32-BIT ARM926EJ-S BASED MCU

NUC946ADN
32-bit ARM926EJ-S Based Microcontroller
Product Data Sheet

Publication Release Date: July 26, 2011
Revision: A5

nuvoTon

32-BIT ARM926EJ-S BASED MCU

Table of Contents

32-BIT ARM926EJ-S BASED MCU

1 General Description

This chip is built around an outstanding CPU core: the 16/32 ARM926EJS RISC processor designed by Advanced RISC Machines, Ltd. The ARM926EJS core, offers 8K-byte I-cache and 8K-byte D-cache with MMU, is a low power, general-purpose integrated circuits. One 10/100 Mb MAC of Ethernet controller is built-in to reduce total system cost. This micro-controller is suitable for a high end, high performance and low cost related products as well as general purpose applications.

The following integrated on-chip functions are described in detail in this document.

Main Function of NUC946ADN
• ARM926EJS CPU with 8K I-Cache and 8K D-Cache
• Ethernet MAC Controllers
• External Bus Interface Controller
• GDMA Controller
• Timers and Watchdog Timer
• Programmable I/O Ports
• Advanced Interrupt Controller
• USB Host Controller
• USB Device Controller
• USI (SPI/MicroWire) Controller
• I2C Controller
• Programmable PLL System Clock Synthesizer
• SD/SDIO Host Controller

2 Features

Architecture
- Efficient and powerful ARM926EJS core with fully 16/32-bit RISC architecture
- Little Endian mode supported
- 8K-byte I-cache and 8K-byte D-cache with MMU

External Bus Interface
- 8/16-bit external bus support for ROM/SRAM, flash memory, SDRAM and external I/Os
- Programmable access cycle (0-7 wait cycle)
- Four-word depth write buffer

Ethernet MAC Controller
- 100/10-Mbps operation
- DMA engine with burst mode
- MAC Tx/Rx buffers (256 bytes Tx, 256 bytes Rx)
- Full compliance with IEEE standard 802.3
- RMII interface only
- Station Management Signaling
- On-Chip CAM (up to 16 destination addresses)
- Full-duplex mode with PAUSE feature
- Long/short packet modes

General DMA Controller
- 2-channel General DMA for memory-to-memory data transfers without CPU intervention
- Increments or decrements a source or destination address in 8-bit or 16-bit data transfers
- 4-data burst mode

UART
- Two UART (serial I/O) blocks with interrupt-based operation
- Support for 5-bit, 6-bit, 7-bit or 8-bit serial data transmit and receive
- Programmable baud rates
- 1,1½ or 2 stop bits
- Odd or even parity
- Break generation and detection
- Parity, overrun and framing error detection
- X16 clock mode
- Support for IrDA and two debug ports

Timers
- Five programmable 24-bit timers with 8-bit pre-scalar
- One programmable 20-bit Watchdog timer
- One-short mode, period mode or toggle mode operation

Programmable I/Os
- Pins individually configurable to input, output or I/O mode for dedicated signals

- I/O ports are Programmable and Configurable for Multiple functions

Advanced Interrupt Controller

- 31 interrupt sources, including 3 external interrupt sources
- Programmable normal or fast interrupt mode (IRQ, FIQ)
- Programmable as either edge-triggered or level-sensitive for 3 external interrupt sources
- Programmable as either low-active or high-active for 3 external interrupt sources
- Priority methodology is encoded to allow for interrupt daisy-chaining
- Automatically mask out the lower priority interrupt during interrupt nesting
- Automatically clear the interrupt flag when the interrupt source is programmed to be edge-triggered

USB Host Controller with tranceiver

- Fully compliant with USB Revision 2.0 specification.
- Enhanced Host Controller Interface (EHCI) Revision 1.0 compatible.
- Open Host Controller Interface (OHCI) Revision 1.0 compatible.
- Supports high-speed (480Mbps), full-speed (12Mbps) and low-speed (1.5Mbps) USB devices.
- Supports Control, Bulk, Interrupt, Isochronous and Split transfers.
- Built-in DMA for real-time data transfer.
- Support two ports (one port transceiver is shared with USB Device Controller)

USB Device Controller with tranceiver

- Compliant with USB version 2.0 specification.
- Software control for device remote-wakeup.
- Supports 6 configurable IN/OUT endpoints in addition to Control Endpoint. Each of these endpoints can be configures as In or Out with Isochronous, Bulk or Interrupt transfer.
- Three different modes of operation of an in-endpoint (Auto validation mode, manual validation mode, Fly mode.
- Supports Endpoint Maximum Packet Size up to 1024 bytes.

PLL

- Supports one on-chip PLLs
- The external clock can be multiplied by on-chip PLL to provide high frequency system clock
- The input frequency range is 4-30MHz; 15MHz is preferred.
- Programmable clock frequency

I2C Master

- support master mode only
- Multi Master Operation
- Clock stretching and wait state generation
- Provide multi-byte transmit operation, up to 4 bytes can be transmitted in a single transfer
- Software programmable acknowledge bit
- Arbitration lost interrupt, with automatic transfer cancellation
- Start/Stop/Repeated Start/Acknowledge generation
- Start/Stop/Repeated Start detection
- Bus busy detection
- Supports 7 bit addressing mode
- Software mode I^2C

Universal Serial Interface (USI)

- Support MICROWIRE/SPI master mode
- Support full/half duplex synchronous serial data transfer
- Variable length of transfer word up to 32 bits
- Provide burst mode operation, transmit/receive can be executed up to four times in one transfer
- MSB or LSB first data transfer
- Receive and Transmit on both rising or falling edge of serial clock independently

Flash Memory Interface (FMI)

- Directly connect to Secure Digital (SD, MMC and SDIO) flash memory card and Memory Stick (Memory stick PRO).
- Supports DMA function to accelerate the data transfer between the internal buffer, external SDRAM, and flash memory card.
- Two 512 bytes internal buffers are embedded inside

Power management

- Programmable clock enable for individual peripherals
- IDLE mode to halt ARM Core and keep peripheral working
- Power-Down mode to stop all clocks included external crystal oscillator.
- Exit IDLE/Power-Down by interrupts

