timer#1 Bit[1]: Timer#2 Bit[2]: CPU Frequency Scaling Interrupt Bit[3]: Watch Dog Timer Bit[4]: GPIO Bit[5]: PCI External Interrupt 0 Bit[6]: PCI External Interrupt 1 Bit[7]: PCI Broken Interrupt Bit[8]: PCI Host Bridge Controller Bit[9]: UART0 Bit[10]: UART1 Bit[11]: Generic DMA Terminal Counter Bit[12]: Generic DMA Error Bit[13]: Reserved Bit[14]: RTC Bit[15]: PCM Controller Bit[16]: IDE Host Controller Bit[17]: IDE Device Bit[18]: NIC Controller Bit[19]: NIC DMA TNTC *(To-NIC-Tx-Complete)* Bit[20]: NIC DMA FNRC *(Fm-NIC-Rx-Complete)* Bit[21]: NIC DMA TNQE*(To-NIC-Queue-Empty)* Bit[22]: NIC DMA FNQF *(Fm-NIC-Queue-Full)* Bit[23]: USB 1.1 host Controller Bit[24]: USB 2.0 host Controller Bit[25]: I2S controller Bit[26]: SPI Controller Bit[27]: TWI Controller Bit[28]: vbus state change interrupt Bit[29]: External Interrupt 29 Sharing with pin GPIOA[0] Bit[30]: External Interrupt 30 Sharing with pin GPIOA[1] Bit[31]: HSDMA Terminal Counter and Error Interrupt **Note for VBUS State Change Interrupt** At self power mode, USB device need detect vbus to detect if the device has plug into a USB Host. When VBUS changes from low to high, it means a plug in action. And when VBUS changed from high to low, it means a pull-out action. No matter plug-in or pull-out, if the mask bit is cleared, the interrupt bit will be asserted. CPU should check VBUS state at bit 0 of register 0x20 of Power Management block to know current state. Note: Since USB device plug-in/pull-out will generate bouncing noise. The interrupt pin maybe asserted many times. User may delay for a while to read VBUS pin state, and then clear the interrupt bit.	

3.23.2 Edge Interrupt Source Clear Register

Address: 0x04

Table 363. Edge Interrupt Source Clear Register

Bits	Type	Name	Description	Default
31:0	WC	Intclear	**Clear Edge Interrupt.** Write 0: no action Write 1: clear relative edge interrupt status This clear bit only takes action on edge triggered interrupt sources. Note: Hardware also implements auto-clear scheme when VectAddr is read and the interrupt source is edge triggered. Note: When VIC is not used, but traditional software parsing scheme is used, Edge triggered interrupt should also be cleared manually.	0x00000000

3.23.3 Interrupt Mask Register

Address: 0x08

Table 364. Interrupt Mask Register

Bits	Type	Name	Description	Default
31:0	RW	Intmaskreg	**Interrupt MASK Register.** Read 1: Mask the corresponding bit interrupt source. 0: UnMask the corresponding bit interrupt source. Write 1: Mask the corresponding bit interrupt source. 0: no effect	0xFFFFFFFF

3.23.4 Interrupt Mask Clear Register

Address: 0x0C

Table 365. Interrupt Mask Clear Register

Bits	Type	Name	Description	Default
31:0	WO	IntMaskClr	Clear corresponding bits in the Interrupt Mask Register. 0: no effect 1: mask cleared in Interrupt Mask Register.	

3.23.5 Interrupt Trigger Mode Register

Address: 0x10

Table 366. Interrupt Trigger Mode Register

Bits	Type	Name	Description	Default
31:0	RW	Inttrigmode	**Interrupt Trigger Mode** 0: Level-trigger 1: Edge-trigger Note: If the interrupt mode has been defined by function block, the corresponding bit is READ ONLY.	0x00000000

3.23.6 Interrupt Trigger Level Register

Address: 0x14

Table 367. Interrupt Trigger Level Register

Bits	Type	Name	Description	Default
31:0	RW	Inttriglevel	**Interrupt Trigger Level** 0: High level trigger or rising edge trigger 1: Low level trigger or falling edge trigger Note: If the interrupt mode has been defined by function block, the corresponding bit is READ ONLY.	0x00000000

3.23.7 FIQ Select Register

Address: 0x18

Table 368. FIQ Select Register

Bits	Type	Name	Description	Default
31:0	RW	FiqSelReg	**FIQ Mode Select** 0: Set to IRQ interrupt. 1: Set to FIQ interrupt.	0x00000000

3.23.8 IRQ Status Register

Address: 0x1C

Table 369. IRQ Status Register

Bits	Type	Name	Description	Default
31:0	RO	IrqStatus	**IRQ Status After Mask.** 0: No interrupt 1: Interrupt active	0x00000000

3.23.9 FIQ Status Register

Address: 0x20

Table 370. FIQ Status Register

Bits	Type	Name	Description	Default
31:0	RO	FiqStatus	**FIQ Status After Mask.** 0: No interrupt 1: Interrupt active	0x00000000

3.23.10 Software Interrupt Register

Address: 0x24

Table 371. Software Interrupt Register

Bits	Type	Name	Description	Default
31:0	RW	SoftInt	Setting a bit HIGH generates a software interrupt for the selected source before interrupt masking Read: 0 = software interrupt inactive (reset) 1 = software interrupt active Write: 0 = no effect 1 = software interrupt enabled There is one bit of the register for each interrupt source.	0x00000000

3.23.11 Software Interrupt Clear Register

Address: 0x28

Table 372. Software Interrupt Clear Register

Bits	Type	Name	Description	Default
31:0	WO	SoftIntClear	**Clears corresponding bits in the Software Interrupt Register:** 0 = no effect 1 = software interrupt disabled in the Software Interrupt Register. There is one bit of the register for each interrupt source.	

3.23.12 Software Priority Mask Register

Address: 0x2C

Table 373. Software Priority Mask Register

Bits	Type	Name	Description	Default
7:0	RW	SWPriorityMask	Controls software masking of the 8 interrupt priority levels 0 = interrupt priority level is not masked	0x00

			1 = interrupt priority level is masked. Each bit of the register is applied to each of the 8 interrupt priority levels.	

3.23.13 Power Management Interrupt Register

Address: 0x34

Table 374. Power Management Interrupt Register

Bits	Type	Name	Description	Default
31:0	RW	PwrIntSel	**Select interrupts as SLEEP mode wakeup interrupt sources.** 0 = not selected as sleep mode wakeup interrupt 1 = selected as sleep mode wakeup interrupt There is one bit of the register for each interrupt source.	0x00000000

3.23.14 Vector Address 0 ~ 31 Register

Address: 0x40~0xBF

Table 375. Vector Address 0 ~ 31 Register

Bits	Type	Name	Description	Default
31:0	RW	VectorAddr0~31	Contains ISR vector addresses Note: These registers must only be updated when the relevant interrupts are masked. Note: If the system does not support interrupt vector address, these registers can be programmed with the numbers of the interrupt source ports they relate to, so that the source of the active interrupt can be easily determinedh Note: There registers are without default value. User must program these registers with initial values before enabling VIC.	

3.23.15 Interrupt 0 ~ 31 Priority Register

Address: 0x0C0~0x13F

Table 376. Interrupt 0 ~ 31 Priority Register

Bits	Type	Name	Description	Default
2:0	RW	IntPriority	Selects vectored interrupt priority level. Eight (8) of vectored interrupt priority levels can be selected for each interrupt source. 0 is the highest priority and 7 is the lowest priority. After power on reset, all of interrupts are in the lowest priority.	0x7

3.23.16 IRQ Vector Address Register

Address: 0x140

Table 377. IRQ Vector Address Register

Bits	Type	Name	Description	Default
31:0	RW	VectAddr	Contains the address of the current active ISR, with reset value 0x0000-0000. A read of this register returns the address of the ISR and sets the current interrupt as being serviced. A read must only be performed while there is an active interrupt. A write of any value to this register clears the current interrupt. A write must only be performed at the end of an interrupt service routine.	0x00000000

3.23.17 VIC Control Register

Address: 0x144

Table 378. VIC Control Register

Bits	Type	Name	Description	Default
0	RW	VIC_EN	Vector Interrupt Controller Enable 0: Disable VIC, IRQ_n is generated directly from IRQStatus[31:0] 1: Enable VIC, IRQ_n is generated from priority logic block.	0x00000000

3.24 Embedded FE PHY Management Registers

3.24.1 MII Control Register

Address: 0x00

Table 379. MII Control Register

Bits	Type	Name	Description	Default
15	RW	Reset	This bit is self clear 0 = normal operation 1 = PHY reset	0x0
14	RW	Loopback	0 = disable loopback mode 1 = enable loopback mode	0x0
13	RW	Speed Selection	0 = 10 Mb/s 1 = 100 Mb/s	0x1
12	RW	AN Enable	0 = disable Auto-Negotiation process 1 = enable Auto-Negotiation process	0x1
11	RW	Power Down	0 = normal operation 1 = power down	0x0
10	RW	Isolate	0 = normal operation 1 = electrically isolate PHY from MII	0x0

9	RW	Restart AN	This bit is self clear 0 = normal operation 1 = restart Auto-Negotiation process	0x0
8	RW	Duplex Mode	0 = half duplex 1 = full duplex	0x1
7:0	RO		Reserved	0x0

3.24.2 MII Status Register

Address: 0x01

Table 380. MII Status Register

Bits	Type	Name	Description	Default
15	RO	100BASE-T4	PHY not able to perform 100BASE-T4, always 0.	0x0
14	RO	100BASE-X FD	0 = PHY is not able to perform full duplex 100BASE-X 1 = PHY is able to perform full duplex 100BASE-X	0x1
13	RO	100BASE-X HD	0 = PHY is not able to perform half duplex 100BASE-X 1 = PHY is able to perform half duplex 100BASE-X	0x1
12	RO	10 Mb/s FD	0 = PHY is not able to operate at 10 Mb/s in full duplex mode 1 = PHY is able to operate at 10 Mb/s in full duplex mode	0x1
11	RO	10 Mb/s HD	0 = PHY is not able to operate at 10 Mb/s in half duplex mode 1 = PHY is able to operate at 10 Mb/s in half duplex mode	0x1
10	RO	100BASE-T2 FD	PHY not able to perform 100BASE-T2, always 0.	0x0
9	RO	100BASE-T2 HD	PHY not able to perform 100BASE-T2, always 0.	0x0
8:7	RO		Reserved	
6	RO	MF Preamble Suppression	0 = PHY will not accept management frames with preamble suppression 1 = PHY will accept management frames with preamble suppression	0x1
5	RO	AN Complete	0 = Auto-Negotiation process not completed 1 = Auto-Negotiation process completed	0x0
4	RO	Remote Fault	0 = no remote fault condition detected 1 = remote fault condition detected	0x0
3	RO	AN Ability	0 = PHY is not able to perform Auto-Negotiation 1 = PHY is able to perform Auto-Negotiation	0x1
2	RO	Link Status	0 = link is down 1 = link is up	0x0
1	RO	Jabber Detect	0 = no jabber condition detected 1 = jabber condition detected	0x0
0	RO	Extended Capability	0 = basic register set capabilities only 1 = extended register capabilities	0x1

3.24.3 PHY Identifier Register-High

Address: 0x02

Table 381. PHY Identifier Register-High

Bits	Type	Name	Description	Default
15:0	RO	PHY_ID[31:16]	This field contains Organizationally Unique Identifier (OUI) bit 3 to 18.	0x1D

3.24.4 PHY Identifier Register-Low

Address: 0x03

Table 382. PHY Identifier Register-Low

Bits	Type	Name	Description	Default
15:10	RO	PHY_ID[15:10]	This field contains Organizationally Unique Identifier (OUI) bit 19 to 24	0x9
9:4	RO	PHY_ID[9:4]	This field contains Manufacturer's Model Number	0x1
3:0	RO	PHY_ID[3:0]	This field contains Revision Number	0x1

3.24.5 Auto-Negotiation Advertisement Register

Address: 0x04

Table 383. Auto-Negotiation Advertisement Register

Bits	Type	Name	Description	Default
15	RW	Next Page	1 = engage to exchange next page 0 = not to exchange next page	0x0
14	RO		Reserved	0x0
13	RW	Remote Fault	1 = Auto-Negotiation remote fault detected 0 = No remote fault	0x0
12:5	RW	Technology Ability	Technology Ability A[7:0], one in a bit means the PHY able to perform this function. A[7]: reserved A[6]: not implemented A[5]: PAUSE operation for full duplex links A[4]: not implemented A[3]: 100BASE-TX full duplex A[2]: 100BASE-TX A[1]: 10BASE-T full duplex A[0]: 10BASE-T	0x2F
4:0	RW	Selector Field	The type of message sent by Auto-Negotiation is IEEE Std 802.3	0x01

3.24.6 Auto-Negotiation Link Parter Base Page Ability Register

Address: 0x05

Table 384. Auto-Negotiation Link Parter Base Page Ability Register

Bits	Type	Name	Description	Default
15	RO	Next Page	1 = Link Parter has Next Page to exchange 0 = No Next Page	0x0
14	RO	Acknowledge	1 = Link Parter has successfully received a Link Code Word. 0 = Link Parter has not successfully received a Link Code Word yet	0x0
13	RO	Remote Fault	1 = Auto-Negotiation fault detected 0 = No remote fault	0x0
12:5	RO	Technology Ability	Technology Ability A[7:0], one in a bit means the bit function capable. A[7]: reserved A[6]: don't care A[5]: PAUSE operation for full duplex links A[4]: don't care A[3]: 100BASE-TX full duplex A[2]: 100BASE-TX A[1]: 10BASE-T full duplex A[0]: 10BASE-T	0x0
4:0	RO	Selector Field	The type of message sent by Link Parter	0x0

3.24.7 Auto-Negotiation Expansion Register

Address: 0x06

Table 385. Auto-Negotiation Expansion Register

Bits	Type	Name	Description	Default
15:5	RO		Reserved	0x0
4	RO	Parallel Detection Fault	1 = A fault has been detected via the Parallel Dectection function 0 = A fault has not been detected via the Parallel Dectection function	0x0
3	RO	Link Parter Next Page Able	1 = Link Parter is Next Page able 0 = Link Parter is not Next Page able	0x0
2	RO	Next Page Able	1 = Local Device is Next Page able 0 = Local Device is not Next Page able	0x1
1	RO	Page Received	1 = A New Page has been received 0 = A New Page has not been received	0x0
0	RO	Link Parter Auto-Negotiation Able	1 = Link Parter is Auot-Negotiation able 0 = Link Parter is not Auot-Negotiation able	0x0

3.24.8 Page Selection Register

Address: 0x1F

Table 386. Page Selection Register

Bits	Type	Name	Description	Default
15	RW	PageSel	Must set to one to enable above basic MII register access.	0x0

14	RW		Reserved	0x0
13:12	RW	LedMode	**LED Mode Selection** 0x0: LED0: Link/Activity LED on: link up off: link down flash: activity LED1: Speed LED on: 100M off: l0M LED2: Duplex/Collision LED on: full off: half flash: collision 0x1: LED0: Link/Activity LED on: link up off: link down flash: activity LED1: Speed LED on: 100M off: l0M LED2: Duplex LED on: full off: half 0x2: LED0: Link/Activity LED if in 100M mode on: link up off: link down flash: activity LED1: Link/Activity LED if in 10M mode on: link up off: link down flash: activity LED2: Duplex/Collision LED on: full off: half flash: collision 0x3: LED0: Link/Activity LED on: link up off: link down flash: activity LED1: Speed LED on: 100M off: l0M LED2: Collision LED flash: collision	0x0
11:0	RW		Reserved	0xF1A

4 Electrical Characteristics

4.1 DC Electrical Characteristics

Table 387. DC Electrical Characteristics

DDR SDRAM Interface (SSTL-2 Class-I) (VDD=2.5V+-10%)

Symbol	Parameter	Condition	Min	Typ.	Max	Unit
V_{ref}	Reference Voltage		1.13	1.25	1.38	V
V_{ih}	Input High Voltage		Vref+0.15		VDD+0.3	V
V_{il}	Input Low Voltage		-0.3		Vref-0.15	V
I_{ih}	Input High Current	Vin=VDD(max)	-10		10	uA
I_{il}	Input Low Current	Vin=GND	-10		10	uA
V_{oh}	Output High Voltage	Ioh=-8.1mA	1.74			V
V_{ol}	Output Low Voltage	Iol=8.1mA			0.76	V
C_{in}	Input Capacitance					pF

SDR SDRAM Interface (VDD=3.3V+-10%)

Symbol	Parameter	Condition	Min	Typ.	Max	Unit
V_{ih}	Input High Voltage		2.0		VDD+0.3	V
V_{il}	Input Low Voltage		-0.3		0.8	V
I_{ih}	Input High Current	Vin=VDD(max)	-10		10	uA
I_{il}	Input Low Current	Vin=GND	-10		10	uA
V_{oh}	Output High Voltage	Ioh=-20mA	2.4			V
V_{ol}	Output Low Voltage	Iol=20mA			0.4	V
C_{in}	Input Capacitance					pF

Flash/SRAM/IDE Interface (VDD=3.3V+-10%)

Symbol	Parameter	Condition	Min	Typ.	Max	Unit
V_{ih}	Input High Voltage		2.0		5.5	V
V_{il}	Input Low Voltage		-0.3		0.8	V
I_{ih}	Input High Current	Vin=VDD(max)	-10		10	uA
I_{il}	Input Low Current	Vin=GND	-10		10	uA
V_{oh}	Output High Voltage	Ioh=-6mA	2.4			V
V_{ol}	Output Low Voltage	Iol=6mA			0.4	V
C_{in}	Input Capacitance					pF

PCI/Cardbus Interface (VCC=3.3V+-10%)

Symbol	Parameter	Condition	Min	Typ.	Max	Unit
V_{ih}	Input High Voltage		0.5VCC		5.5	V
V_{il}	Input Low Voltage		-0.5		0.3VCC	V

I_{ih}	Input High Current	Vin=VCC(max)	-10		10	uA
I_{il}	Input Low Current	Vin=GND	-10		10	uA
V_{oh}	Output High Voltage	Ioh=-500uA	0.9VCC			V
V_{ol}	Output Low Voltage	Iol=1500uA			0.1VCC	V
C_{in}	Input Pin Capacitance			10	16	pF
C_{clk}	Clock Pin Capacitance		5		12	pF

TWI/SPI/PCM/I2S/MISC Interface (VDD=3.3V+-10%)

Symbol	Parameter	Condition	Min	Typ.	Max	Unit
V_{ih}	Input High Voltage		2.0		VDD+0.3	V
V_{il}	Input Low Voltage		-0.3		0.8	V
I_{ih}	Input High Current	Vin=VDD(max)	-10		10	uA
I_{il}	Input Low Current	Vin=GND	-10		10	uA
V_{oh}	Output High Voltage	Ioh=-8mA	2.4			V
V_{ol}	Output Low Voltage	Iol=8mA			0.4	V
C_{in}	Input Capacitance					pF

RGMII Interface (VDD= 2.5V+-10%)

Symbol	Parameter	Condition	Min	Typ.	Max	Unit
V_{ih}	Input High Voltage		1.7		VDD+0.3	V
V_{il}	Input Low Voltage		-0.3		0.7	V
I_{ih}	Input High Current	Vin=VDD(max)	-10		10	uA
I_{il}	Input Low Current	Vin=GND	-10		10	uA
V_{oh}	Output High Voltage	Ioh=-7mA	2.0			V
V_{ol}	Output Low Voltage	Iol=7mA			0.4	V
C_{in}	Input Capacitance		5		12	pF

MII Interface (VDD= 3.3V+-10%)