Operation Voltage Range

- 3.3 V for IO Buffer
- 1.8 V for Core Logic

Operation Temperature Range

- -40℃~+85℃

Operating Frequency

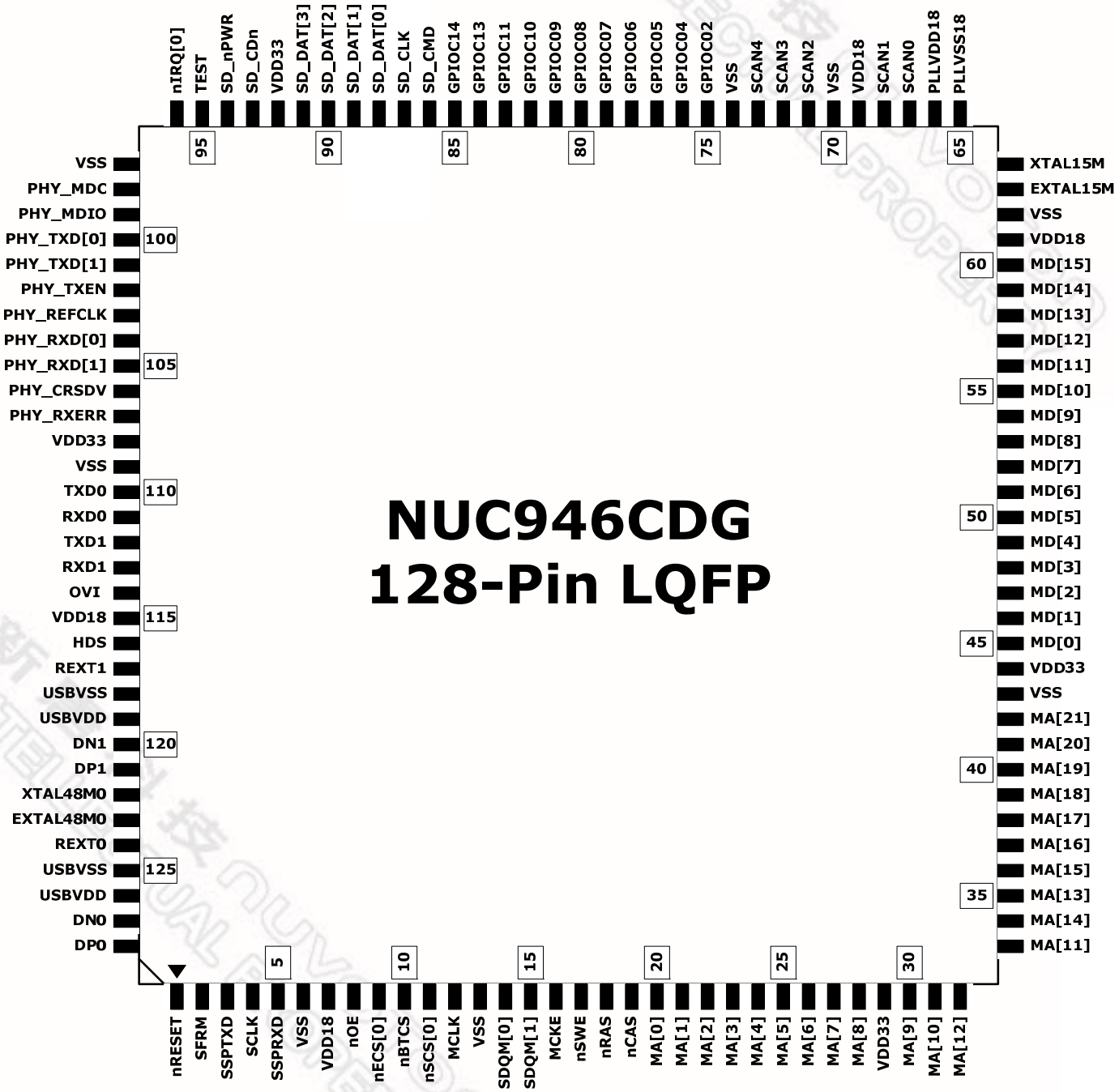
- Up to 200 MHz for ARM926EJS CPU

Package Type

- 128-Pin LQFP

3 Pin Diagram

NUC946ADN Pin Diagram



4 Pin Assignment

Table 4.1 NUC946 Pins Assignment

Pad Name	NUC946
Clock & Reset	**(5 pins)**
EXTAL15M	63
XTAL15M	64
EXTAL48M0	123
XTAL48M0	122
nRESET	1
External Bus Interface	**(49 pins)**
MA [21:0]	42 - 30 28 - 20
MD [15:0]	60 – 45
nWBE [1:0] / SDQM [1:0]	15 - 14
nSCS0	11
nSRAS	18
nSCAS	19
MCKE	16
nSWE	17
MCLK	12
nBTCS	10
nECS0	9
nOE	8

Table 4.1 NUC946 Pins Assignment (Continued)

Pad Name	NUC946
Ethernet Interface	**(10 pins)**
PHY_MDC / GPIOF[0]	98
PHY_MDIO / GPIOF[1]	99
PHY_TXD [1:0] / GPIOF[3:2]	101-100
PHY_TXEN / GPIOF[4]	102
PHY_REFCLK / GPIOF[5]	103
PHY_RXD [1:0] / GPIOF[7:6]	105-104
PHY_CRSDV / GPIOF[8]	106
PHY_RXERR / GPIOF[9]	107
USB Interface	**(8 pins)**
DP0	128
DN0	127
REXT0	124
OVI	114
HDS	116
DP1	121
DN1	120
REXT1	116

Table 4.1 NUC946 Pins Assignment (Continued)

Pad Name	NUC946
I2C/USI(SPI/MW)	(4 pins)
SCL0 / SFRM / GPIOG[0]	2
SDA0 / SSPTXD / GPIOG[1]	3
SCL1 / SCLK / GPIOG[2]	4
SDA1 / SSPRXD / GPIOG[3]	5
Pad Name	**NUC946**
UART	(4 pins)
TXD0 / GPIOE[0]	110
RXD0 / GPIOE[1]	111
TXD1(B) / GPIOE[2]	112
RXD1(B) / GPIOE[3]	113
Pad Name	**NUC946**
SDIO(SD)/ Memory Stick	(8 pins)
SD_CMD / MS_BS / GPIOD[0]	86
SD_CLK / MS_CLK / GPIOD[1]	87
SD_DAT0 / MS_DAT0 / GPIOD[2]	88
SD_DAT1 / MS_DAT1 / GPIOD[3]	89
SD_DAT2 / MS_DAT2 / GPIOD[4]	90
SD_DAT3 / MS_DAT3 / GPIOD[5]	91
SD_CDn / MS_CDn / GPIOD[6]	93
SD_nPWR / MS_nPWR / GPIOD[8]	94

Table 4.1 NUC946 Pins Assignment (Continued)

Pad Name	NUC946
GPIOC	**(11pins)**
GPIOC[2]	75
GPIOC[11:4]	83:76
GPIOC[14:13]	85,84

Pad Name	NUC946
SCAN	**(5 pins)**
SCAN[1:0]	68,67
SCAN[4:2]	73,72,71

Pad Name	NUC946
Miscellaneous	**(2 pins)**
nIRQ 0 / GPIOH 0	96
TEST	95
Power/Ground	**(22 pins)**
VDD18	7, 61,69,115
VDD33	29,44,92,108
VSS	6,13,43,62,70,74,97,109
USBVSS	118, 125
USBVDD (3.3V)	119,126
PLLVDD (1.8V)	66
PLLVSS	65

5 Pin Description

5.1 Pin Description for Interface

Pin Name	IO Type	Description
Clock & Reset (5)		
EXTAL15M	I	15MHz External Clock / Crystal Input for PLL
XTAL15M	O	15MHz Crystal Output
EXTAL48M0	I	48MHz External Clock / Crystal Input for USB2.0 PHY
XTAL48M0	O	48MHz Crystal Output
nRESET	I	System Reset (Low active)
External Bus Interface (73)		
MA [21:0]	O	Address Bus of external memory and IO devices. (MA[21:13] are set to input mode when nRESET low active)
MD [16:0]	IO (D)	Data Bus of external memory and IO device (Pull-down are programmable)
nWBE [1:0] / SDQM [1:0]	O	Write Byte Enable for specific device (nECS [3:0]). Data Bus Mask signal for SDRAM (nSCS [1:0]), (Low active)
nSCS [0]	O	SDRAM chip select for two external banks, (Low active)
nSRAS	O	Row Address Strobe for SDRAM, (Low active)
nSCAS	O	Column Address Strobe for SDRAM, (Low active)
nSWE	O	SDRAM Write Enable, (Low active)
MCKE	O	SDRAM Clock Enable
MCLK	O	System Master Clock Out, SDRAM clock
nBTCS	O	ROM/Flash Chip Select, (Low active)
nECS [0]	O	External I/O Chip Select, (Low active)
nOE	O	ROM/Flash, External Memory Output Enable, (Low active)
Ethernet RMII Interface (10)		
PHY_MDC	O(IS)	RMII Management Data Clock
PHY_MDIO	IO(D)	RMII Management Data I/O (Pull-down is programmable)
PHY_TXD [1:0]	O(IU)	RMII Transmit Data bus (Pull-up are programmable)
PHY_TXEN	O(ID)	RMII Transmit Enable (Pull-down is programmable)
PHY_REFCLK	O(ID)	RMII Reference Clock. (Pull-down is programmable)
PHY_RXD [1:0]	I(OU)	RMII Receive Data bus (Pull-up are programmable)
PHY_CRSDV	I(OD)	RMII Carrier Sense / Receive Data Valid (Pull-down is programmable)
PHY_RXERR	I(OD)	RMII Receive Data Error (Pull-down is programmable)

USB Interface (8)		
DP0	IO	Differential Positive USB Port0 IO signal
DN0	IO	Differential Negative USB Port0 IO signal
REXT0	A	External Resister Connect for Port0
DP1	IO	Differential Positive USB Port1 IO signal
DN1	IO	Differential Negative USB Port1 IO signal
REXT1	A	External Resister Connect for Port1
OVI	I	USB Over Current Detection signal
HDS	I	USB PHY 0 Device/Host Mode Select Control signal

I2C/USI(SPI/MW) Interface (4)		
SCL0 / SFRM	IOS	I2C Serial Clock Line 0. USI Serial Frame. (Input with Schmitt trigger)
SDA0 / SSPTXD	IOS	I2C Serial Data Line 0. USI Serial Transmit Data. (Input with Schmitt trigger)
SCL1 / SCLK	IOS	I2C Serial Clock Line 1. USI Serial Clock. (Input with Schmitt trigger)
SDA1 / SSPRXD	IOS	I2C Serial Data Line 1. USI Serial Receive Data. (Input with Schmitt trigger)

UART0/UART1/UART2 Interface (4)		
TXD0	IO(D)	UART0 Transmit Data. (Pull-down is programmable)
RXD0	IO(D)	UART0 Receive Data. (Pull-down is programmable)
TXD1	IO(D)	UART1 Transmit Data (Pull-down is programmable)
RXD1	IO(D)	UART1 Receive Data (Pull-down is programmable)

SD/SDIO/Memory Stick Interface (8)		
SD0_CMD / MS0_BS	IO(U)	SD/SDIO Mode – Command/Response (SPI Mode – Data In) Memory Stick Mode – Bus State. (Pull-up is programmable)
SD0_CLK / MS0_CLK	IO(U)	SD/SDIO Mode – Clock; (SPI Mode – Clock) Memory Stick Mode – Clock (Pull-up is programmable)
SD0_DAT0 / MS0_DAT0	IO(U)	SD/SDIO Mode – Data Line Bit 0; Memory Stick Mode – Data Line Bit 0; (Pull-up is programmable)
SD0_DAT1 / MS0_DAT1	IO(U)	SD/SDIO Mode – Data Line Bit 1; Memory Stick Mode – Data Line Bit 1; (Pull-up is programmable)
SD0_DAT2 / MS0_DAT2	IO(U)	SD/SDIO Mode – Data Line Bit 2; Memory Stick Mode – Data Line Bit 2; (Pull-up is programmable)
SD0_DAT3 / MS0_DAT3	IO(U)	SD/SDIO Mode – Data Line Bit 3; Memory Stick Mode – Data Line Bit 3; (Pull-up is programmable)
SD0_CDn / MS0_CDn	IO(U)	SD/SDIO Mode – Card Detect. Memory Stick Mode – Card Detect. (Pull-up is programmable)
SD_nPWR	IO(U)	SD/SDIO Power FET Control Signal Output. (Pull-up is programmable)

Miscellaneous(2)		
nIRQ[0]	I(OU)	External Interrupt Request (Pull-up is programmable)
TEST	I	Test Mode This pin has to pull low in normal operation.
Power/Ground		
VDD18	P	Core Logic power (1.8V)
VDD33	P	IO Buffer power (3.3V)
VSS	G	IO Buffer and Core ground (0V)
USBVDD33	P	USB Port1 PHY Transceiver power (3.3V)
USBVSS	G	USB Port1 PHY Transceiver ground (0V)
PLLVDD18	P	PLL power (1.8V)
PLLVSS18	G	PLL ground (0V)

5.2 GPIO Share Pin Description

In this chip, there are GPIOC~GPIOH groups for general IO control. All of GPIO pins are shared with the other interface and define as the following

GPIO Group	Shared pin function
GPIOC (11 pins)	GPIO Interface
GPIOC[2]	GPIO only
GPIOC[4]	GPIO only
GPIOC[5]	GPIO only
GPIOC[6]	GPIO only
GPIOC[7]	GPIO only
GPIOC[8]	GPIO only
GPIOC[9]	GPIO only
GPIOC[10]	GPIO only
GPIOC[11]	GPIO only
GPIOC[13]	GPIO only
GPIOC[14]	GPIO only