Symbol	Parameter	Condition	Min	Typ.	Max	Unit
V_{ih}	Input High Voltage		2.0		VDD+0.3	V
V_{il}	Input Low Voltage		-0.3		0.8	V
I_{ih}	Input High Current	Vin=VDD(max)	-10		10	uA
I_{il}	Input Low Current	Vin=GND	-10		10	uA
V_{oh}	Output High Voltage	Ioh=-4mA	2.4			V
V_{ol}	Output Low Voltage	Iol=4mA			0.4	V
C_{in}	Input Capacitance		5		12	pF

USB1.1/2.0 Interface (D+/D-)

Symbol	Parameter	Condition	Min	Typ.	Max	Unit
V_{DIHS}	HS Differential Input Sensitivity	|(D+)-(D-)|	150			mV
V_{CMHS}	HS Common Mode Range		-50		500	mV
V_{HSOH}	HS Output Voltage High		360		440	mV

V_{HSOL}	HS Output Voltage Low		-10		10	mV
V_{DIFS}	FS Differential Input Sensitivity	|(D+)-(D-)|	200			mV
V_{CMFS}	FS Common Mode Range		0.8		2.5	V
V_{FSOH}	FS Output Voltage High		2.8		3.6	V
V_{FSOL}	FS Output Voltage Low				0.3	V
Z_{DRV}	Drive Output Impedance for HS & FS		40.5	45	49.5	Ω
I_{LO}	Hi-Z State Data Line Leakage		-10		10	uA

REF_32768 (VDD= 1.8V+-10%)

Symbol	Parameter	Condition	Min	Typ.	Max	Unit
V_{ih}	Input High Voltage		1.5		2.0	V
V_{il}	Input Low Voltage		-0.2		0.4	V
I_{ih}	Input High Current	Vin=VDD(max)	-10		10	uA
I_{il}	Input Low Current	Vin=GND	-10		10	uA

4.2 Absolute Maximum Ratings

Table 388. Absolute Maximum Ratings

Parameter	Symbol	Min	Max	Unit
I/O Supply Voltage		-0.5	4.6	V
Core Supply Voltage		-0.5	2.5	V
Input Voltage		-0.5	6.0	V
Output Voltage		-0.5	6.0	V
Storage Temperature		-65	150	℃
ESD Protection(Human Body Mode)	*HBM*	2000		V
ESD Protection(Machine Mode)	*MM*	200		V

4.3 Recommended Operation Conditions

Table 389. Recommended Operation Conditions

Parameter	Symbol	Min.	Max.	Unit
I/O Supply Voltage		3.0	3.6	V
I/O Supply Voltage (for MII/Reverse MII)		3.0	3.6	V
I/O Supply Voltage (for RGMII)		2.25	2.75	V
I/O Supply Voltage (for DDR)		2.25	2.75	V
Core Supply Voltage		1.62	1.98	V
Operation Temperature		0	70	℃
Junction Temperature		0	125	℃

Low-level Input Voltage		-0.3	0.8	V
High-level Input Voltage		2.0	5.5	V
PQFP-128 package (with heat spreader) thermal resistance from junction to ambient – no air flow @1.1W	$\theta_{JA(0m/sec)}$		35	℃/W
PQFP-128 package (with heat spreader) thermal resistance from junction to ambient – air flow 2m/sec @1.1W	$\theta_{JA(2m/sec)}$		29	℃/W
LFBGA-269 package thermal resistance from junction to ambient – no air flow @1.2W	$\theta_{JA(0m/sec)}$		29.2	℃/W
LFBGA-269 package thermal resistance from junction to ambient – air flow 2m/sec @1.2W	$\theta_{JA(2m/sec)}$		24.6	℃/W

4.4 Power Consumption

4.4.1 Maximum Current Consumption

The following tables show the absolute maximum operating current for STR81XX/CNS21XX device. Please note, not all power domains consume maximum power at the same time. The relevant test conditions are also shown in the table below.

Table 390. STR81XX/CNS21XX Maximum Current Consumption

Power Supply	Power Domains	Condition	Maximum	Unit
AVDD_E (3.6V)	Fast Ethernet PHY power supply	Max power supply voltage At 0℃ PHY link down	115	mA
VCCHRST (3.6V) VCCA_U20 (3.6V)	USB Device PHY power supply USB Device PHY power supply	Max power supply voltage At 70℃	33	mA
PVDD (3.6V) AVDD_U33 (3.6V) AVDD_R33 (3.6V)	IO power supply USB Host PHY power supply Regulator power supply	Max power supply voltage At 0℃	45	mA
PVDD_D (2.75V)	DDR IO power supply	Max power supply voltage At 0℃	64	mA
PVDD_E (2.75V)	RGMII IO power supply	Max power supply voltage At 0℃	20	mA
CVDD (1.98V) AVDD_U (1.98V) AVDD_UP (1.98V) AVDD_SP (1.98V)	Core power supply USB Host PHY power supply USB Host PHY power supply System PLL power supply	Max power supply voltage At 70℃	429	mA

4.4.2 Typical and Sleep Current Consumption

The following tables show typical values of current consumption for STR81XX /CNS21XX in various situations. The values are measured under typical conditions (process,

temperature, and supply voltage), and using DDR SDRAM. The ”Sleep” column below means chip in sleep state. The ”Typ-200” column below means chip in full speed operation with CPU operating at 200MHz, AHB clock at 100MHz, and APB clock at 50MHz. The ”Typ-250” column below means chip in full speed operation with CPU operating at 250MHz, AHB clock at 125MHz, and APB clock at 62.5MHz.

Table 391. STR81XX/CNS21XX Typical and Sleep Current Consumption

Power Supply	Power Domains	Sleep	Typ-200	Typ-250	Unit
AVDD_E (3.3V)	Fast Ethernet PHY power supply	5	76	76	mA
VCCHRST (3.3V) VCCA_U20 (3.3V)	USB Device PHY power supply USB Device PHY power supply	5	29	30	mA
PVDD (3.3V) AVDD_U33 (3.3V) AVDD_R33 (3.3V)	IO power supply USB Host PHY power supply Regulator power supply	17	36	37	mA
PVDD_D (2.5V)	DDR IO power supply	3	55	56	mA
PVDD_E (2.5V)	RGMII IO power supply	0	15	15	mA
CVDD (1.8V) AVDD_U (1.8V) AVDD_UP (1.8V) AVDD_SP (1.8V)	Core power supply USB Host PHY power supply USB Host PHY power supply System PLL power supply	23	301	347	mA

4.4.3 Power Consumption

The following table shows the sleep, typical, and maximum power consumption based on the current and supply voltages listed in above two current consumption tables. Please note that STR813X/CNS213X has no RGMII interface, PVDD_E power consumption item is not included in STR813X /CNS213X’s power consumption. And STR818X/CNS218X has no FE-PHY, so AVDD_E power consumption item is not included in STR818X/CNS218X’s power consumption.

Table 392. STR813X/CNS213X, STR818X/CNS218X Power Consumption

Part Number	Sleep	Typ-200	Typ-250	Max.	Unit
STR813X/ CNS213X	138	1145	1237	1721	mW
STR818X/ CNS218X	122	932	1024	1362	mW

4.5 Voltage Regulator and PLL Supply Voltage Specification

The following tables show the voltage regulator of internal CPU core power supply and PLL supply voltage specification

Table 393. STR81XX/CNS21XX Voltage Regulator Specification

Part Number	Range		Unit
STR81XX/ CNS21XX	Low Bound	Upper Bound	
	1.66	1.98	Volt

Table 394. STR81XX/CNS21XX PLL Supply Voltage Specification

Part Number	Range		Unit
STR81XX/ CNS21XX	Low Bound	Upper Bound	
	1.65	1.98	Volt

4.6 AC Timing Specifications

4.6.1 SMC Interface Timing

4.6.1.1 SRAM/Flash

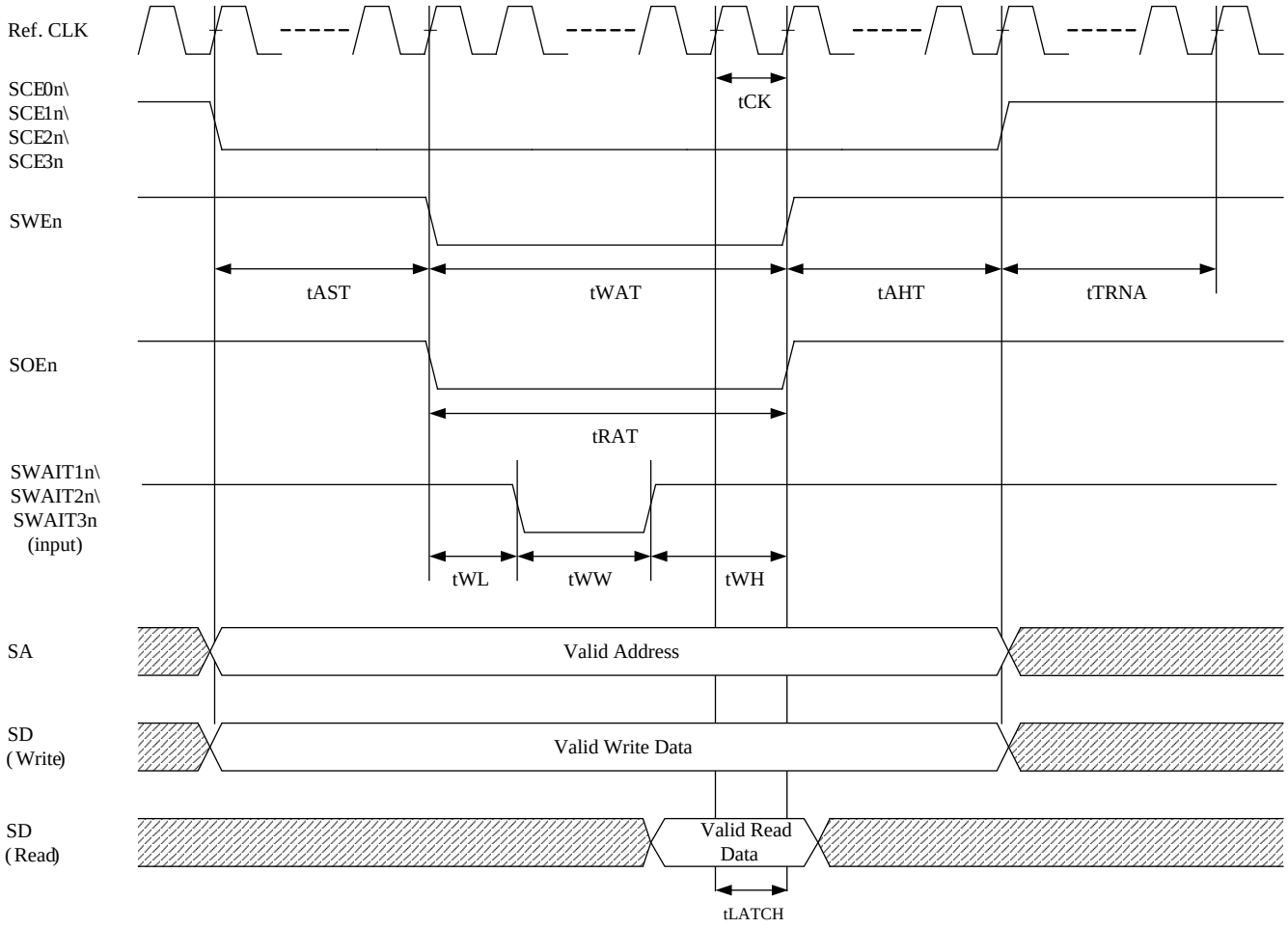


Figure 22. SRAM/Flash Interface Timing Diagram

Table 395. SRAM/Flash Interface Timing

Symbol	Parameter	Min	Typ	Max	Unit	Note
t_{AST}	Address Setup Time		1~16		t_{CK}	1, 2
t_{WAT}	Write Access Time		1~128		t_{CK}	3
t_{RAT}	Read Access Time		1~128		t_{CK}	4
t_{AHT}	Address Hold Time		1~16		t_{CK}	5
t_{TRNA}	Turn-Around Time		1~128		t_{CK}	6
t_{LATCH}	Read Data Latch Time		1		t_{CK}	7

t_{WL}	SWEn low to SWAITn low	0		t_{WAT} - 2	t_{CK}	
t_{WL}	SOEn low to SWAITn low	0		t_{RAT} - 2	t_{CK}	
t_{WW}	SWAITn width	0			t_{CK}	
t_{WH}	SWAITn high to SWEn/SOEn high	1			t_{CK}	

Note:

1. Reference clock (t_{CK}) is an internal system clock running at 125/112.5/100/87.5 MHz. This corresponds to a period of t_{CK} is 8/8.89/10/11.4 ns. The system clock frequency is programmable.
2. t_{AST} is programmable from 1 to 16 system clock cycles. Please refer register description of SMC.
3. t_{WAT} is programmable from 1 to 128 system clock cycles. When SWAITn asserted, the t_{WAT} will be the programmed cycle + SWAITn width (t_{WW}).
4. t_{RAT} is programmable from 1 to 128 system clock cycles. When SWAITn asserted, the t_{RAT} will be the programmed cycle + SWAITn width (t_{WW}).
5. t_{AHT} is programmable from 1 to 16 system clock cycles. Please refer register description of SMC.
6. t_{TRNA} is programmable from 1 to 128 system clock cycles. Please refer register description of SMC.
7. Read Data is latched at one cycle before SOEn de-asserted.

4.6.2 SDMC Interface Timing

4.6.2.1 DDR SDRAM

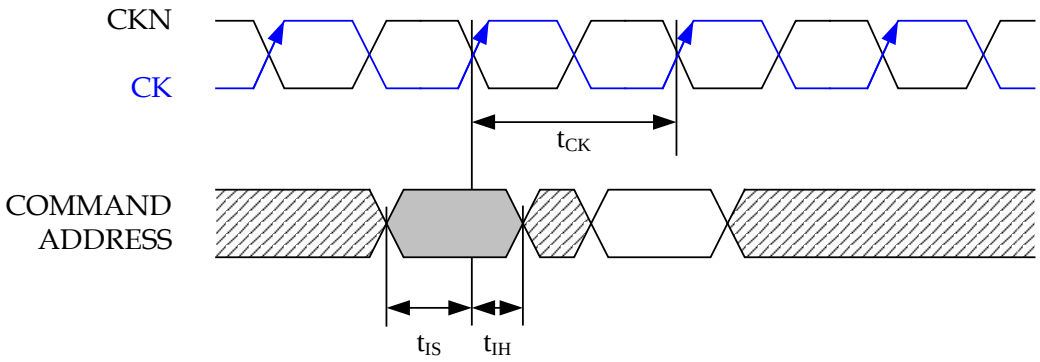


Figure 23. DDR Command Interface Timing Diagram

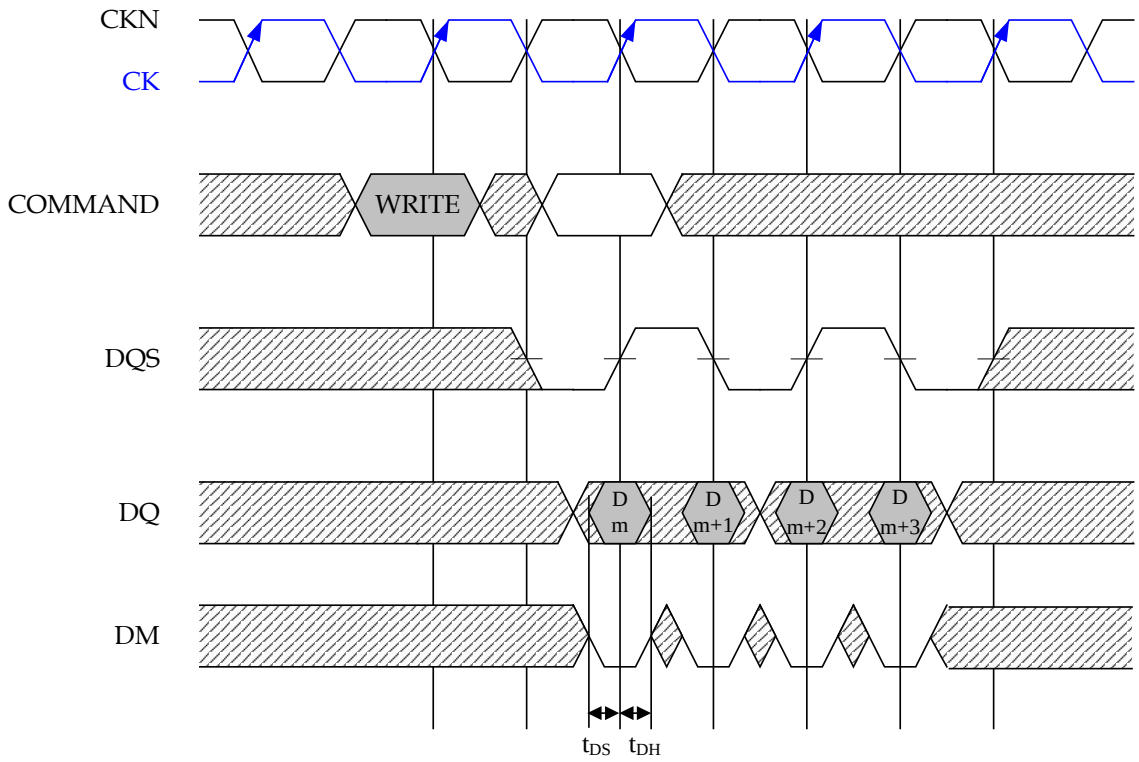


Figure 24. DDR Interface Write Timing Diagram

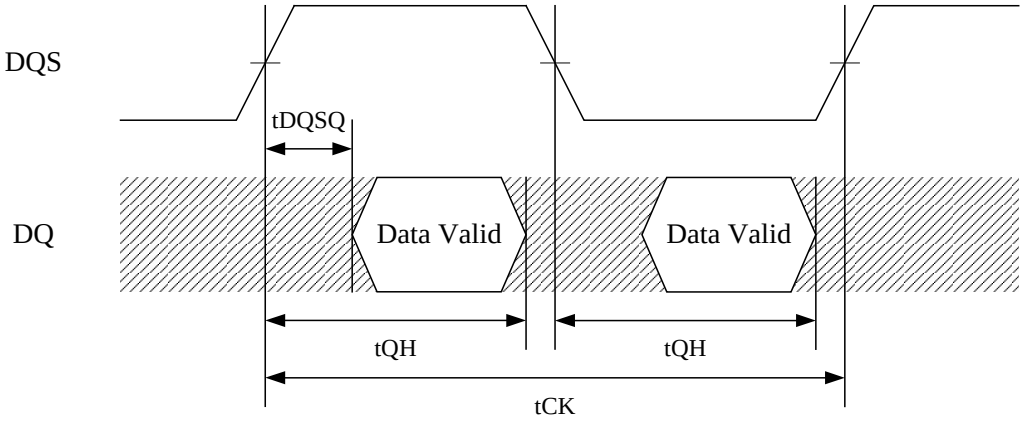


Figure 25. DDR Interface Read Timing Diagram

Table 396. DDR Interface Timing

Symbol	Parameter	Min	Typ	Max	Unit
t_{IS}	Address and control setup time	3.4			ns
t_{IH}	Address and control hold time	1.5			ns
t_{DS}	DQ and DM setup time relative to DQS	1.0			ns
t_{DH}	DQ and DM hold time relative to DQS	1.4			ns
t_{DOSO}	DQS to last DQ valid			0.8	ns

t_{QH}	DQS to first DQ to go non-valid	3.2			ns

Note:

1. Reference clock (t_{CK}) is an internal system clock running at 125 MHz. This corresponds to a period of t_{CK} is 8 ns.
2. Values of t_{DQSQ} and t_{QH} are measured based on SDMC.DQS_IN_DLY_CTRL = 0x00330033.

4.6.2.2 SDR SDRAM

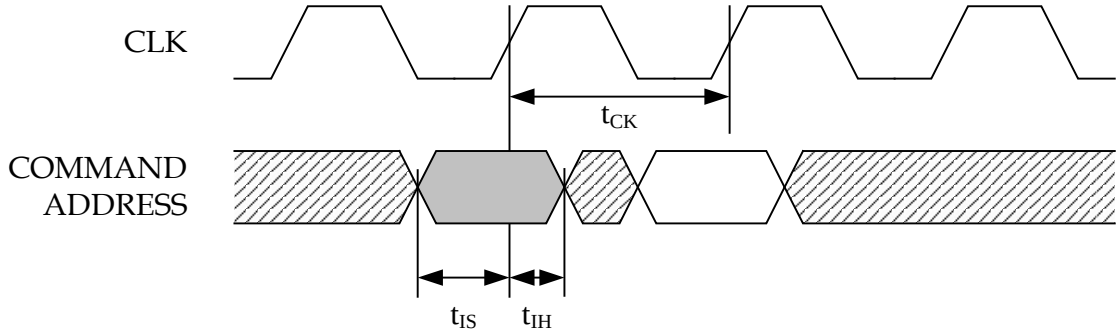


Figure 26. SDR Command Interface Timing Diagram

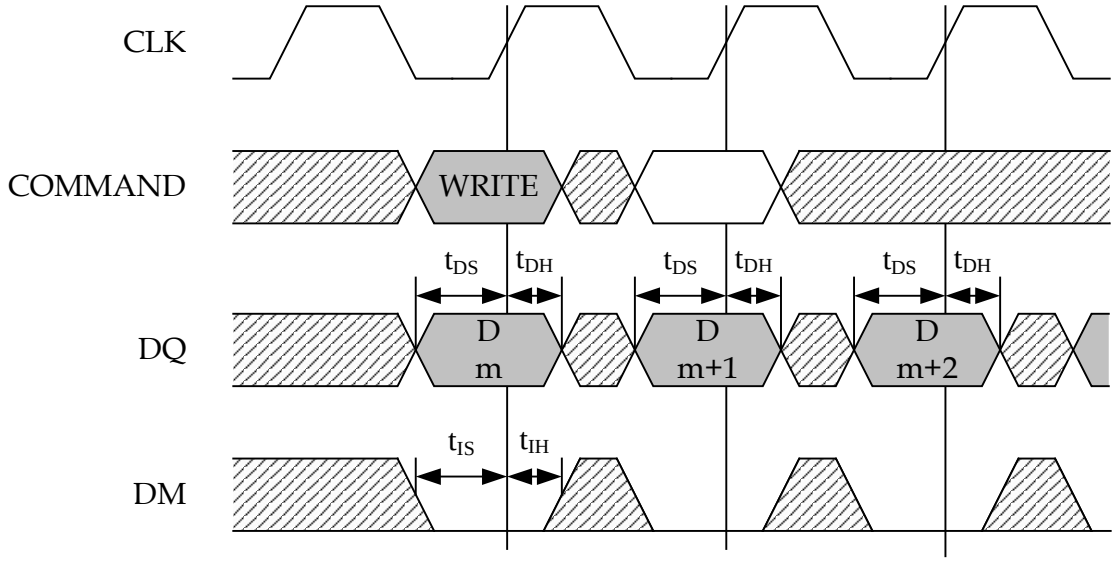


Figure 27. SDR Interface Write Timing Diagram

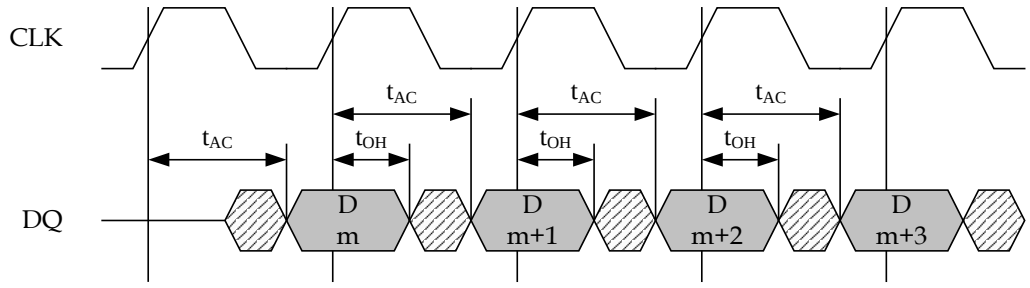


Figure 28. SDR Interface Read Timing Diagram

Table 397. SDR Interface Timing

Symbol	Parameter	Min	Typ	Max	Unit	Note
t_{IS}	Address and control setup time	1.5			ns	1
t_{IH}	Address and control hold time	1.0			ns	
t_{DS}	Data setup time	1.7			ns	
t_{DH}	Data hold time	1.0			ns	
t_{AC}	Access time from CLK			7.2	ns	
t_{OH}	Data hold time from CLK	0.2			ns	

Note:
1. Reference clock (t_{CK}) is an internal system clock running at 125MHz. This corresponds to a period of t_{CK} is 8ns.

4.6.3 IDE Interface Timing

4.6.3.1 PIO Mode – Register transfer to/from device

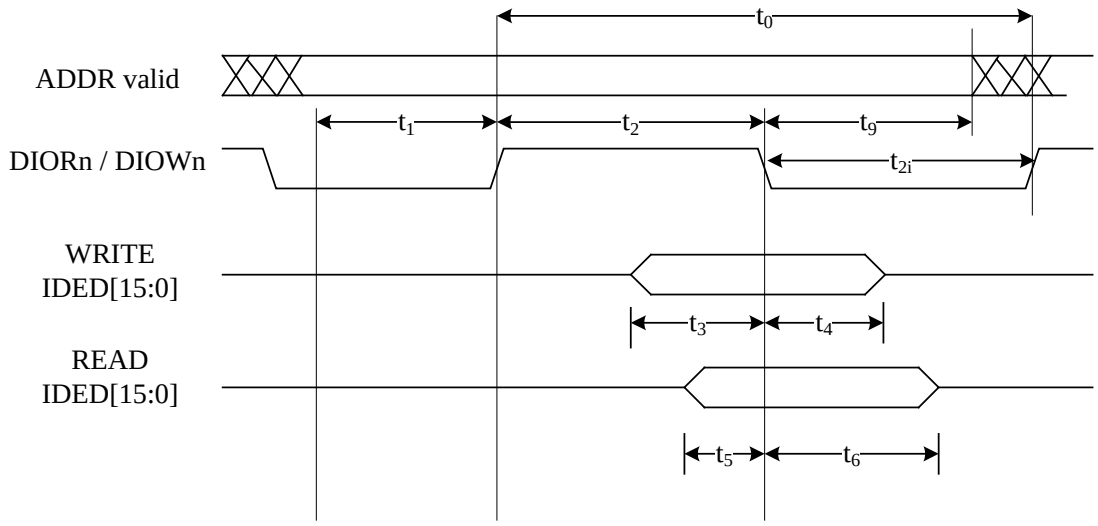


Figure 29. PIO Mode – Register transfer to/from device timing Diagram

Table 398. PIO Mode Timing

Symbol	Parameter	Mode0	Mode1	Mode2	Mode3	Mode4	Unit	Note
		Min	Min	Min	Min	Min		
t_0	Cycle time	600	383	330	180	120	ns	
t_1	Address valid to DIORn/DIOWn setup	70	50	30	30	25	ns	2
t_2	DIORn/DIOWn asserted pulse width	290	290	290	80	70	ns	2
t_{2i}	DIORn/DIOWn recovery time	-	-	-	70	25	ns	2

t_3	DIOWn data setup	60	45	30	30	20	ns	
t_4	DIOWn data hold	30	20	15	10	10	ns	
t_5	DIORn data setup	50	35	20	20	20	ns	
t_6	DIORn data hold	5	5	5	5	5	ns	
t_9	DIORn/DIOWn to address valid hold	20	15	10	10	10	ns	

Note:

1. In timing diagrams, the lower line indicates negated, and the upper line indicates asserted.
2. The timing can be configured with IDE Register -- "*IDE Drive0/1 PIO Timing Configuration Register*"

4.6.3.2 Multiword DMA mode

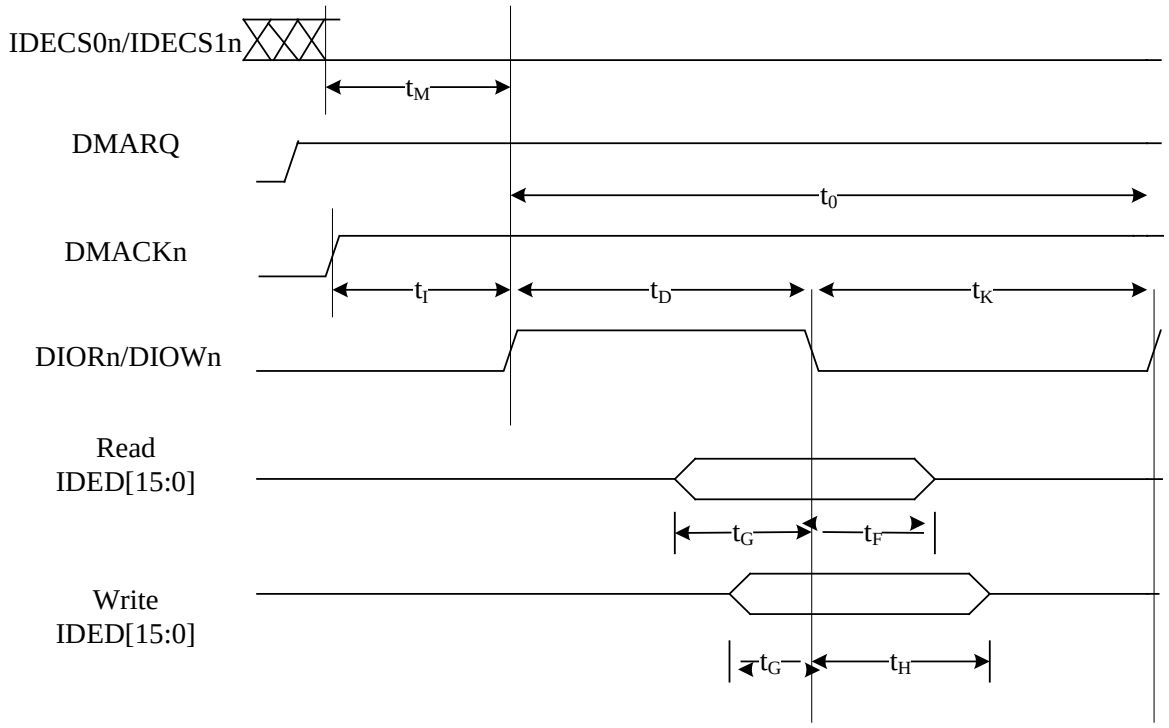


Figure 30. Multiword DMA mode timing diagram

Table 399. Multiword DMA mode timing

Symbol	Parameter	Mode0	Mode1	Mode2	Unit	Note
		Min	Min	Min		
t_0	Cycle time	480	150	120	ns	
t_D	DIORn/DIOWn asserted pulse width	215	80	70	ns	2
t_{KR}	DIORn negated pulse width	50	50	25	ns	2
t_{KW}	DIOWn negated pulse width	215	50	25	ns	2
t_F	DIORn data hold	10	10	10	ns	
t_G	DIORn/DIOWn data setup	100	30	20	ns	
t_H	DIOWn data hold	20	15	10	ns	
t_M	IDECSn(1:0) valid to DIORn/DIOWn	50	30	25	ns	2

t_I	DMACK to DIORn/DIOWn setup	40	40	40	ns	

Note:

1. In timing diagrams, signals, the lower line indicates negated, and the upper line indicates asserted.
2. The timing can be configured with IDE Register -- "*IDE Drive0/1 DMA Timing Configuration Register*"