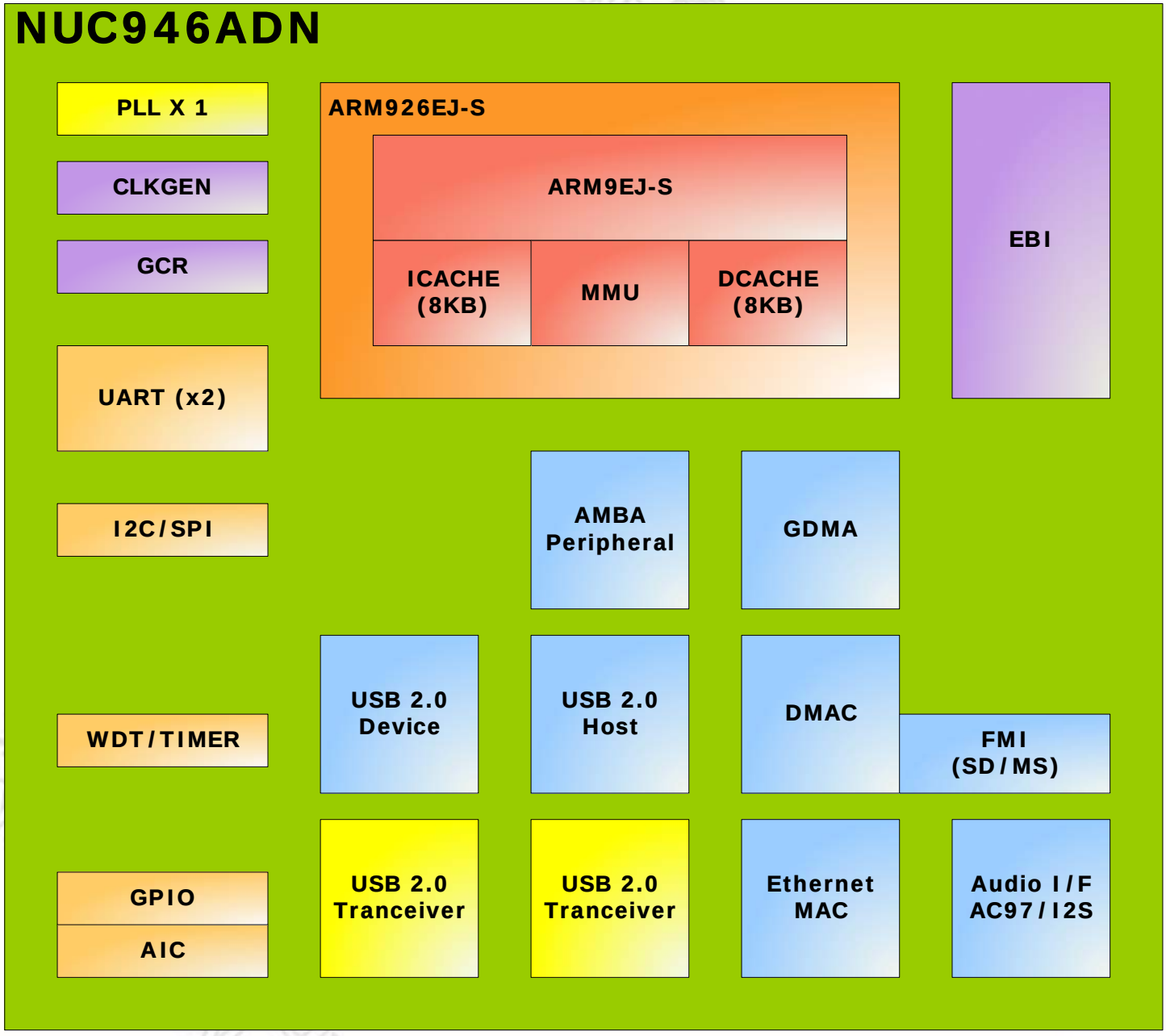
GPIOD (8 pins)	SD(SDIO) / Memory Stick Interface
GPIOD[0]	SD_CMD / MS_BS
GPIOD[1]	SD_CLK / MS_CLK
GPIOD[2]	SD_DAT0 / MS_DAT0
GPIOD[3]	SD_DAT1 / MS_DAT1
GPIOD[4]	SD_DAT2 / MS_DAT2
GPIOD[5]	SD_DAT3 / MS_DAT3
GPIOD[6]	SD_CDn / MS_CDn
GPIOD[8]	SD_nPWR / MS_nPWR

GPIOE (4 pins)	UART Interface
GPIOE[0]	TXD0
GPIOE[1]	RXD0
GPIOE[2]	TXD1
GPIOE[3]	RXD1
GPIOF (10 pins)	RMII Interface
GPIOF[0]	PHY_MDC
GPIOF [1]	PHY_MDIO
GPIOF [3:2]	PHY_TXD [1:0]
GPIOF [4]	PHY_TXEN
GPIOF [5]	PHY_REFCLK
GPIOF [7:6]	PHY_RXD [1:0]
GPIOF [8]	PHY_CRSDV
GPIOF [9]	PHY_RXERR

GPIOG (4 pins)	I2C/USI
GPIOG[0]	SCL0 / SFRM
GPIOG[1]	SDA0 / SSPTXD
GPIOG[2]	SCL1 / SCLK
GPIOG[3]	SDA1 / SSPRXD

GPIOH (1 pins)	nIRQ Interface
GPIOH[0]	nIRQ[0]

6 Functional Block



7 Functional Description

7.1 ARM926EJ-S CPU CORE

The ARM926EJ-S CPU core is a member of the ARM9 family of general-purpose microprocessors. The ARM926EJ-S CPU core supports the 32-bit ARM and 16-bit Thumb instruction sets, enabling the user to trade off between high performance and high code density. The ARM926EJ-S processor has a Harvard cached architecture with MMU.

7.2 System Manager

7.2.1 Overview

The System Manager has the following functions.

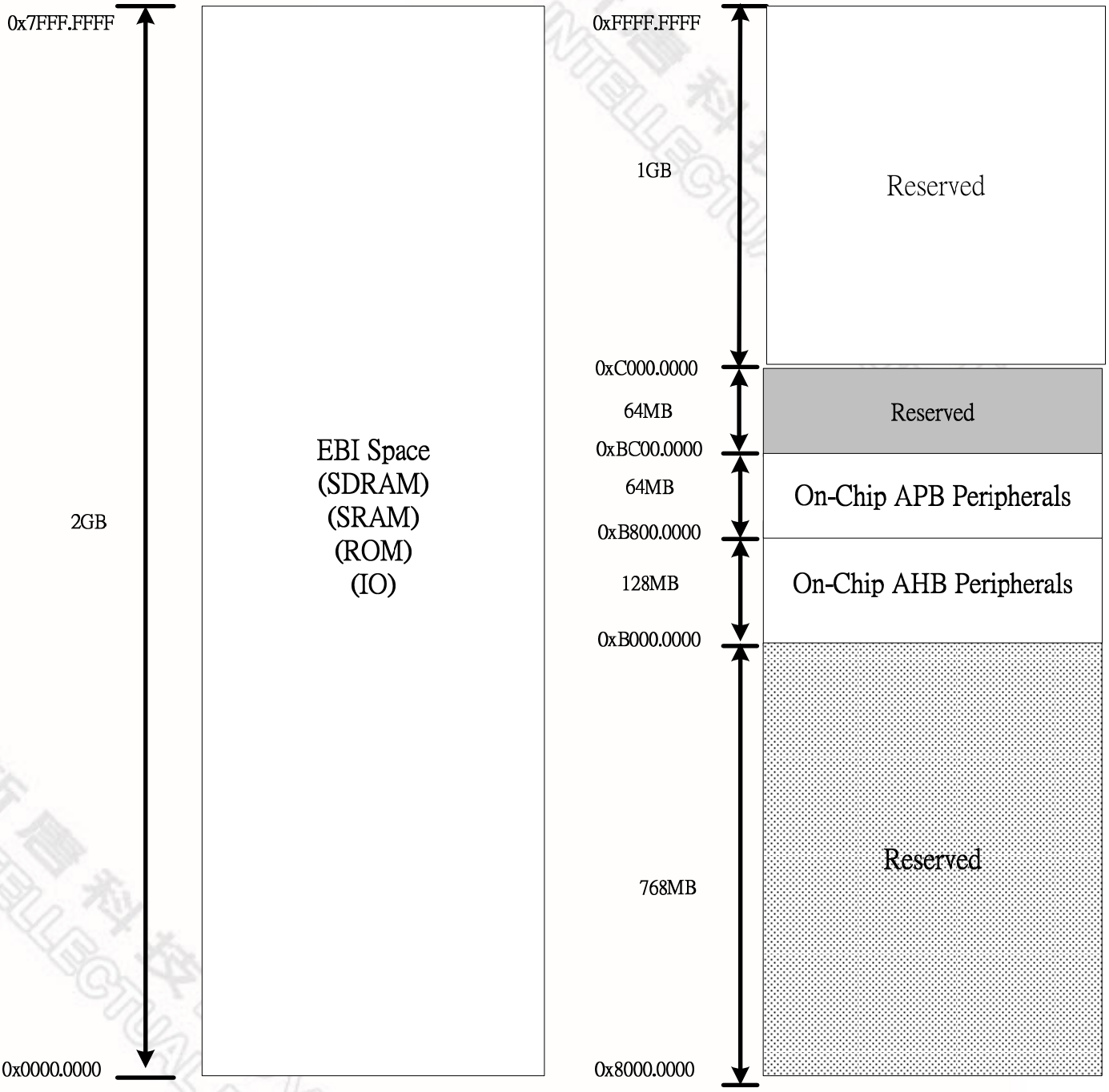
- System memory map
- The width of external memory address
- Data bus connection with external memory
- Product identifier register
- Bus arbitration
- PLL module
- Clock select register
- Power-On setting