4.6.3.3 UltraDMA Mode

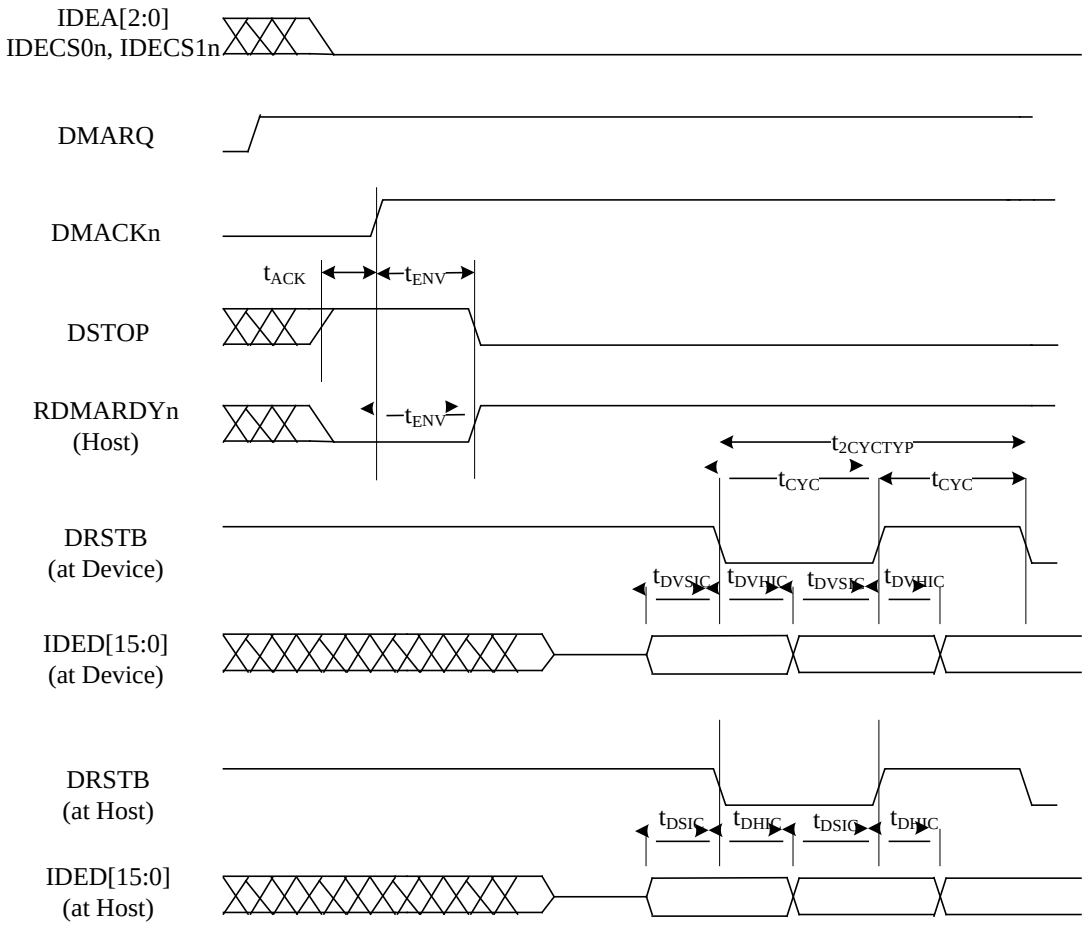


Figure 31. UltraDMA data-in burst

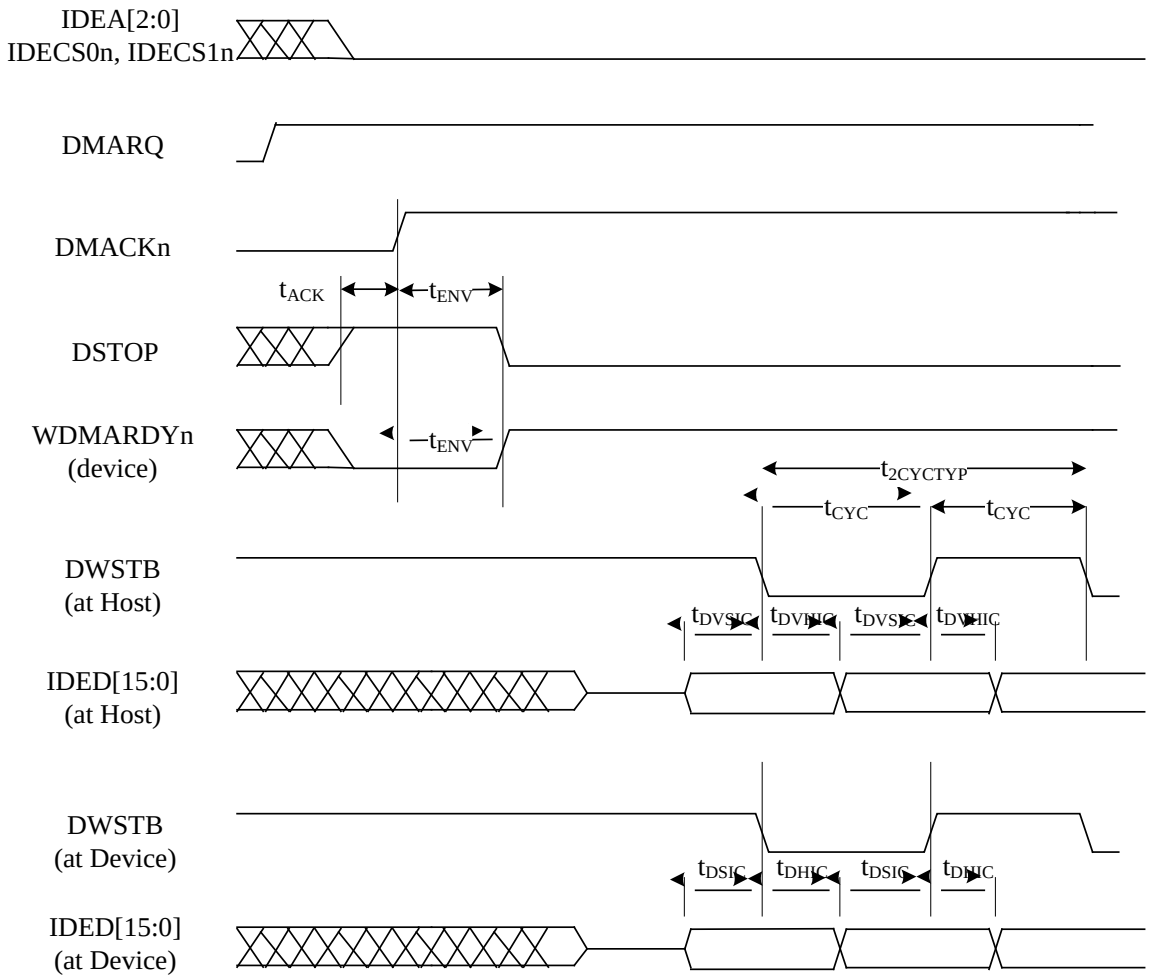


Figure 32. UltraDMA data-out burst

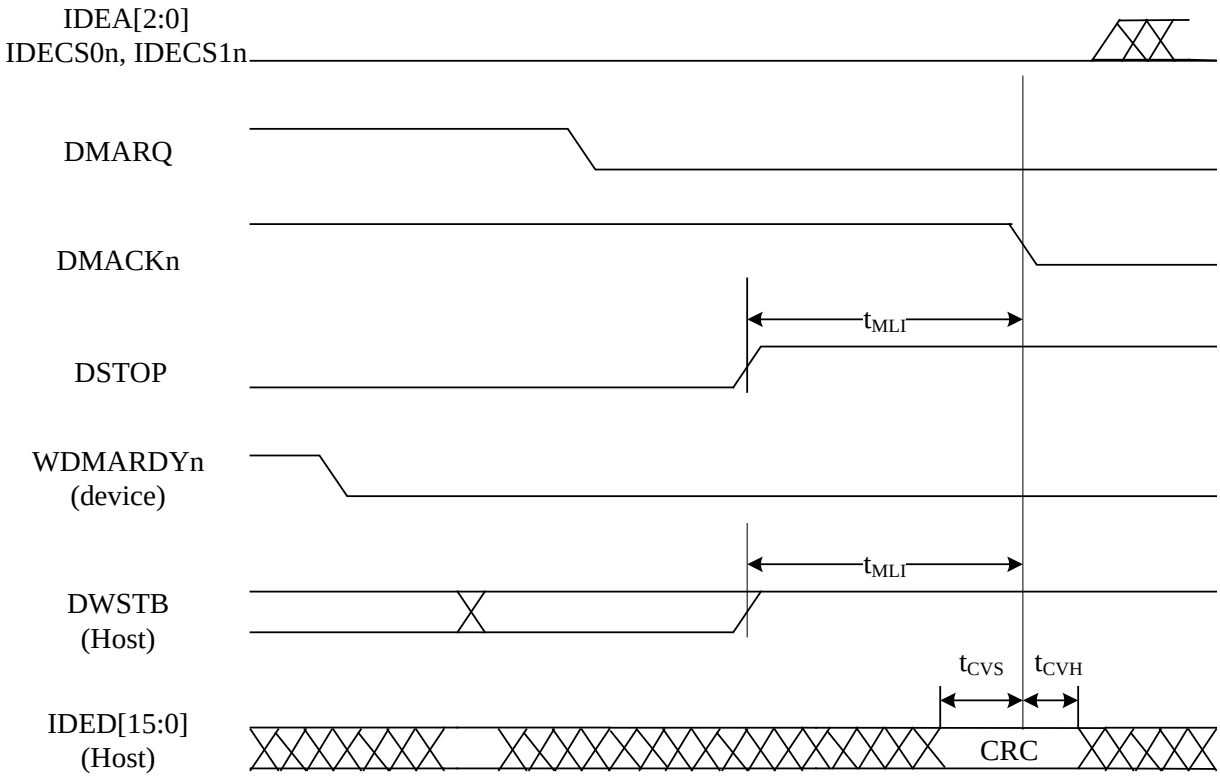


Figure 33. Device terminating an UltraDMA data-out burst

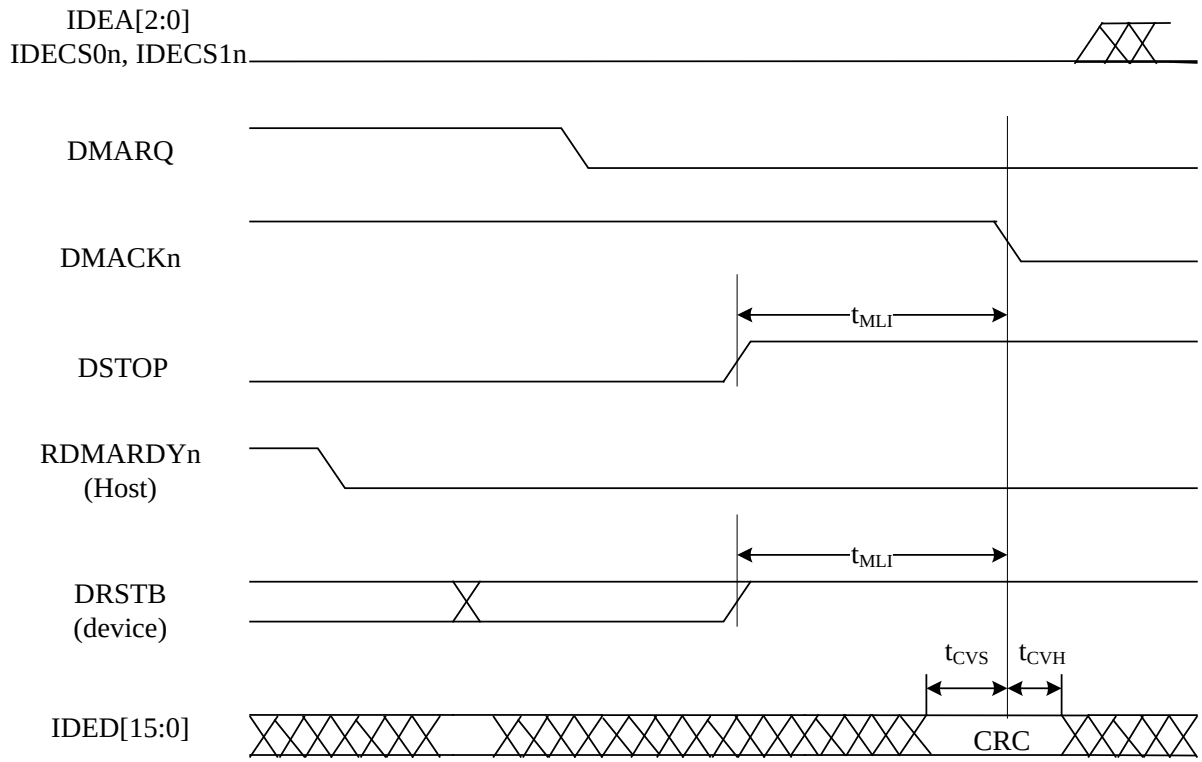


Figure 34. Device terminating an UltraDMA data-in burst

Table 400. UltraDMA Mode Timing

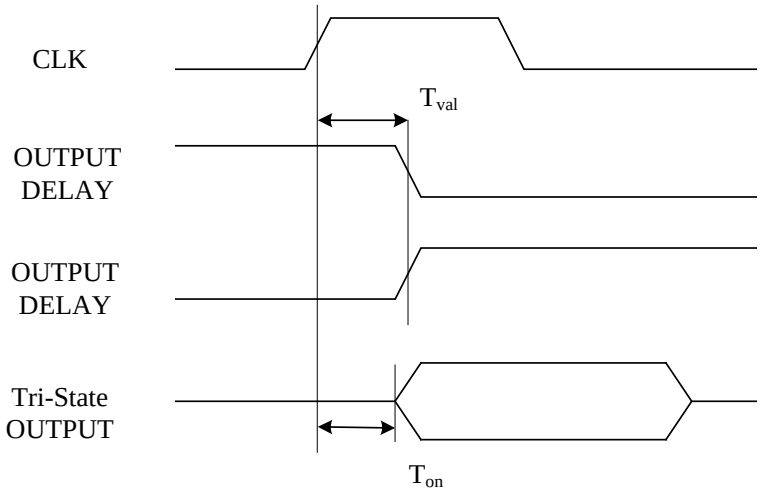
Symbol	Parameter	(Mode0)		(Mode5)		Unit	Note
		Min	Max	Min	Max		
$t_{2CYCTYP}$	Typical sustained average two cycle time	240		40		ns	
t_{CYC}	Cycle time allowing for asymmetry and clock variations (from STROBE edge to STROBE edge)	112		16.8		ns	2
t_{ACK}	Setup and hold times for DMACKn	20		20		ns	
t_{ENV}	Envelope time (from DMACKn to DSTOP and RDMARDYn during data in burst initiation and from DMACKn to DSTOP during data out burst initiation)	20	70	20	50	ns	2
t_{DVSIC}	Sender IC data valid setup time	72.9		10.7		ns	3
t_{DVHIC}	Sender IC data valid hold time	8.3		8.3		ns	3
t_{DSIC}	Recipient IC data setup time	14.7		2.3		ns	
t_{DHIC}	Recipient IC data hold time	4.8		2.8		ns	
t_{CVS}	CRC word valid setup time at host (from CRC valid until DMACKn negation)	70		11.6		ns	
t_{CVH}	CRC word valid hold time at sender (from DMACKn negation until CRC may become invalid)	8.4		8.4		ns	

t_{MLI}	Interlock time with minimum	20		20		ns	2

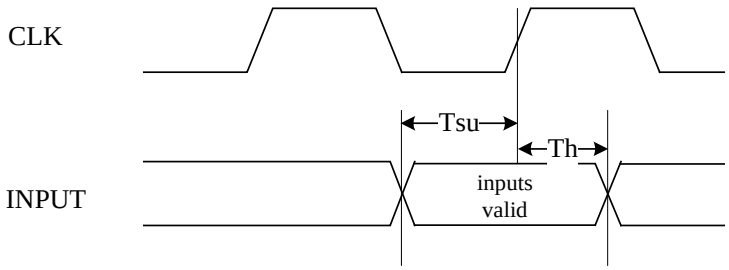
Note:

1. In timing diagrams, the lower line indicates negated, and the upper line indicates asserted.
2. The timing can be configured with IDE Register -- "*IDE Ultra DMA Timing Configuration Register*"
3. The values are for data out burst (Host drived). For data in burst, the value t_{DVSIC} is 6ns for mode5 and 72.9ns for mode0. The value t_{DVHIC} is 6ns for mode5 and 9ns for mode0.
4. At the table, just only list mode 0 and mode5 timing parameter. (Actually chip can support UltraDMA mode0 to UltraDMA mode5)

4.6.4 PCI Interface Timing



Output Timing Measurement Conditions



Input Timing Measurement Conditions

Figure 35. PCI Interface Timing

Table 401. PCI Interface Timing

Symbol	Parameter	33/66MHz		Unit
		Min	Max	
T_{val}	**CLK** to Signal Valid Delay - bused signals	2	8	ns
$T_{val(ptp)}$	**CLK** to Signal Valid Delay - point to point signals	2	6.6	ns

T_{on}	Float to Active Delay	2		ns
T_{su}	Input Setup Time to **CLK** - bused signals	3		ns
$T_{su(ptp)}$	Input Setup Time to **CLK** - point to point signals	4		ns
T_h	Input Hold Time from **CLK**	0		ns

Note:

1. **REQn** and **GNTn** are point-to-point (ptp) signals and have different input setup times than do bused signals.

4.6.5 TWI Interface Timing

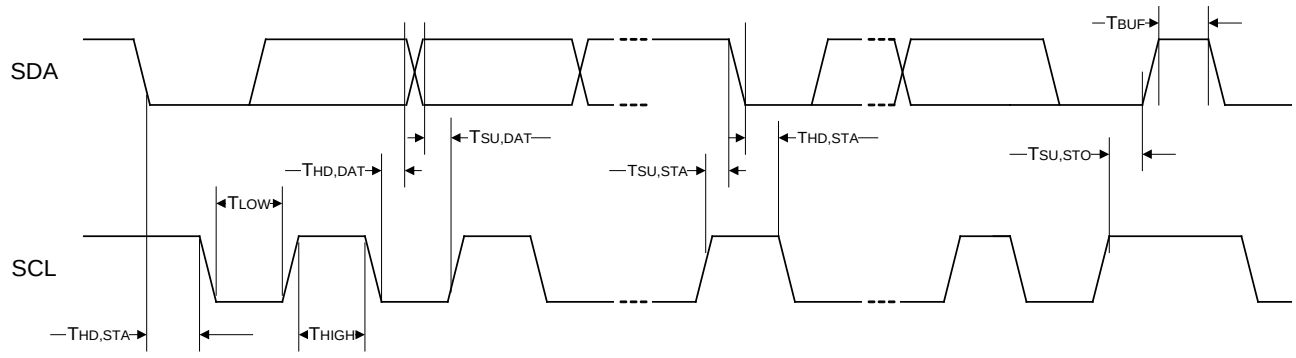


Figure 36. TWI Read

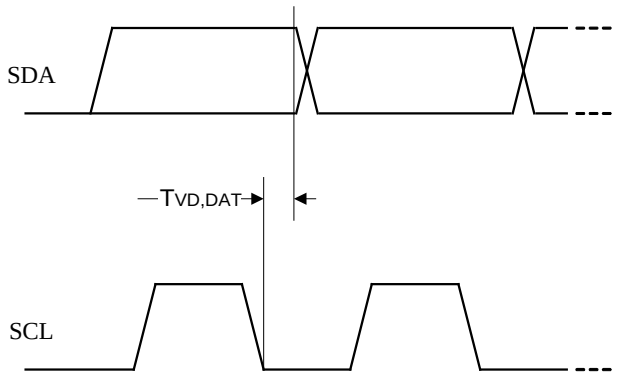


Figure 37. TWI Write

Table 402. TWI Interface Timing

Symbol	Parameter	Min	Typ	Max	Unit	Notes
T_{CYC}	SCL period	80			ns	1
T_{LOW}	Clock low period	0.4	0.5	0.6	T_{CYC}	
T_{HIGH}	Clock high period	0.4	0.5	0.6	T_{CYC}	
T_{BUF}	Bus free time between STOP and START condition	0.5			T_{CYC}	
$T_{HD,STA}$	Hold time after (repeated) START condition	1.0			T_{CYC}	

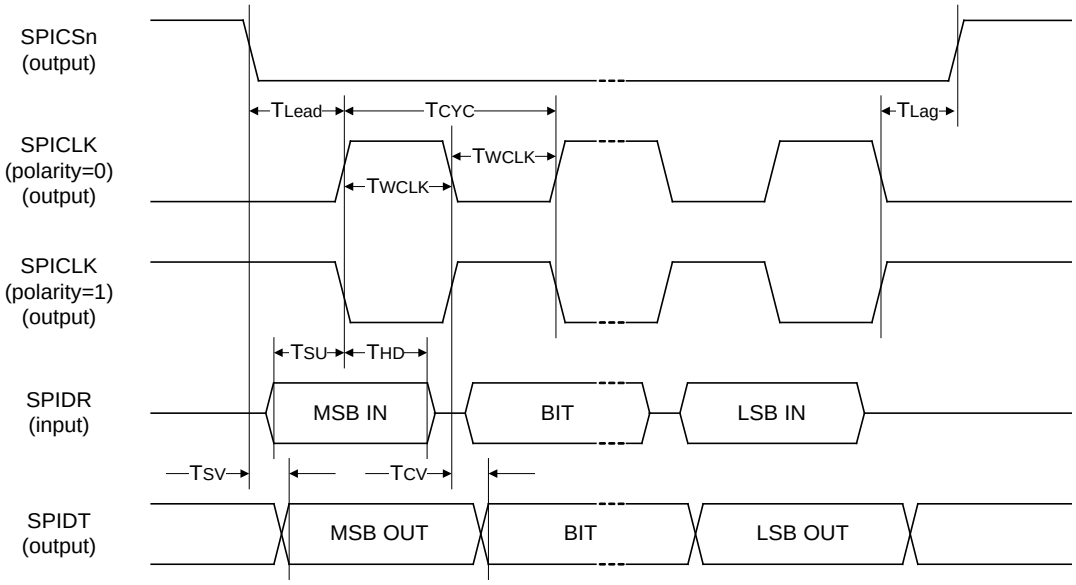
$T_{SU,STA}$	Repeated START condition setup time	0.5			T_{CYC}	
$T_{SU,STO}$	Setup time for STOP condition	0.5			T_{CYC}	
$T_{HD,DAT}$	Data in hold time	0			ns	
$T_{SU,DAT}$	Data in setup time	14			ns	
$T_{VD,DAT}$	Data out valid time	0		8.0	ns	

Note:

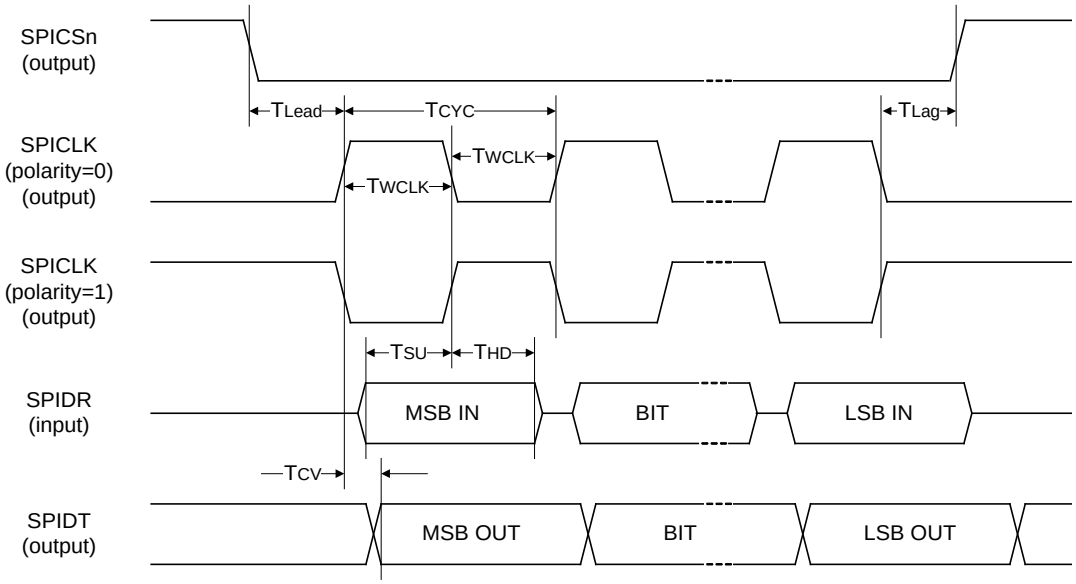
1. The clock rate can be configured by bit [16:8] of “TWI Time-Out Register”.

4.6.6 SPI Interface Timing

4.6.6.1 SPI Master



Phase = 0



Phase = 1

Figure 38. SPI Master Timing Diagram

Table 403. SPI Master Timing

Symbol	Parameter	Min	Typ	Max	Unit	Notes
T_{CYC}	SPICLK period	20			ns	1

T_{WCLK}	SPICLK High or Low time	0.4	0.5	0.6	T_{CYC}	
T_{Lead}	SPICSn Lead time	T_{CYC} – 4.5			ns	
T_{Lag}	SPISCSn Lag time	0.5			T_{CYC}	
T_{SU}	Data input setup time	1.0			ns	
T_{HD}	Data input hold time	2.0			ns	
T_{SV}	Data output valid after SPICSn falling edge	0.0		0.5	ns	2
T_{CV}	Data output valid after SPICLK edge	0.3		4.0	ns	

Note:

1. The minimum period of T_{CYC} is 20ns. The clock rate can be configured by bit [2:0] of “SPI Bit Rate Register”.
2. Only for Phase = 0.

4.6.6.2 SPI Slave

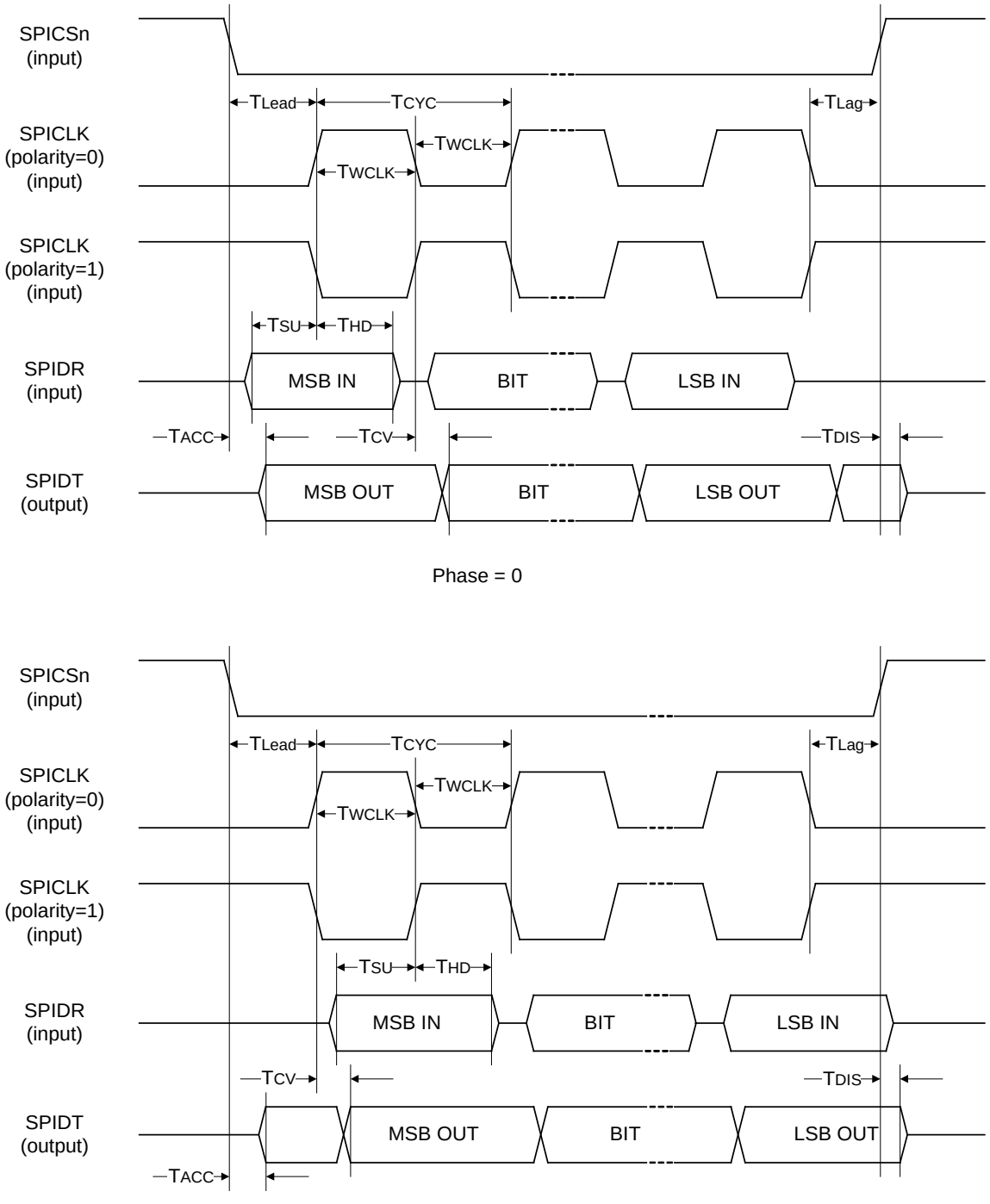


Figure 39. SPI Slave Timing Diagram

Table 404. SPI Slave Timing

Symbol	Parameter	Min	Typ	Max	Unit	Notes
T_{CYC}	SPICLK period	40			ns	1

T_{WCLK}	SPICLK High or Low time	0.4	0.5	0.6	T_{CYC}	
T_{Lead}	SPICSn Lead time	14.9			ns	
T_{Lag}	SPICSn Lag time	0.0			ns	
T_{SU}	Data input setup time	0.0			ns	
T_{HD}	Data input hold time	3.6			ns	
T_{CV}	Data output valid after SPICLK edge	3.6		10.6	ns	
T_{ACC}	Access time	3.4		9.9	ns	
T_{DIS}	SPIDT Disable time	3.4		13.9	ns	

Note:
1. The minimum period of T_{CYC} is 40ns. The clock rate can be configured by bit [2:0] of "SPI Bit Rate Register".

4.6.6.3 MPI Master

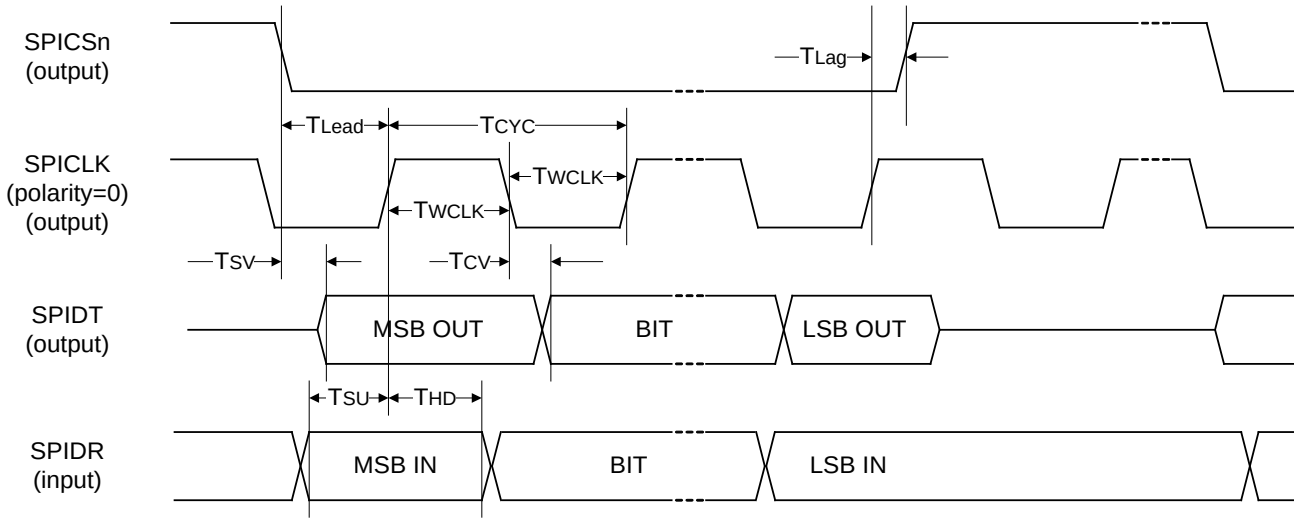


Figure 40. MPI Master Timing Diagram

Table 405. MPI Master Timing

Symbol	Parameter	Min	Typ	Max	Unit	Notes
T_{CYC}	SPICLK period	20			ns	1
T_{WCLK}	SPICLK High or Low time	0.4	0.5	0.6	T_{CYC}	
T_{Lead}	SPISSn Lead time	T_{CYC} – 4.5			ns	
T_{Lag}	SPISSn Lag time	0.6		4.1	ns	
T_{SU}	Data input setup time	1.0			ns	
T_{HD}	Data input hold time	2.0			ns	
T_{SV}	Data output valid after SPISSn falling edge	0.0		0.5	ns	
T_{CV}	Data output valid after SPICLK edge	0.3		4.0	ns	

Note:
1. The minimum period of T_{CYC} is 20ns. The clock rate can be configured by bit [2:0] of "SPI Bit Rate Register".

4.6.7 PCM Interface Timing

4.6.7.1 Long Frame Sync PCM

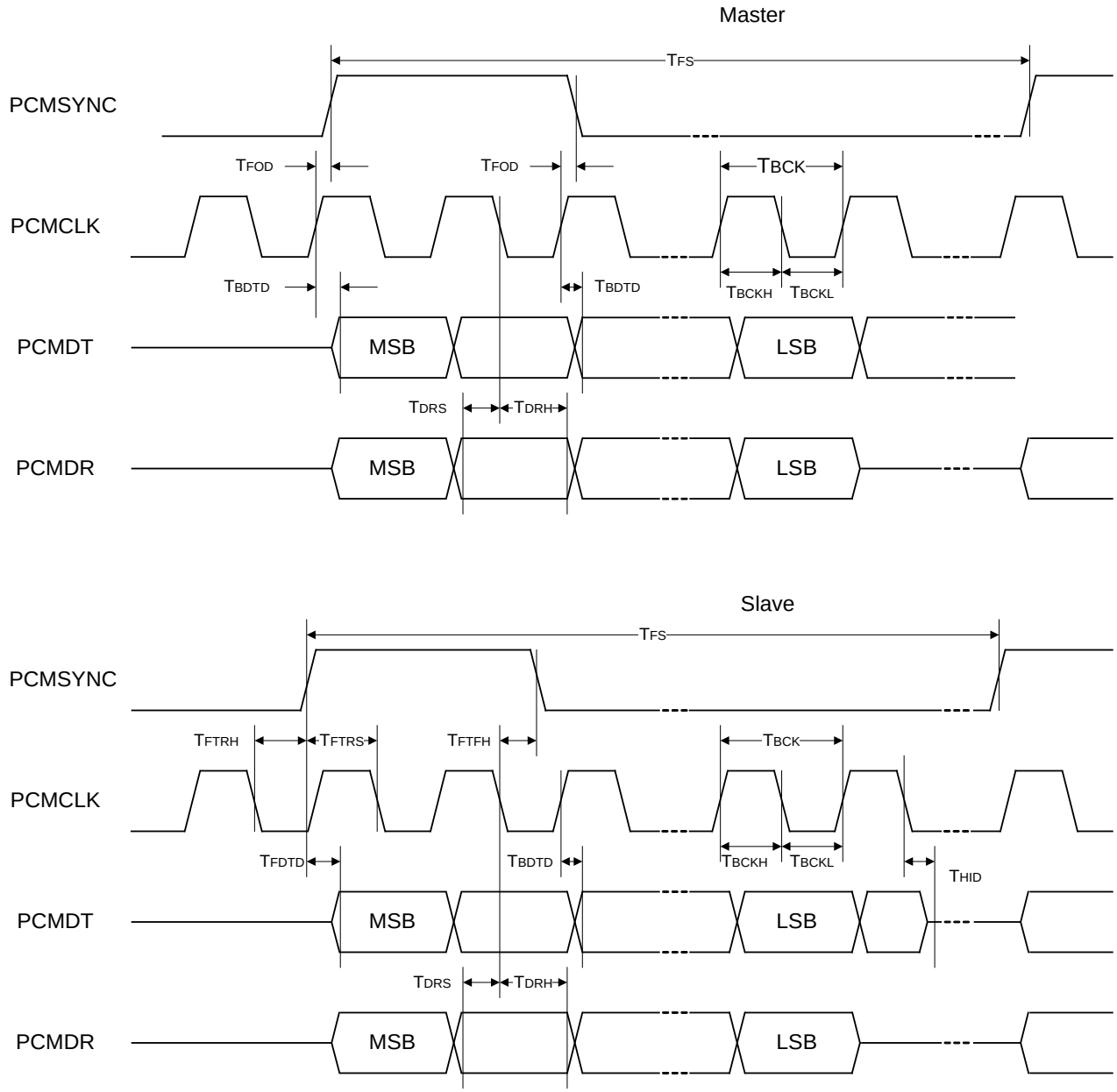


Figure 41. Long Frame Sync PCM

Table 406. Long Frame Sync PCM Timing

Symbol	Parameter	Min	Typ	Max	Unit	Notes
$1 / T_{FS}$	PCMSYNC frequency		8.0		KHz	
$1 / T_{BCK}$	PCMCLK frequency	128		4096	KHz	1, 2
T_{BCKH}	PCMCLK high pulse width	0.45	0.5	0.55	T_{BCK}	

T_{BCKL}	PCMCLK low pulse width	0.45	0.5	0.55	T_{BCK}	
T_{DRS}	PCMDR to PCMCLK fall edge setup time	5.5			ns	2
T_{DRH}	PCMDR to PCMCLK fall edge hold time	0.0			ns	2
T_{FOD}	PCMCLK rise edge to PCMSYNC edge delay time	0.9		5.4	ns	2
T_{BDTD}	PCMCLK rise edge to valid PCMDT delay time	0.4		8.7	ns	2
$1 / T_{BCK}$	PCMCLK frequency	128		2048	KHz	1, 3
T_{DRS}	PCMDR to PCMCLK fall edge setup time	0.0			ns	3
T_{DRH}	PCMDR to PCMCLK fall edge hold time	0.5			ns	3
T_{HID}	PCMCLK rise edge to high impedance delay time	2.9		13.3	ns	3
T_{BDTD}	PCMCLK rise edge to valid PCMDT delay time	2.9		13.3	ns	3
T_{FDTD}	PCMSYNC rise edge to Valid PCMDT delay time	2.7		11	ns	3
T_{FTRH}	PCMCLK fall edge to PCMSYNC rise edge hold time	0.2			ns	3
T_{FTRS}	PCMSYNC rise edge to PCMCLK fall edge setup time	61.0			ns	3
T_{FTFH}	PCMSYNC rise edge to PCMCLK fall edge setup time	0.2			ns	3

Note:

1. The minimum period of T_{CYC} is around 244ns. The clock rate can be configured by bit [2:0] of "PCM Configuration Register".
2. It is for master. In master mode, PCMCLK, PCMSYNC and PCMDT are outputs; and PCMDR is input.
3. It is for slave. In slave mode, PCMDT is output; and PCMCLK, PCMSYNC and PCMDR are inputs.

4.6.7.2 Short Frame Sync PCM

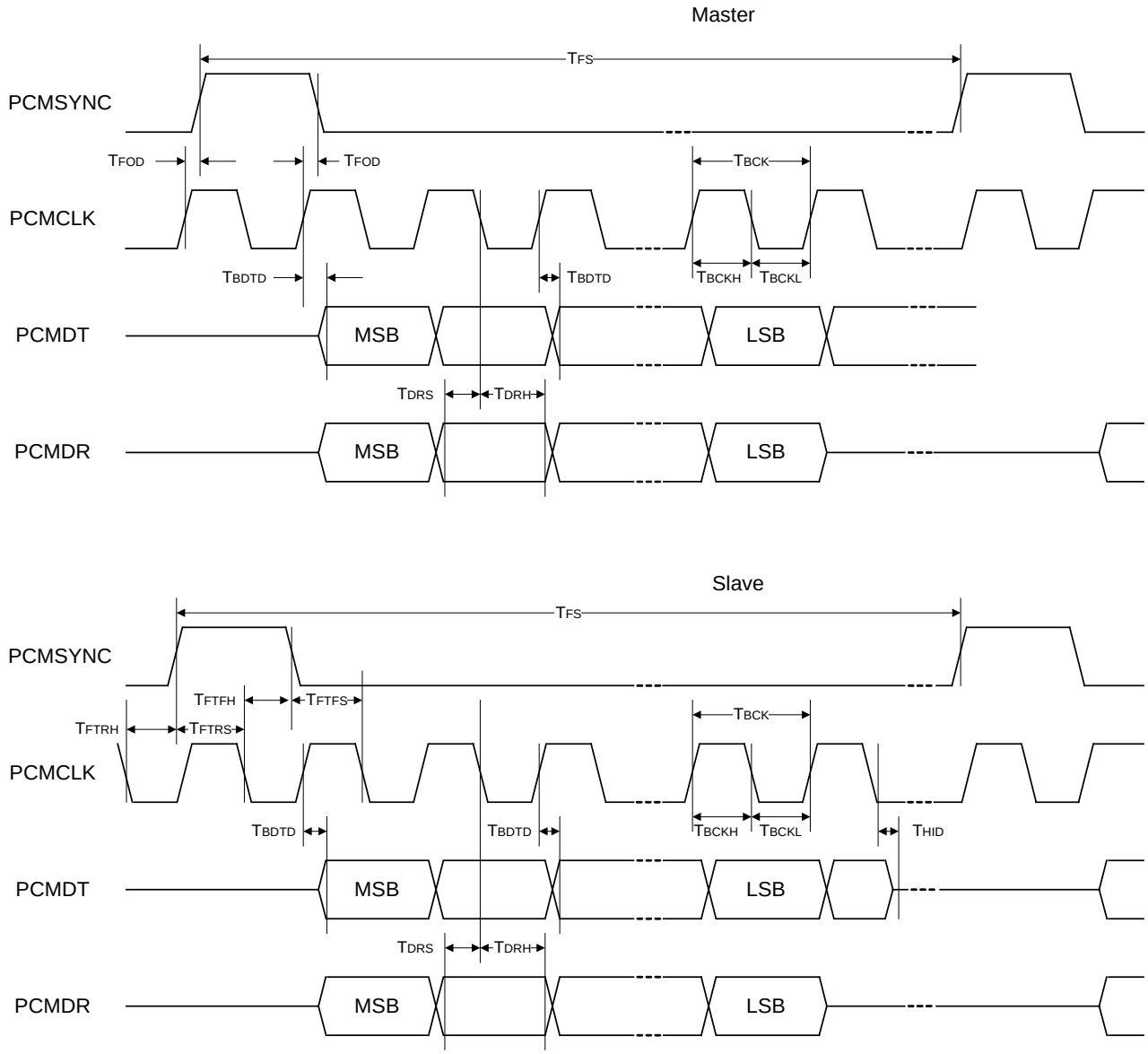


Figure 42. Short Frame Sync PCM

Table 407. Short Frame Sync PCM Timing

Symbol	Parameter	Min	Typ	Max	Unit	Notes
$1 / T_{FS}$	PCMSYNC period		8.0		KHz	
$1 / T_{BCK}$	PCMCLK period	128		4096	KHz	1, 2
T_{BCKH}	PCMCLK high pulse width	0.45	0.5	0.55	T_{BCK}	
T_{BCKL}	PCMCLK low pulse width	0.45	0.5	0.55	T_{BCK}	
T_{DRS}	PCMDR to PCMCLK fall edge setup time	5.5			ns	2
T_{DRH}	PCMDR to PCMCLK fall edge hold time	0.0			ns	2

T_{FOD}	PCMCLK rise edge to PCMSYNC edge delay time	0.9		5.4	ns	2
T_{BDTD}	PCMCLK rise edge to valid PCMDT delay time	0.4		8.7	ns	2
$1 / T_{BCK}$	PCMCLK frequency	128		2048	KHz	1, 3
T_{DRS}	PCMDR to PCMCLK fall edge setup time	0.0			ns	3
T_{DRH}	PCMDR to PCMCLK fall edge hold time	0.5			ns	3
T_{HID}	PCMCLK rise edge to high impedance delay time	2.9		13.3	ns	3
T_{BDTD}	PCMCLK rise edge to valid PCMDT delay time	2.9		13.3	ns	3
T_{FTRH}	PCMCLK fall edge to PCMSYNC rise edge hold time	0.2			ns	3
T_{FTRS}	PCMSYNC rise edge to PCMCLK fall edge setup time	3.9			ns	3
T_{FTFH}	PCMSYNC rise edge to PCMCLK fall edge setup time	0.2			ns	3
T_{FRFS}	PCMSYNC fall edge to PCMCLK fall edge setup time	3.9			ns	3

Note:

1. The minimum period of T_{CYC} is around 244ns. The clock rate can be configured by bit [2:0] of "PCM Configuration Register".
2. It is for master. In master mode, PCMCLK, PCMSYNC and PCMDT are outputs; and PCMDR is input.
3. It is for slave. In slave mode, PCMDT is output; and PCMCLK, PCMSYNC and PCMDR are inputs.