7.2.2 System Memory Map

This chip provides 2G bytes memory space (0x0000_0000~0x7FFF_FFFF) for the SDRAM, RAM, ROM and IO Devices, 192M bytes space (0xB000_0000~0xBBFF_FFFF) for On-Chip Peripherals and the other memory spaces are reserved.

The size and location of each SDRAM memory bank is determined by the register settings for "current bank base address pointer" and "current bank size" (SDCONF0 and SDCONF1). Please note that when setting the bank control registers, the address boundaries of consecutive banks must not be overlapped.

Except On-Chip Peripherals, the start address of each memory bank is not fixed. You can use bank control registers to assign a specific bank start address by setting the bank's base pointer (13 bits). The address resolution is 256K bytes. The bank's start address is defined as "base pointer << 18" and the bank's size is "current bank size". (EXT0CON)
The CPU booting start address (from external ROM) is fixed at address 0x0000_0000 after reset or power-on. In the event of an access request to an address outside any programmed bank size, an abort signal is generated. The maximum accessible memory size of each external IO bank is 8M bytes, and 128M bytes on SDRAM banks.


0x7FFF.FFFF
2GB
0x0000.0000
EBI Space
(SDRAM)
(SRAM)
(ROM)
(IO)
0xFFFF.FFFF
1GB
Reserved
0xC000.0000
64MB
Reserved
0xBC00.0000
64MB
On-Chip APB Peripherals
0xB800.0000
128MB
On-Chip AHB Peripherals
0xB000.0000
768MB
Reserved
0x8000.0000

Address Space	Token	Modules
0x0000_0000 – 0x7FFF_FFFF	------	EBI(SDRAM, ROM, RAM, IO) Memory Space
0x8000_0000 – 0xAFFF_FFFF	------	Reserved Shadow of EBI Memory Space(0x0000_0000~0x2FFF_FFFF)
0xB000_0000 – 0xB000_01FF	GCR_BA	System Global Control Registers
0xB000_0200 – 0xB000_02FF	CLK_BA	Clock Control Registers
0xB000_1000 – 0xB000_1FFF	EBI_BA	EBI Control Registers
0xB000_3000 – 0xB000_3FFF	EMC_BA	Ethernet MAC Control Registers
0xB000_4000 – 0xB000_4FFF	------	Reserved
0xB000_5000 – 0xB000_5FFF	USBH_BA	EHCI USB Host Control Registers
0xB000_6000 – 0xB000_6FFF	USBD_BA	USB Device Control Registers
0xB000_7000 – 0xB000_7FFF	USBO_BA	OHCI USB Host Control Registers
0xB000_8000 – 0xB000_8FFF	------	Reserved
0xB000_9000 – 0xB000_9FFF	------	Reserved
0xB000_A000 – 0xB000_AFFF	------	Reserved
0xB000_B000 – 0xB000_BFFF	------	Reserved
0xB000_C000 – 0xB000_CFFF	DMAC_BA	DMA Controller Registers
0xB000_D000 – 0xB000_DFFF	FMI_BA	Flash Memory Interface Control Registers

Address Space	Token	Modules
0xB800_0000 – 0xB800_00FF	UART0_BA	UART 0 Control Registers (Tx,Rx for console)
0xB800_0100 – 0xB800_01FF	UART1_BA	UART 1 Control Registers (Tx,Rx)
0xB800_0200 – 0xB800_02FF	------	Reserved
0xB800_0300 – 0xB800_03FF	------	Reserved
0xB800_0400 – 0xB800_04FF	------	Reserved
0xB800_1000 – 0xB800_1FFF	TMR_BA	Timer Control Registers
0xB800_2000 – 0xB800_2FFF	AIC_BA	Interrupt Controller Registers
0xB800_3000 – 0xB800_3FFF	GPIO_BA	GPIO Control Registers
0xB800_4000 – 0xB800_4FFF	------	Reserved
0xB800_5000 – 0xB800_5FFF	------	Reserved
0xB800_6000 – 0xB800_60FF	I2C0_BA	I2C 0 Control Register
0xB800_6100 – 0xB800_61FF	I2C1_BA	I2C 1 Control Register
0xB800_6200 – 0xB800_62FF	USI_BA	Universal Serial Interface Register (USI)
0xB800_7000 – 0xB800_7FFF		Reserved
0xB800_8000 – 0xB800_8FFF	------	Reserved
0xB800_9000 – 0xB800_9FFF	------	Reserved
0xB800_A000 – 0xB800_AFFF	------	Reserved

7.2.3 Address Bus Generation

The address bus generation is depended on the required data bus width **(DBWD)** and address bus alignment control bit **(ADRS)** of each IO bank. The maximum accessible memory size of each external IO bank is 32M bytes. (EXT0CON)

Table 7.2.1 Address Bus Generation Guidelines (When ADRS bit = 0)

Data Bus Width	External Address Pins			Maximum Accessible Memory Size
	MA [21:0]			
8-bit	MA21 – MA0 (Internal)			32M bytes
16-bit	MA22 – MA1 (Internal)			32M bytes (16M half-words)

Table 7.2.2 Address Bus Generation Guidelines (When ADRS bit = 1)

Data Bus Width	External Address Pins			Maximum Accessible Memory Size
	MA [21:0]			
8-bit	MA21 – MA0 (Internal)			32M bytes
16-bit	MA21 – MA0 (Internal)			32M bytes, MA[0] ignored (16M half-words)

7.2.4 AHB Bus Arbitration

The system bus is AHB-compliant and supports modules with standard AHB master or slave interfaces.
The AHB arbiter has two priority-decision modes, i.e., the fixed priority mode and the rotate priority mode. In the rotate priority mode, there are three types for AHB-Master bus. The selection of modes and types is determined on the **PRTMOD0** and **PRTMOD1**bits in the Arbitration Control Register. **PRTMOD0** is used to control the fixed priority of AHB1 (CPU AHB-Lite) Bus and **PRTMOD1** is used to control the fixed priority of AHB2 Master Bus.