4.6.7.3 GCI PCM

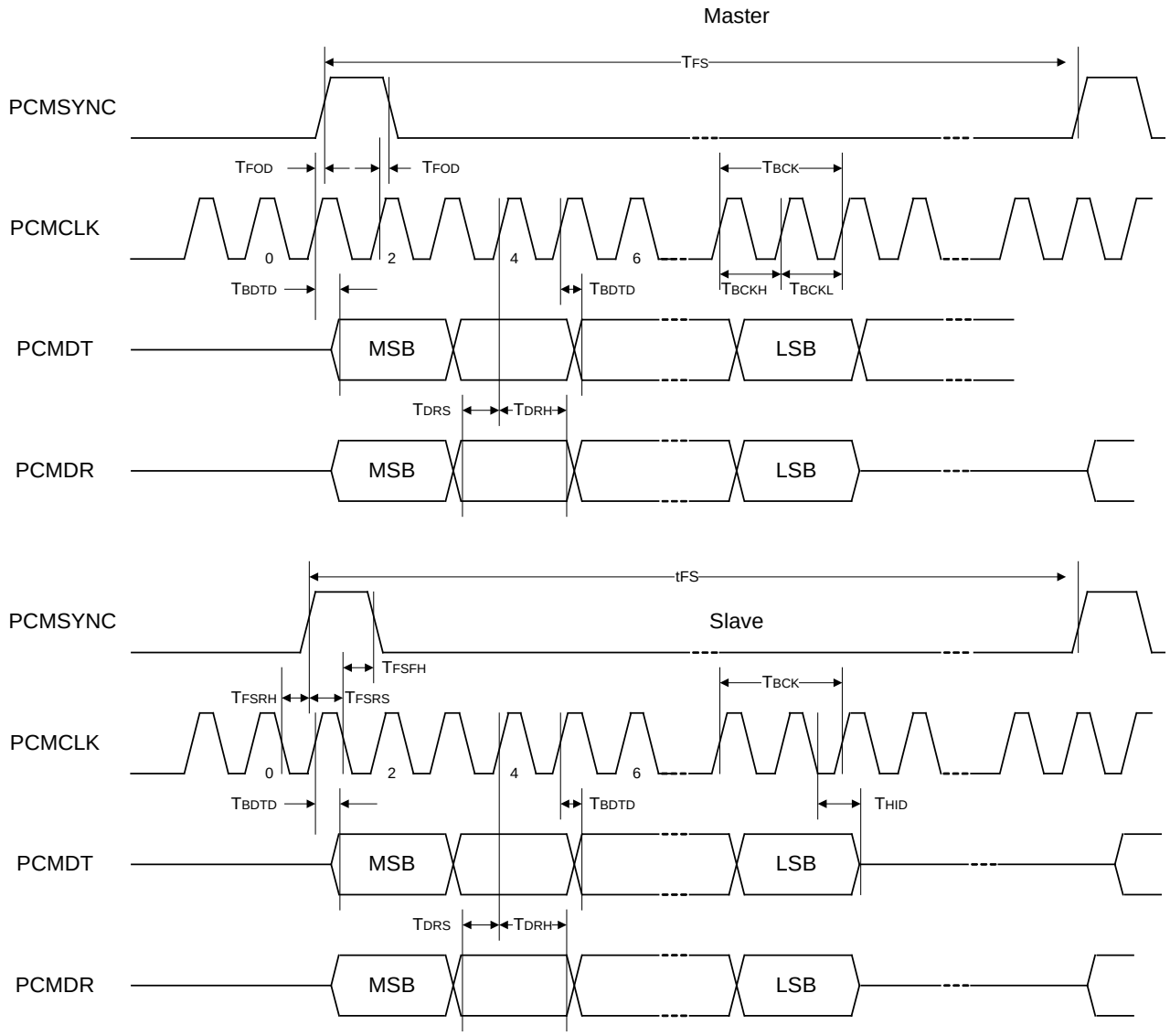


Figure 43. GCI PCM

Table 408. GCI PCM Timing

Symbol	Parameter	Min	Typ	Max	Unit	Notes
$1 / T_{FS}$	PCMSYNC frequency		8.0		KHz	
$1 / T_{BCK}$	PCMCLK frequency	256		4096	KHz	1, 2
T_{BCKH}	PCMCLK high pulse width	0.45	0.5	0.55	T_{BCK}	
T_{BCKL}	PCMCLK low pulse width	0.45	0.5	0.55	T_{BCK}	
T_{DRS}	PCMDR to PCMCLK fall edge setup time	4.4			ns	2
T_{DRH}	PCMDR to PCMCLK fall edge hold time	0.0			ns	2
T_{FOD}	PCMCLK rise edge to PCMSYNC edge delay time	0.9		5.4	ns	2

T_{BDTD}	PCMCLK rise edge to valid PCMDT delay time	0.4		8.7	ns	2
$1 / T_{BCK}$	PCMCLK frequency	256		2048	KHz	1, 3
T_{DRS}	PCMDR to PCMCLK fall edge setup time	0.0			ns	3
T_{DRH}	PCMDR to PCMCLK fall edge hold time	0.5			ns	3
T_{HID}	PCMCLK rise edge to high impedance delay time	2.9		13.3	ns	3
T_{BDTD}	PCMCLK rise edge to valid PCMDT delay time	2.9		13.3	ns	3
T_{FSRH}	PCMCLK fall edge to PCMSYNC rise edge hold time	0.2			ns	3
T_{FSRS}	PCMSYNC rise edge to PCMCLK fall edge setup time	3.9			ns	3
T_{FSFH}	PCMSYNC rise edge to PCMCLK fall edge setup time	0.2			ns	3

Note:

1. The minimum period of T_{CYC} is around 244ns. The clock rate can be configured by bit [2:0] of "PCM Configuration Register".
2. It is for master. In master mode, PCMCLK, PCMSYNC and PCMDT are outputs; and PCMDR is input.
3. It is for slave. In slave mode, PCMDT is output; and PCMCLK, PCMSYNC and PCMDR are inputs.

4.6.8 I2S/LJF/RJF Interface Timing

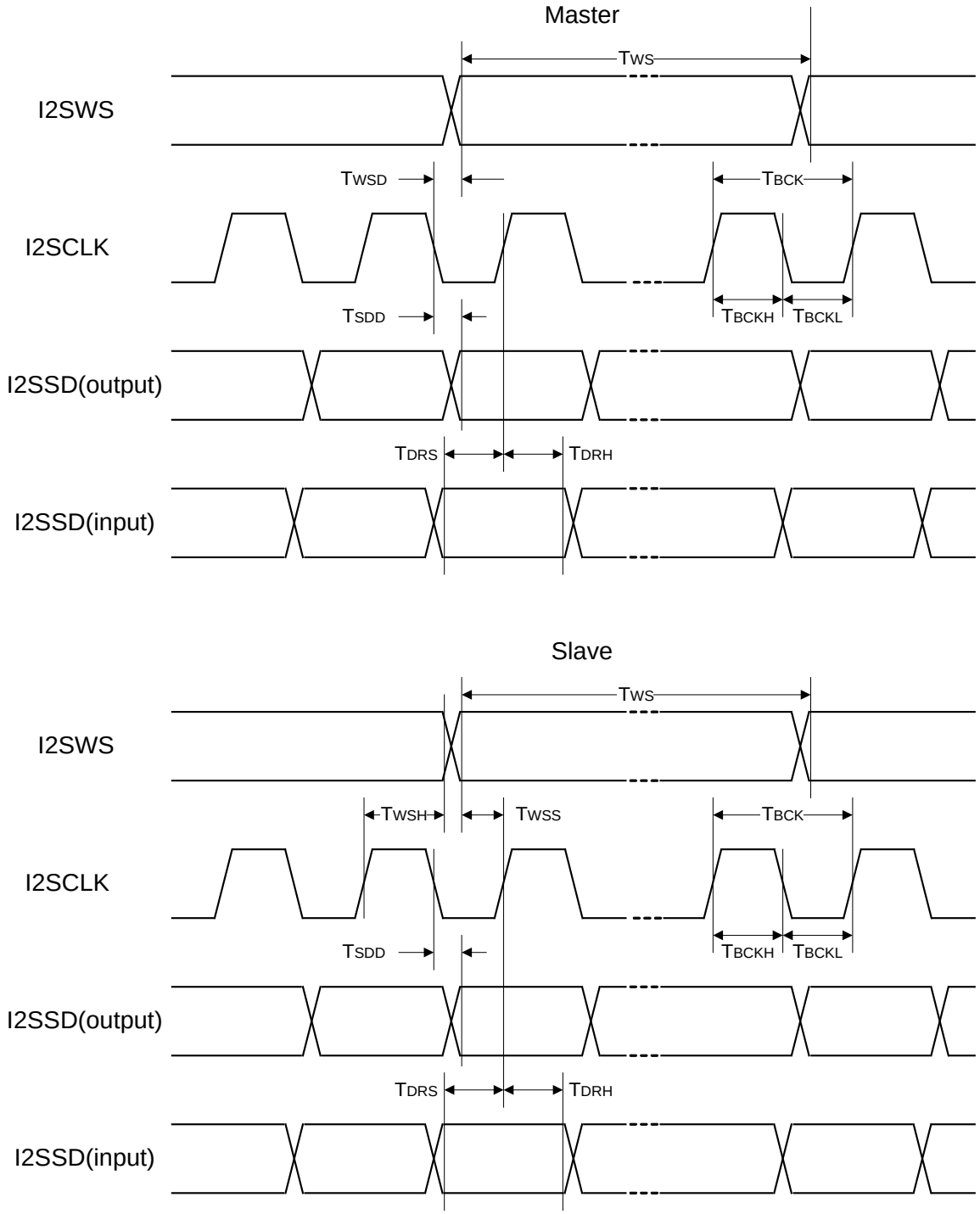


Figure 44. I2S/LJF/RJF Interface Timing Diagram

Table 409. I2S/LJF/RJF Interface Timing

Symbol	Parameter	Min	Typ	Max	Unit	Notes
$1/T_{ws}$	I2SWS frequency			48 44.1	KHz	1

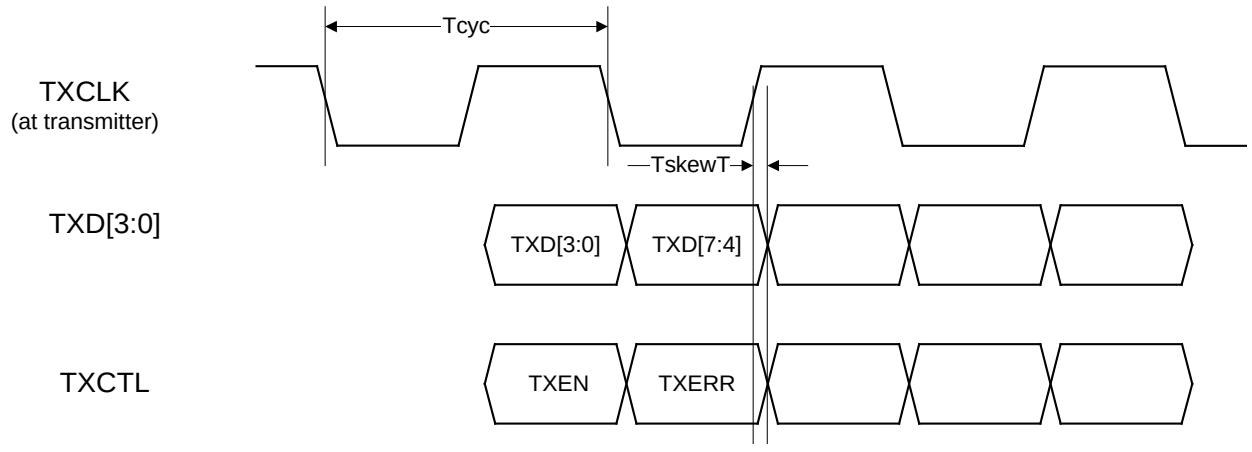
				32		
$1 / T_{BCK}$	I2SCLK frequency	1536 1208 1024		12288 11289.6 8192	KHz	2, 3
T_{BCKH}	I2SCLK high pulse width	0.4	0.5	0.6	T_{BCK}	
T_{BCKL}	I2SCLK low pulse width	0.4	0.5	0.6	T_{BCK}	
T_{DRS}	I2SSD to I2SCLK rising edge setup time	9.1			ns	3
T_{DRH}	I2SSD to I2SCLK rising edge hold time	0.0			ns	3
T_{WSD}	I2SCLK falling edge to I2SWS edge delay time	0.4		5.8	ns	3
T_{SDD}	I2SCLK falling edge to valid I2SSD delay time	1.2		3.5	ns	3
$1 / T_{BCK}$	I2SCLK frequency	1536 1208 1024		6144 5644.8 4096	KHz	2, 4
T_{SDD}	I2SCLK falling edge to valid I2SSD delay time	4.7		12.4	ns	4
T_{DRS}	I2SSD input to I2SCLK rising edge setup time	1.4			ns	4
T_{DRH}	I2SSD input to I2SCLK rising edge hold time	0.2			ns	4
T_{WSS}	I2SWS edge to I2SCLK rising edge setup time	3.5			ns	4
T_{WSH}	I2SWS edge to I2SCLK rising edge hold time	0.4			ns	4

Note:

1. The audio sampling frequency can be configured by bit [13:12] of “System Clock Control Register”.
2. The clock rate can be configured by bit [5:4] of “I2S Configuration Register”.
3. It is for master. And in master mode, I2SCLK and I2SWS are outputs; and I2SSD may be input or output.
4. It is for slave. And in slave mode, I2SCLK and I2SWS are inputs; and I2SSD may be input or output.

4.6.9 GEC Interface Timing

4.6.9.1 RGMII



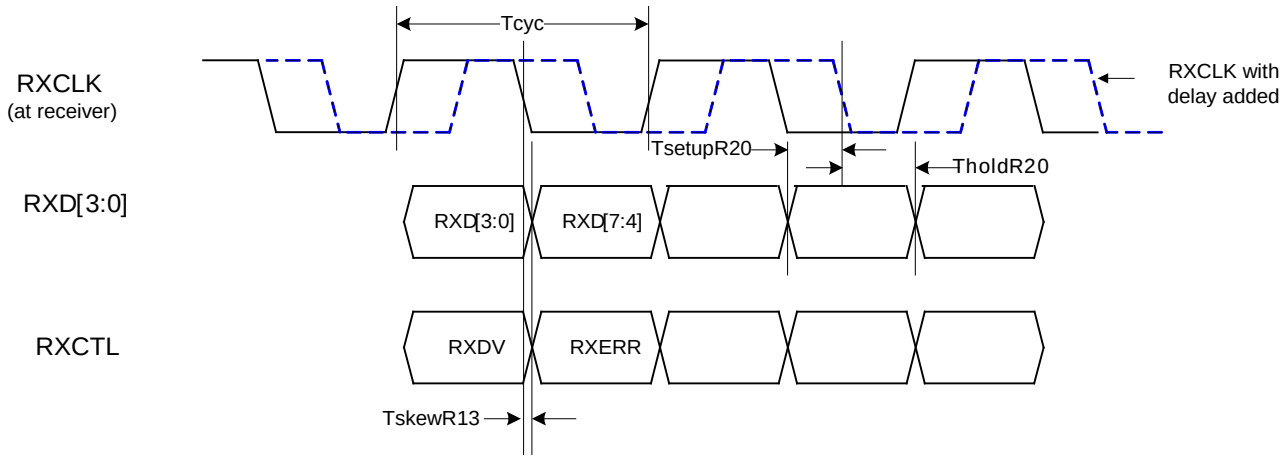


Figure 45. RGMII Interface Timing Diagram

Table 410. RGMII Interface Timing

Symbol	Parameter	Min	Typ	Max	Unit	Notes
TskewT	Data to Clock output Skew(at transmitter) (tx_skew = 0x02, Note 1)	-0.3		0.2	ns	1
TskewR13	Data to Clock input skew (at receiver) (rx_skew = 0x2, Note 2)	-0.4		0.6	ns	2
TsetupR20	Data to Clock input setup time (at receiver) (rx_skew = 0x0, Note 2)	0.3			ns	2
TholdR20	Data to Clock input hold (rx_skew = 0x0, Note 2)	0.8			ns	2
Tcyc	Clock cycle	7.2	8.0	8.8	ns	
Duty_G	Duty Cycle for Gigabit	45	50	55	%	
Duty_T	Duty Cycle for 10/100T	40	50	60	%	
Tr/Tf	Rise/Fall Time (20-80%)			0.75	ns	3

Note:

1. Skew of RGMII TX clock and TX data/enable can be fine tuned by “tx_skew” (bit [3:2] of "Test 0 Register" of NIC Controller block) as follows,
   - 0x0: -0.5 ns
   - 0x1: -0.5 ns
   - 0x2: 0.0 ns (default)
   - 0x3: 0.5 ns
2. Skew of RGMII RX clock and RX data/enable can be fine tuned by “rx_skew” (bit [1:0] of "Test 0 Register" of NIC Controller block) as follows,
   - 0x0: 0.0 ns (This is recommended value for RGMII 2.0 spec. with no PCB trace delay)
   - 0x1: 1.5 ns
   - 0x2: 2.0 ns (This is recommended value for RGMII 1.3 spec. with no PCB trace delay)
   - 0x3: 2.5 ns
3. The Rise/Fall Time is under condition of 5pF external loading.

4.6.9.2 MII Interface Timing

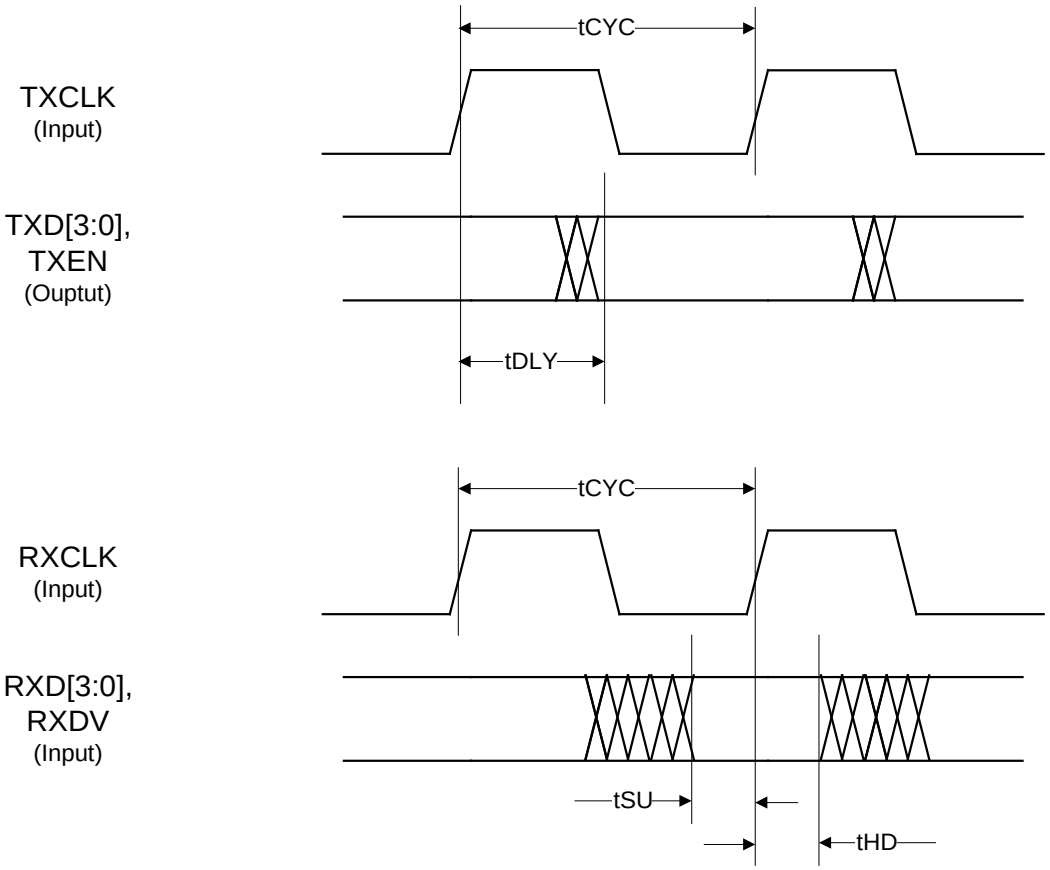


Figure 46. MII Interface Timing Diagram

Table 411. MII Interface Timing

Symbol	Parameter	Min	Typ	Max	Unit	Notes
t_{DLY}	Data to Clock Output Delay (at transmitter)	3.0		9.0	ns	
t_{SU}	Data to Clock Setup Time (at receiver)	2.0			ns	
t_{HD}	Data to Clock Hold Time (at receiver)	1.0			ns	
t_{CYC}	Clock Cycle		40(100T) 400(10T)		ns	1

Note:
1. For 100M MAC, the period is 40ns. For 10M MAC, the period is 400ns.