7.2.4.1 Fixed Priority Mode

Fixed priority mode is selected if **PRTMODx** = 0. The order of priorities on the AHB mastership among the on-chip master modules, listed in Table 7.2.3, is fixed. If two or more master modules request to AHB at the same time, the mastership is always granted to the module with the highest priority.

Table 7.2.3 AHB Bus Priority Order in Fixed Priority Mode

Priority Sequence	PRTMOD0 = 0 AHB1 Bus	PRTMOD1 = 0 AHB2 Bus
1 (Lowest)	ARM CPU Instruction	AHB Bridge
2	ARM CPU Data	---
3	---	---
4	---	SDIO(FMI)
5	---	USB Device
6	---	USB Host
7	---	EMC Controller
8	---	---
9 (Highest)	---	

The ARM core normally has the lowest priority under the fixed priority mode; however, this chip provides a mechanism to raise the priority to the highest. If the IPEN bit (bit-1 of Arbitration Control Register) is set to 1, the **IPACT** bit (bit-2 of Arbitration Control Register) will be automatically set to 1 while an unmasked external interrupt occurs. Under this circumstance, the ARM core gains the highest AHB priority.
The programmer can recover the original priority order by directly writing "0" to clear the **IPACT** bit. For example, this can be done that at the end of an interrupt service routine. Note that **IPACT** only can be automatically set to 1 by an external interrupt when **IPEN** = 1. It will not take effect if a programmer to directly write 1 to **IPACT** to raise ARM core's AHB priority.

7.2.4.2 Rotate Priority Mode

Rotate priority mode is selected if PRTMODx = 1. The AHB arbiter uses a round robin arbitration scheme by which every master module can gain the bus ownership in turn.
For AHB2 DMA Master Bus, the Audio and LCD Display, have the higher priority in the rotate type.

7.2.5 Power-On Setting

After power on reset, Power-On setting registers are latched from EBI Address pins (MA [21:13]) to configure this chip.

Power-On Setting	Pin
Booting Device Select	MA [21:20]
Internal System Clock Select	MA17
GPIO Pin Configuration Select	MA [16:14]
USB PHY0 Mode Select	HDS

MA [21:20] : Booting Device Select

MA[21:20]		Booting Device
Pull-down	Pull-down	SPI Flash ROM
Pull-down	Pull-up	Reserved
Pull-up	Pull-down	USB ISP
Pull-up	Pull-up	NOR-type Flash ROM

MA19 : Pull-up is necessary
MA18 : Can either Pull-up or Pull-down
MA17 : Internal System Clock Select
If pin MA17 is pull-down, the external clock from EXTAL15M pin is served as internal system clock.
If pin MA17 is pull-up, the PLL output clock is used as internal system clock.

MA [16:14] : GPIO Pin Configuration Select

MA[16:14]	State	GPIO Pin Function
MA14	Pull-down	GPIOC/D/E Group Select
	Pull-up	UART Group Select
MA15	Pull-down	GPIOF Group Select
	Pull-up	RMII Group Select
MA16	Pull-down	GPIOI Group Select
	Pull-up	Reserved

MA13 : Pull-up is necessary

HDS: USB PHY0 Mode Select

HDS	USB PHY0 Mode
Pull-down	USB20 Host
Pull-up	USB20 Device

7.2.6 System Booting

NUC946ADN supports four kinds of system booting devices, which including
(1) SPI Flash ROM
(2) USB ISP
(3) NOR-type Flash ROM

Booting Device Select

MA[21:20]		Booting Device
Pull-down	Pull-down	SPI Flash ROM
Pull-down	Pull-up	Reserved
Pull-up	Pull-down	USB ISP
Pull-up	Pull-up	NOR-type Flash ROM

7.2.7 System Global Control Registers Map

Register	Address	R/W	Description	Reset Value
GCR_BA = 0xB000_0000				
PDID	0xB000_0000	R	Product Identifier Register	0xxx90_0910
PWRON	0xB000_0004	R/W	Power-On Setting Register	N/A
ARBCON	0xB000_0008	R/W	Arbitration Control Register	0x0000_0000
MFSEL	0xB000_000C	R/W	Multiple Function Pin Select Register	0x0800_0000
EBIDPE	0xB000_0010	R/W	EBI Data Pin Pull-up/down Enable Register	0xFFFF_FFFF
GPIOCPE	0xB000_0018	R/W	GPIOC Pin Pull-up/down Enable Register	0x0000_7FFF
GPIODPE	0xB000_001C	R/W	GPIOD Pin Pull-up/down Enable Register	0x0000_07FF
GPIOEPE	0xB000_0020	R/W	GPIOE Pin Pull-up/down Enable Register	0x0000_3FFF
GPIOFPE	0xB000_0024	R/W	GPIOF Pin Pull-up/down Enable Register	0x0000_03FF
GPIOGPE	0xB000_0028	R/W	GPIOG Pin Pull-up/down Enable Register	0x0001_FFFF
GPIOHPE	0xB000_002C	R/W	GPIOH Pin Pull-up/down Enable Register	0x0000_00FF
GTMP1	0xB000_0034	R/W	General Temporary Register 1	N/A
GTMP2	0xB000_0038	R/W	General Temporary Register 2	N/A
GTMP3	0xB000_003C	R/W	General Temporary Register 3	N/A

Product Identifier Register (PDID)

This register is for only read and enables software to recognize certain characteristics of the chip ID and the version number.

Register	Address	R/W	Description	Reset Value
PDID	0xB000_0000	R	Product Identifier Register	0xxx90_0910

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

Bits	Descriptions	
[31:24]	**VERSION**	**Version of chip** 02: Version C
[23:0]	**CHIPID**	**Chip identifier** The NUC946ADN Chip identifier is 0x90_0910.

Power-On Setting Register（PWRON）

This register latches the chip power-on setting from EBI Address Bus during chip reset.

Register	Address	R/W	Description	Reset Value
PWRON	0xB000_0004	R/W	Power-On Setting Register	Undefined

31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED					USBDEN	USBHD	RESERVED
7	6	5	4	3	2	1	0
Booting Device Select		RESERVED		GPIOSEL		Reserved	PLL

Bits	Descriptions	
[0]	**PLL**	**Internal System Clock Select (Read/Write)** Power-On value latched from MA17 0= the external clock from EXTAL15M pin is served as internal system clock. 1= the PLL output clock is used as internal system clock.
[3:2]	**GPIOSEL**	**GPIO Pin Configuration Select(Read Only)** [1] / MA14: 0 = GPIOC/D/E; 1 = UART [2] / MA15: 0 = GPIOF; 1 = RMII
[7:6]	**Booting Device Select**	**Booting Device Select (Read Only)** these two bits are power-on reset from MA[21:20] [7:6] = 00: SPI Flash ROM [7:6] = 01: Reserved [7:6] = 10: USB ISP [7:6] = 11: NOR-type Flash ROM

[9]	**USBHD**	**USB PHY0 Mode Select (Read/Write)** this bit is power-on reset from HDS **USBHD** / **USB PHY0 Mode** / **HDS Pin** 0 / USB20 Device / External Pull-Up 1 / USB20 Host / External Pull-Down
[10]	**USBDEN**	**USB PHY0 Enable Control for USB Device Mode (Read/Write)** This bit is only active when the USBHD bit be zero (Device Mode) **USBDEN** / **USB PHY0 Enable** 0 / Set Device PHY at SE0 (Not active to external host) 1 / Set Device PHY controlled by the UTMI interface of the USB Device Controller

Arbitration Control Register (ARBCON)

Register	Address	R/W	Description	Reset Value
ARBCON	0xB000_0008	R/W	Arbitration Control Register	0x0000_0000

31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
RESERVED			DGMASK	IPACT	IPEN	PRTMOD1	PRTMOD0

Bits	Descriptions	
[4]	**DGMASK**	**Default Grant Master Mask Control** 0 = AHB-Bridge always be the default grant master (default) 1 = No default grant master on AHB-2 Bus
[3]	**IPACT**	**Interrupt Priority Active** When **IPEN**="1", this bit is set when the ARM core has an unmasked interrupt request. This bit is available only when the **PRTMOD1**=0 and **PRTMOD0**=0.
[2]	**IPEN**	**Interrupt Priority Enable Bit** 0 = the ARM core has the lowest priority. 1 = enable to raise the ARM core priority to second This bit is available only when the **PRTMOD**=0 and **PRTMOD0**=0.
[1]	**PRTMOD1**	**Priority Mode Select for AHB2 (AHB Master Bus)** 0 = Fixed Priority Mode (default) 1 = Rotate Priority Mode
[0]	**PRTMOD0**	**Priority Mode Select for AHB1 (CPU AHB-Lite Bus)** 0 = Fixed Priority Mode (default) 1 = Rotate Priority Mode

Multiple Function Pin Select Register (MFSEL)

Register	Address	R/W	Description	Reset Value
MFSEL	0xB000_000C	R/W	Multiple Function Pin Select Register	0x0800_0000

31	30	29	28	27	26	25	24
RESERVED		USBPHY0		Reserved		GPSELH	
23	22	21	20	19	18	17	16
Reserved		Reserved				GPSELG	
15	14	13	12	11	10	9	8
GPSELG		Reserved		Reserved	Reserved	GPSELE	
7	6	5	4	3	2	1	0
GPSELD				GPSELC		GPSELF	Reserved

Bits	Descriptions	
[29:28]	**USBPHY0**	**USB PHY0 Select Control Register** **00 : Normal USB operation mode (Default)**
[25:24]	**GPSELH**	**GPIOH Pin Function Select Control Register** PIN / GPSELH[24] / GPIO Pin Function GPIOH[0] / 0 / GPIOH[0] GPIOH[0] / 1 / nIRQ[0] **GPSELG [25:24] default value is 0 for GPIOH group.**
[23:22] [17:14]	**GPSELG**	**GPIOG Pin Function Select Control Register** PIN / GPSELG[17:16] / GPIO Pin Function GPIOG[3:2] / 00 / GPIOG[3:2] GPIOG[3:2] / 01 / I2C Line1 GPIOG[3:2] / 10 / USI Interface GPIOG[3:2] / 11 / Reserved PIN / GPSELG[15:14] / GPIO Pin Function GPIOG[1:0] / 00 / GPIOG[1:0] GPIOG[1:0] / 01 / I2C Line0 GPIOG[1:0] / 10 / USI Interface GPIOG[1:0] / 11 / Reserved **See GPIO Shared Pin Description for more detail** **GPSELG [23:22], [17:14] default value is 0, GPIOG group.**

<table>
<tr>
<td>[9:8]</td>
<td>GPSELE</td>
<td>
GPIOE Pin Function Select Control Register
See GPIO Shared Pin Description for more detail
GPSELE [9:8] default value is 0 for GPIOE group.
<table>
<tr><th></th><th></th><th>GPIO Pin Function</th></tr>
<tr><td>PIN</td><td>GPSELE[9]</td><td></td></tr>
<tr><td rowspan="2">GPIOE[3:2]</td><td>0</td><td>GPIOE[3:2]</td></tr>
<tr><td>1</td><td>UART1</td></tr>
<tr><td>PIN</td><td>GPSELE[8]</td><td></td></tr>
<tr><td rowspan="2">GPIOE[1:0]</td><td>0</td><td>GPIOE[1:0]</td></tr>
<tr><td>1</td><td>UART0</td></tr>
</table>
</td>
</tr>
<tr>
<td>[7:4]</td>
<td>GPSELD</td>
<td>
GPIOD Pin Function Select Control Register
<table>
<tr><th>PIN</th><th>GPSELD[7:6]</th><th>GPIO Pin Function</th></tr>
<tr><td rowspan="4">GPIOD[8], GPIOD[6:5]</td><td>00</td><td>GPIOD[8], GPIOD[6:5]</td></tr>
<tr><td>01</td><td>Reserved</td></tr>
<tr><td>10</td><td>SD 0 Interface</td></tr>
<tr><td>11</td><td>Memory Stick 0</td></tr>
<tr><td>PIN</td><td>GPSELD[5:4]</td><td>GPIO Pin Function</td></tr>
<tr><td rowspan="4">GPIOD[4:0]</td><td>00</td><td>GPIOD[4:0]</td></tr>
<tr><td>01</td><td>Reserved</td></tr>
<tr><td>10</td><td>SD 0 Interface</td></tr>
<tr><td>11</td><td>Memory Stick 0</td></tr>
</table>
See GPIO Shared Pin Description for more detail
GPSELD[7:4] default value is depend on power-on setting
</td>
</tr>
<tr>
<td>[3:2]</td>
<td>GPSELC</td>
<td>
GPIOC Pin Function Select Control Register
<table>
<tr><th>PIN</th><th>GPSELC[3:2]</th><th>GPIO Pin Function</th></tr>
<tr><td rowspan="4">GPIOC[14:13], GPIOC[11:4], GPIOC[2]</td><td>00</td><td>GPIOC[14:13], GPIOC[11:4], GPIOC[2]</td></tr>
<tr><td>01</td><td>Reserved</td></tr>
<tr><td>10</td><td>Reserved</td></tr>
<tr><td>11</td><td>Reserved</td></tr>
</table>
See GPIO Shared Pin Description for more detail
GPSELC[3:2] default value is depend on power-on setting
</td>
</tr>
<tr>
<td>[1]</td>
<td>GPSELF</td>
<td>
GPIOF Pin Function Select Control Register
<table>
<tr><th>PIN</th><th>GPSELF[1]</th><th>GPIO Pin Function</th></tr>
<tr><td rowspan="2">GPIOF[9:0]</td><td>0</td><td>GPIOF[9:0]</td></tr>
<tr><td>1</td><td>RMII Interface</td></tr>
</table>
See GPIO Shared Pin Description for more detail
GPSELF[1] default value is depend on power-on setting
</td>
</tr>
</table>

EBI Data Pin Pull-up/down Enable Register (EBIDPE)

GPIOC~GPIOI Pin Pull-up/down Enable Register (GPIOCPE~GPIOIPE)

These registers are used to control the IO pins to be internal pull-up or down, which can avoid the input pins floating if there is no external resistors.