4.6.9.3 Reverse MII

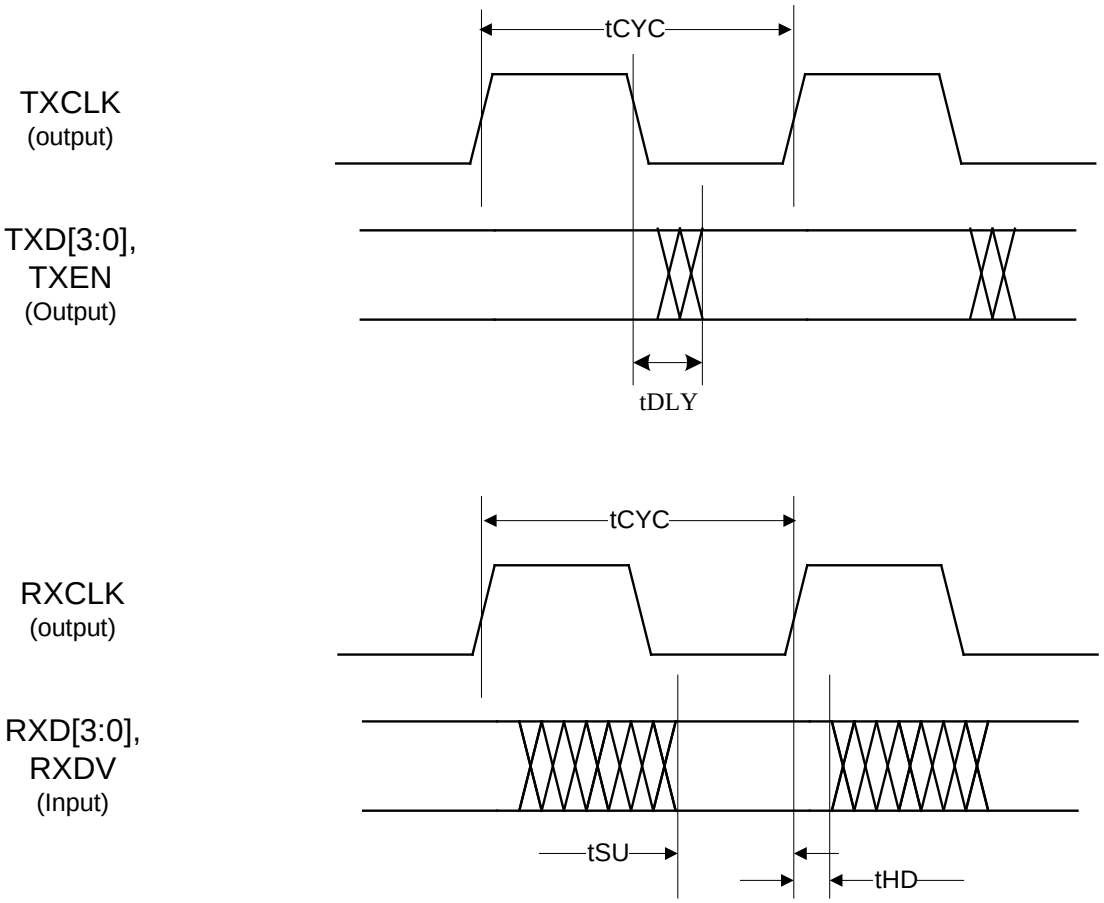


Figure 47. Reverse MII Interface Timing Diagram

Table 412. Reverse MII Interface Timing

Symbol	Parameter	Min	Typ	Max	Unit	Notes
t_{DLY}	Data to Clock Falling Edge Output Delay(at transmitter)	0		3	ns	
t_{SU}	Data to Clock Setup Time(at receiver)	7.0			ns	
t_{HD}	Data to Clock Hold Time(at receiver)	0.0			ns	
t_{CYC}	Clock Cycle		40(100T) 400(10T)		ns	1

Note:
1: For 100M MAC, the period is 40ns. For 10M MAC, the period is 400ns.

4.6.9.4 MDC/MDIO

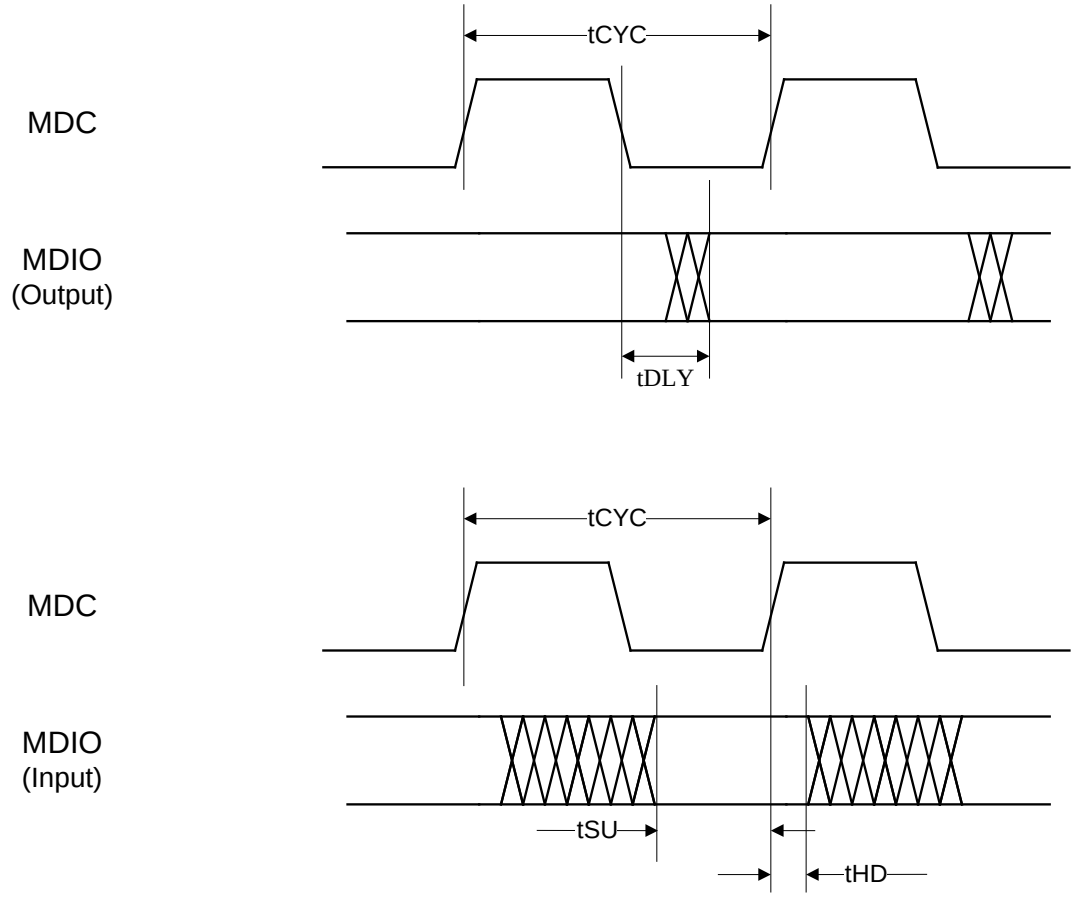


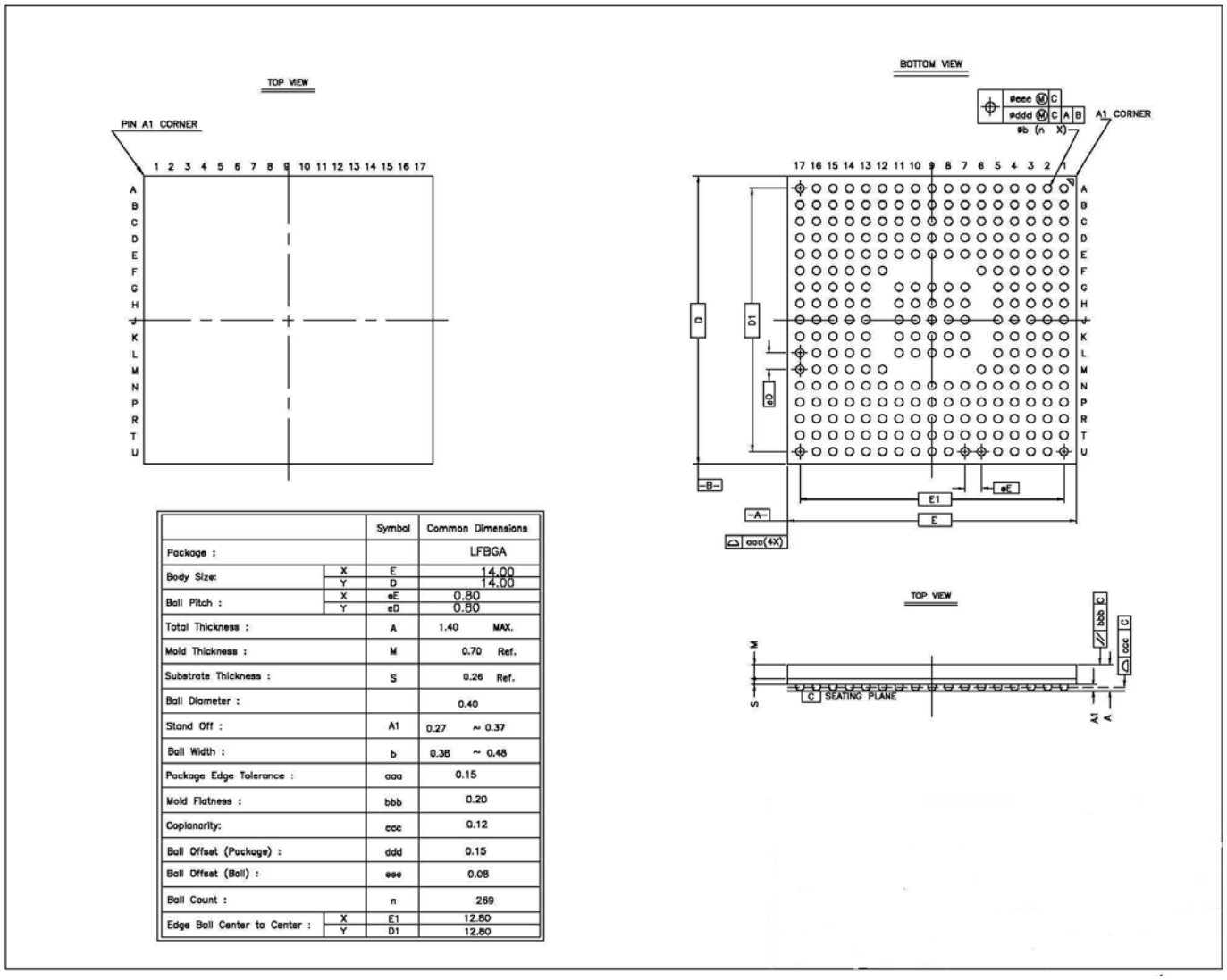
Figure 48. MDC/MDIO Timing Diagram

Table 413. MDC/MDIO Timing

Symbol	Parameter	Min	Typ	Max	Unit	Notes
t_{DLY}	Data to Clock Falling Edge Output Delay (at transmitter)	0		20	ns	
t_{SU}	Data to Clock Setup Time (at receiver)	8.0			ns	
t_{HD}	Data to Clock Hold Time (at receiver)	2.0			ns	
t_{CYC}	Clock Cycle		400		ns	

5 Mechanical Specifications

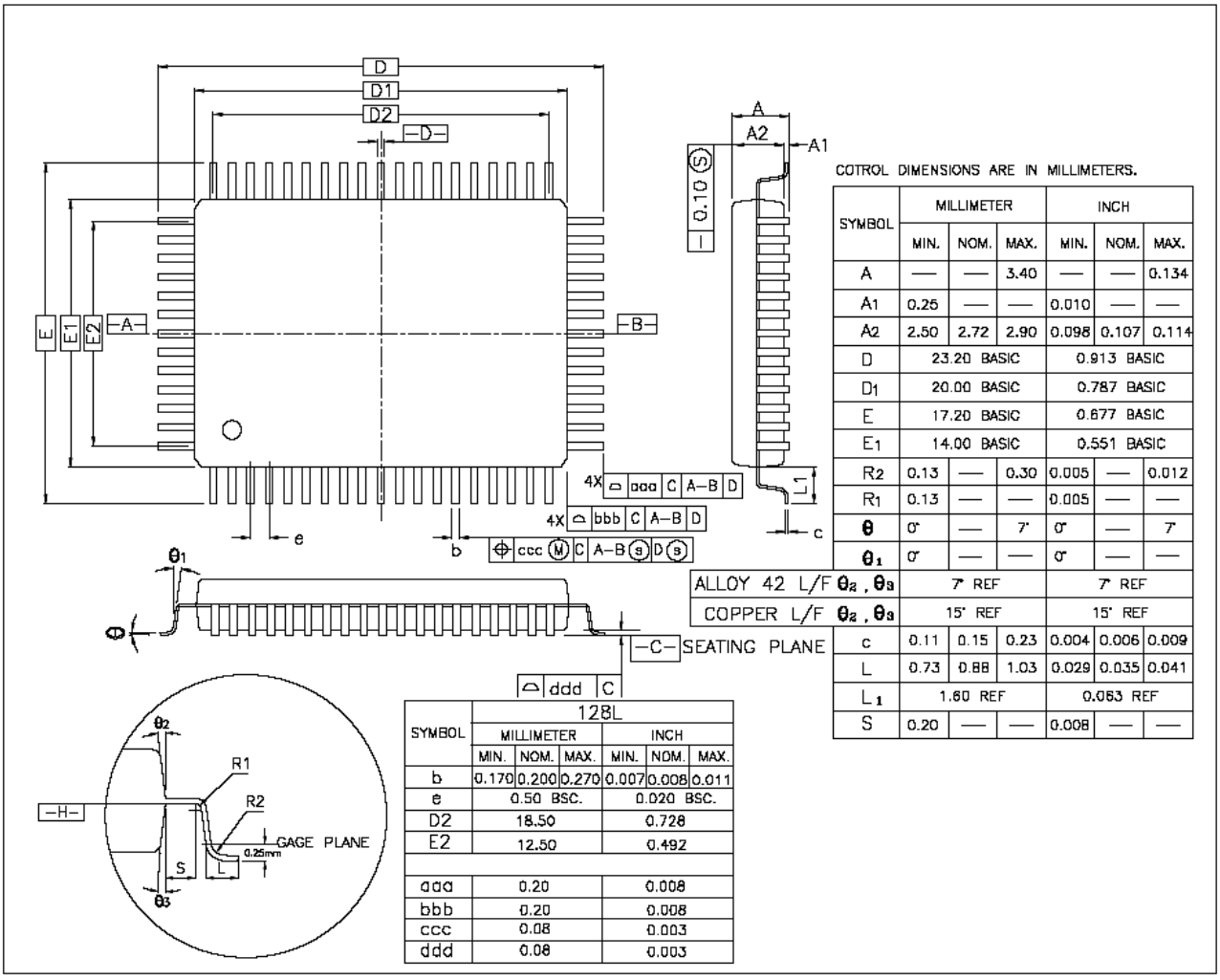
5.1 LFBGA-269 Package Outline and Dimension (for STR8133/CNS2133)



		Symbol	Common Dimensions
Package :			LFBGA
Body Size:	X	E	14.00
	Y	D	14.00
Ball Pitch :	X	eE	0.80
	Y	eD	0.80
Total Thickness :		A	1.40 MAX.
Mold Thickness :		M	0.70 Ref.
Substrate Thickness :		S	0.26 Ref.
Ball Diameter :			0.40
Stand Off :		A1	0.27 ~ 0.37
Ball Width :		b	0.38 ~ 0.48
Package Edge Tolerance :		aaa	0.15
Mold Flatness :		bbb	0.20
Coplanarity:		ccc	0.12
Ball Offset (Package) :		ddd	0.15
Ball Offset (Ball) :		eee	0.08
Ball Count :		n	269
Edge Ball Center to Center :	X	E1	12.80
	Y	D1	12.80

Figure 49. STR8133/CNS2133 Package Outline and Dimensions-LFBGA-269

5.2 PQFP-128 Package Outline and Dimension (for STR8131/CNS2131, STR8132/CNS2132, STR8181/CNS2181)



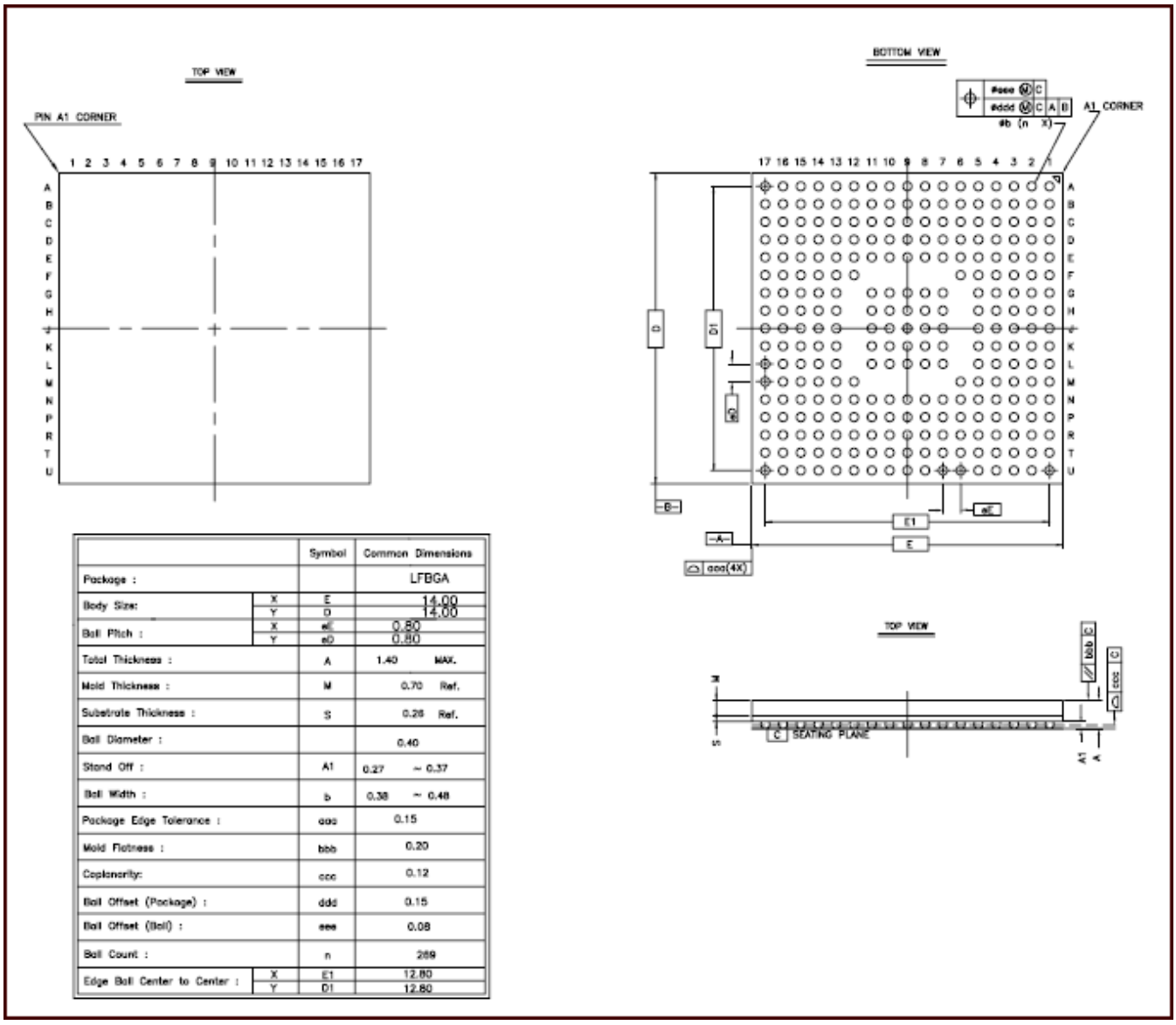
COTROL DIMENSIONS ARE IN MILLIMETERS.

SYMBOL	MILLIMETER			INCH		
	MIN.	NOM.	MAX.	MIN.	NOM.	MAX.
A	—	—	3.40	—	—	0.134
A1	0.25	—	—	0.010	—	—
A2	2.50	2.72	2.90	0.098	0.107	0.114
D	23.20 BASIC			0.913 BASIC		
D1	20.00 BASIC			0.787 BASIC		
E	17.20 BASIC			0.677 BASIC		
E1	14.00 BASIC			0.551 BASIC		
R2	0.13	—	0.30	0.005	—	0.012
R1	0.13	—	—	0.005	—	—
θ	0°	—	7°	0°	—	7°
θ1	0°	—	—	0°	—	—
ALLOY 42 L/F θ2, θ3	7° REF			7° REF		
COPPER L/F θ2, θ3	15° REF			15° REF		
c	0.11	0.15	0.23	0.004	0.006	0.009
L	0.73	0.88	1.03	0.029	0.035	0.041
L1	1.60 REF			0.063 REF		
S	0.20	—	—	0.008	—	—

SYMBOL	128L					
	MILLIMETER			INCH		
	MIN.	NOM.	MAX.	MIN.	NOM.	MAX.
b	0.170	0.200	0.270	0.007	0.008	0.011
e	0.50 BSC.			0.020 BSC.		
D2	18.50			0.728		
E2	12.50			0.492		
aaa	0.20			0.008		
bbb	0.20			0.008		
ccc	0.08			0.003		
ddd	0.08			0.003		

Figure 50. STR8131/CNS2131, STR8132/CNS2132, STR8181/CNS2181 Package Outline and Dimensions— PQFP-128

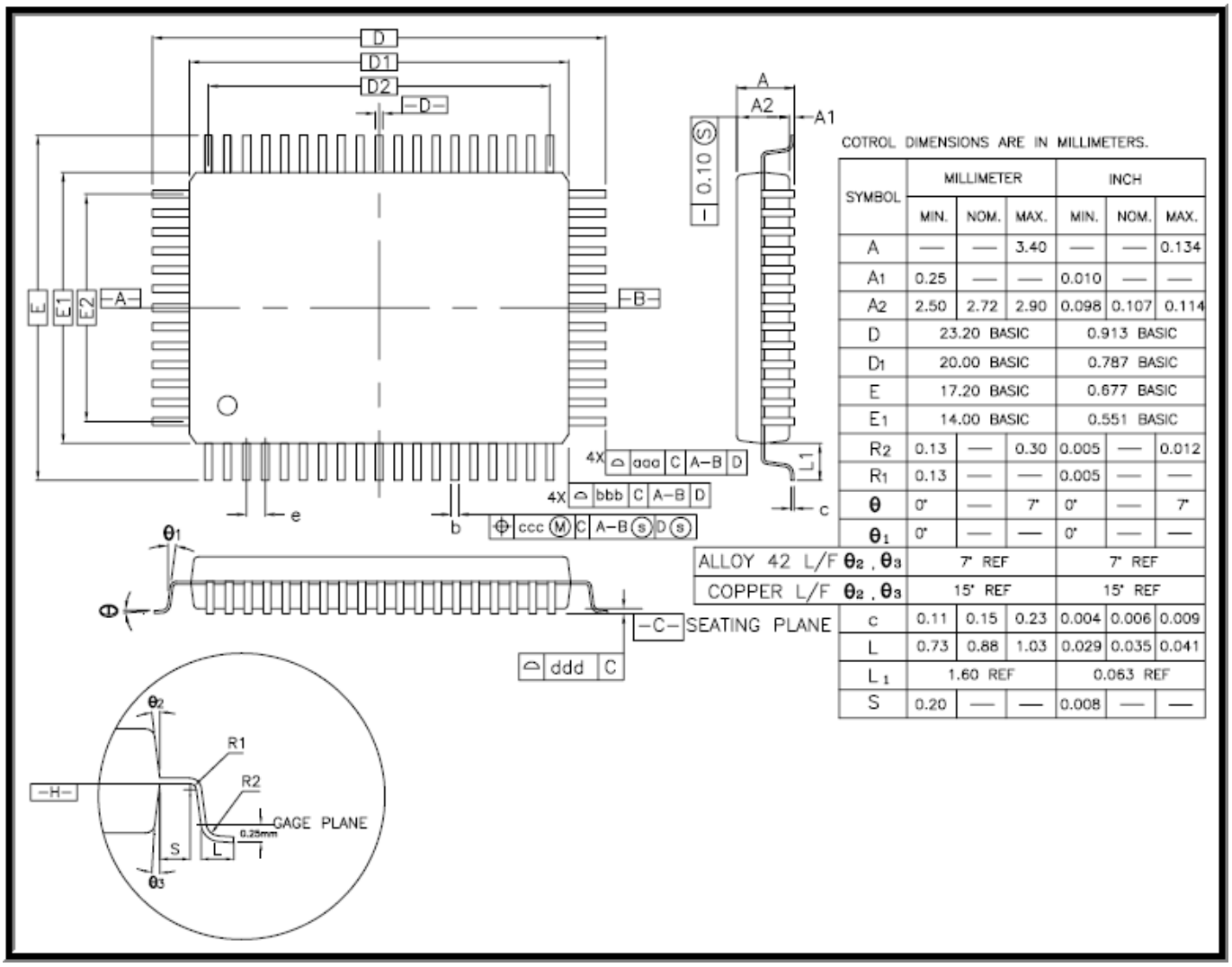
5.3 LFBGA-269 Package Outline and Dimension (for STR8182/CNS2182, STR8182X/CNS2182X)



		Symbol	Common Dimensions
Package :			LFBGA
Body Size:	X	E	14.00
	Y	D	14.00
Ball Pitch :	X	eE	0.80
	Y	eD	0.80
Total Thickness :		A	1.40 MAX.
Mold Thickness :		M	0.70 Ref.
Substrate Thickness :		S	0.28 Ref.
Ball Diameter :			0.40
Stand Off :		A1	0.27 ~ 0.37
Ball Width :		b	0.38 ~ 0.48
Package Edge Tolerance :		aaa	0.15
Mold Flatness :		bbb	0.20
Coplanarity:		ccc	0.12
Ball Offset (Package) :		ddd	0.15
Ball Offset (Ball) :		eee	0.08
Ball Count :		n	269
Edge Ball Center to Center :	X	E1	12.80
	Y	D1	12.80

Figure 51. STR8182/CNS2182, STR8182X/CNS2182X Package Outline and Dimensions-LFBGA-269

5.4 QFP-128 Package Outline and Dimension (for CNS2135)



COTROL DIMENSIONS ARE IN MILLIMETERS.

SYMBOL	MILLIMETER			INCH		
	MIN.	NOM.	MAX.	MIN.	NOM.	MAX.
A	—	—	3.40	—	—	0.134
A1	0.25	—	—	0.010	—	—
A2	2.50	2.72	2.90	0.098	0.107	0.114
D	23.20 BASIC			0.913 BASIC		
D1	20.00 BASIC			0.787 BASIC		
E	17.20 BASIC			0.677 BASIC		
E1	14.00 BASIC			0.551 BASIC		
R2	0.13	—	0.30	0.005	—	0.012
R1	0.13	—	—	0.005	—	—
θ	0°	—	7°	0°	—	7°
θ1	0°	—	—	0°	—	—
ALLOY 42 L/F θ2 , θ3	7° REF			7° REF		
COPPER L/F θ2 , θ3	15° REF			15° REF		
c	0.11	0.15	0.23	0.004	0.006	0.009
L	0.73	0.88	1.03	0.029	0.035	0.041
L 1	1.60 REF			0.063 REF		
S	0.20	—	—	0.008	—	—

Figure 52. CNS2135 Package Outline and Dimensions— PQFP-128

6 Part Order Numbering and Package Marking

The following figure is an example of the part order numbering scheme for the STR813X/CNS213X , STR818X/CNS218X family.

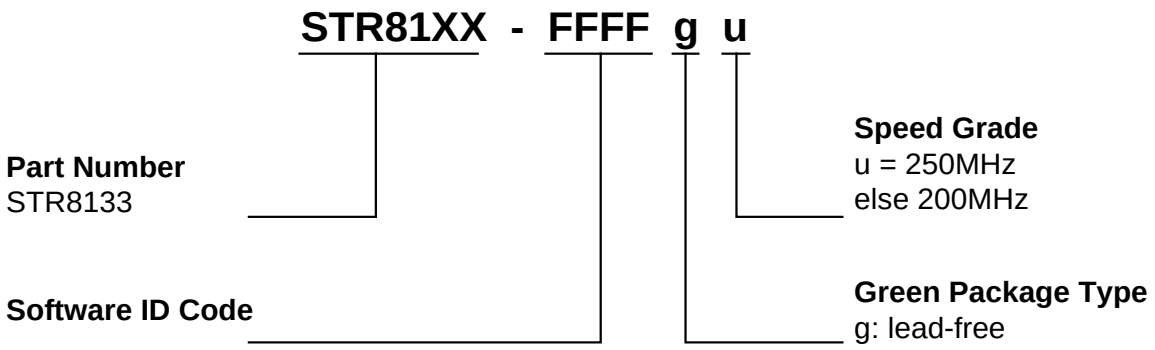


Figure 53. Sample Part Order Number (STAR Part Number)

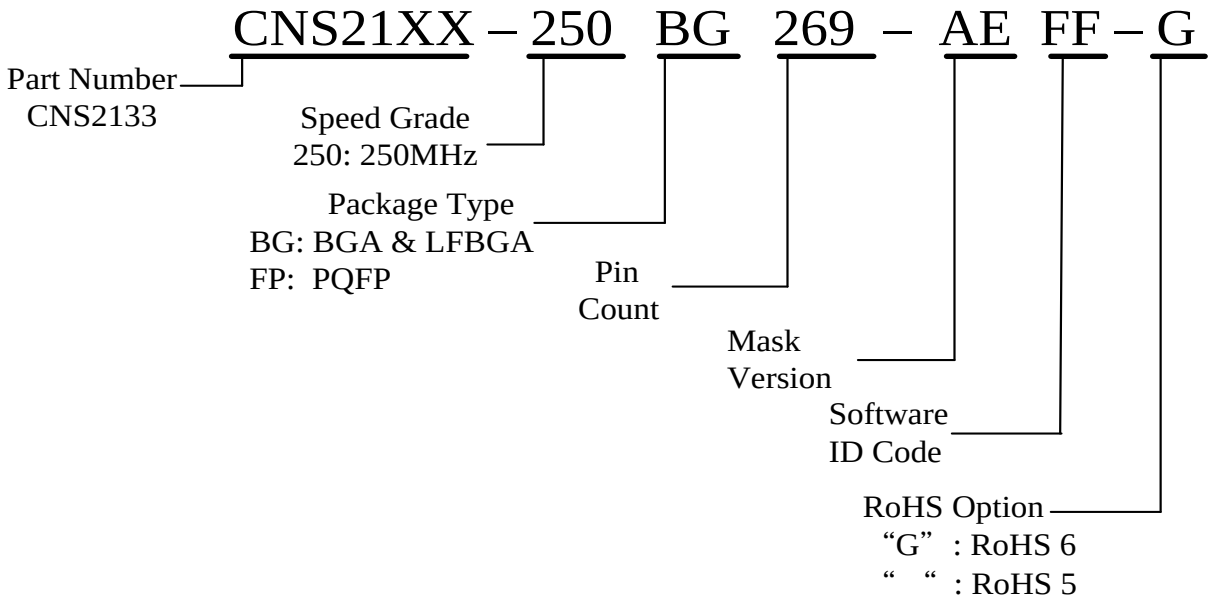


Figure 54. Sample Part Order Number(Cavium Part Number)

Table 414. STR813X/CNS213X and STR818X/CNS218X Part Order Number List

Part Number	CPU Clock
CNS2131-250FP128-AEFF-G/STR8131-FFFFgu	250MHz
CNS2132-250FP128-AEFF-G/STR8132-FFFFgu	250MHz
CNS2133-250BG269-AEFF-G/STR8133-FFFFgu	250MHz
CNS2181-250FP128-AEFF-G/STR8181-FFFFgu	250MHz
CNS2182-250BG269-AEFF-G/STR8182-FFFFgu	250MHz
CNS2182X-250BG269-AEFF-G/STR8182-FFFFXgu	250MHz
CNS2131-200FP128-AEFF-G/STR8131-FFFFg	200MHz
CNS2132-200FP128-AEFF-G/STR8132-FFFFg	200MHz
CNS2133-200BG269-AEFF-G/STR8133-FFFFg	200MHz
CNS2135-200FH128-AEFF-G	200MHz
CNS2181-200FP128-AEFF-G/STR8181-FFFFg	200MHz
CNS2182-200BG269-AEFF-G/STR8182-FFFFg	200MHz
CNS2182X-200BG269-AEFF-G/STR8182-FFFFXg	200MHz

Note:

In terms of STR8182X's series, STR8182-FFFFXg and STR8182-FFFFXgu are existent part numbers.

The following figure is an example of the package marking and pin 1 location for the STR8131/CNS2131 package. STR8132/CNS2132, STR8181/CNS2181 (PQFP-128) has the same format of package marking.

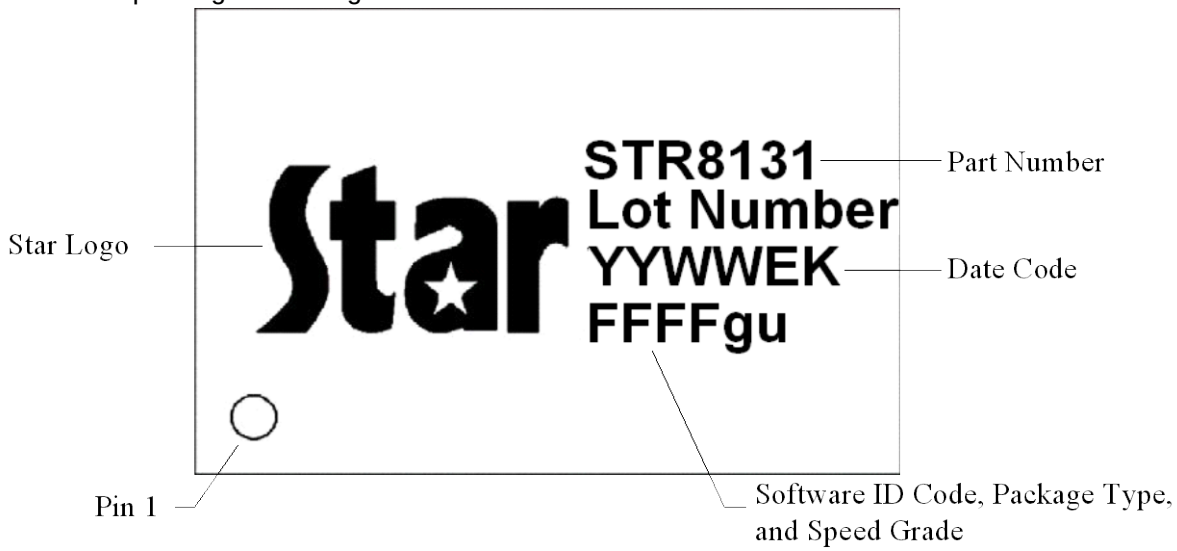


Figure 55. STR8131/CNS2131 (PQFP-128) Package Marking and Pin 1 Location (STAR Part Number)

The following figure is an example of the package marking and pin 1 location for the STR8133/CNS2133 package. STR8182/CNS2182, STR8182X/CNS2182X (LFBGA-269) have the same format of package marking.

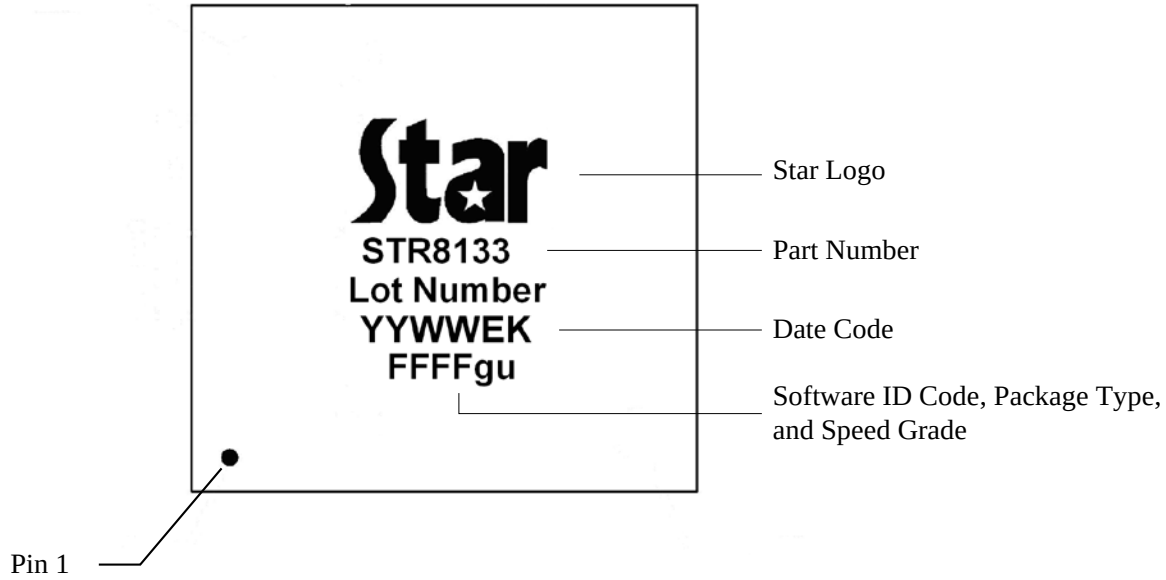


Figure 56. STR8133/CNS2133 (LFBGA-269) Package Marking and Pin 1 Location (STAR Part Number)

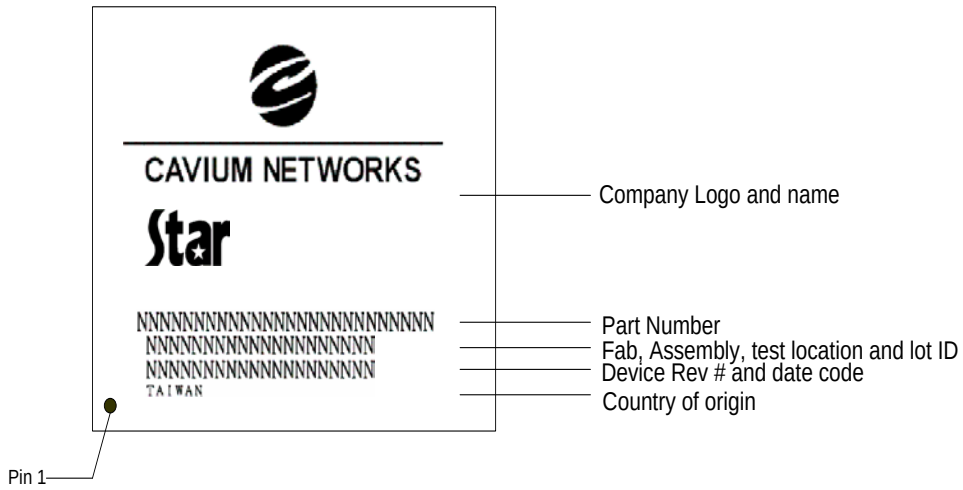


Figure 57. STR81XX /CNS21XX Package Marking and Pin 1 Location (Cavium Part Number)