Register	Address	R/W	Description	Reset Value
EBIDPE	0xB000_0010	R/W	EBI Data Pin Pull-down Enable Register	0xFFFF_FFFF
---------	0xB000_0014	R/W	Reserved	N/A
GPIOCPE	0xB000_0018	R/W	GPIOC Pin Pull-up Enable Register	0x0000_FFFF
GPIODPE	0xB000_001C	R/W	GPIOD Pin Pull-up Enable Register	0x0000_07FF
GPIOEPE	0xB000_0020	R/W	GPIOE Pin Pull-up/down Enable Register	0x0000_3FFF
GPIOFPE	0xB000_0024	R/W	GPIOF Pin Pull-up/down Enable Register	0x0000_03FF
GPIOGPE	0xB000_0028	R/W	GPIOG Pin Pull-up/down Enable Register	0x0001_FFFF
GPIOHPE	0xB000_002C	R/W	GPIOH Pin Pull-up Enable Register	0x0000_00FF

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

Bits	Descriptions	
[31:0]	**PPE**	**Pin Pull-down Enable Register** 1 = Disable the Pull-high/down for each relative pin (default) 0 = Enable the Pull-high/down for each relative pin

Register	Descriptions
EBIDPE	**EBI Data Pin Pull-down Enable Register** PPE[15:0] controls the Pull-down of the EBI Data Bus[15:0] PPE[31:16] are reserved in this register.
GPIOCPE	**GPIOC Pin Pull-up Enable Register** PPE[31:15], PPE[12], PPE[3] and PPE[1:0] are reserved in this register PPE[14:13], PPE[11:4] and PPE[2] control the Pull-up of the related GPIOC signals.
GPIODPE	**GPIOD Pin Pull-up Enable Register** PPE[31:9] and PPE[7] are reserved in this register PPE[8] and PPE[6:0] control the Pull-up of the related GPIOD signals.
GPIOEPE	**GPIOE Pin Pull-up/down Enable Register** PPE[31:4] are reserved in this register PPE[3:0] controls the Pull-up/down of the GPIOE[3:0] Pull-down : GPIOE[3:0]
GPIOFPE	**GPIOF Pin Pull-up/down Enable Register** PPE[31:10] is reserved in this register PPE[9:0] controls the Pull-up/down of the GPIOF[9:0] Pull-down : GPIOF[9:8], GPIOF[5:4], GPIOF[1] Pull-up : GPIOF[7:6], GPIOF[3:2] No action : GPIOF[0]
GPIOGPE	**GPIOG Pin Pull-up/down Enable Register** PPE[31:4] are reserved in this register PPE[3:0] controls the Pull-up of the GPIOG[3:0] Pull-down : GPIOG[3:0]
GPIOHPE	**GPIOH Pin Pull-up Enable Register** PPE[31:1] is reserved in this register PPE[0] controls the Pull-up of the GPIOH[0]

1 = Disable the Pull-high/down for each relative pin
0 = Enable the Pull-high/down for each relative pin

General Temporary Register 1 ~ 3 (GTMP1 ~GTMP3)

Register	Address	R/W	Description	Reset Value
GTMP1	0xB000_0034	R/W	General Temporary Register 1	Undefined
GTMP2	0xB000_0038	R/W	General Temporary Register 2	Undefined
GTMP3	0xB000_003C	R/W	General Temporary Register 3	Undefined

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

Bits	Descriptions	
[31:0]	DATA	General Temporary Data

7.3 Clock Controller

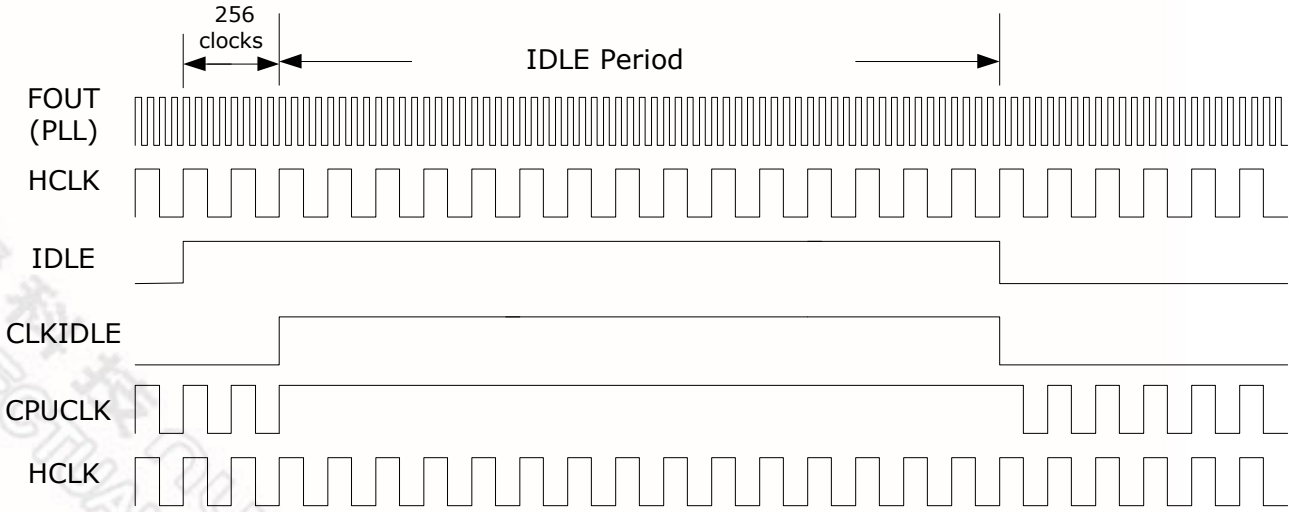
The clock controller generates all clocks for Display, Audio, CPU, AMBA and all the engine modules. In this chip includes two PLL modules. The clock source for each module is come from the PLL, or from the external crystal input directly. The CLKEN register controls the IP clock ON or OFF individually, and the CLKDIV register controls the divider setting. The register can also be used to control the clock enable or disable for power management control.

7.3.1 Power management

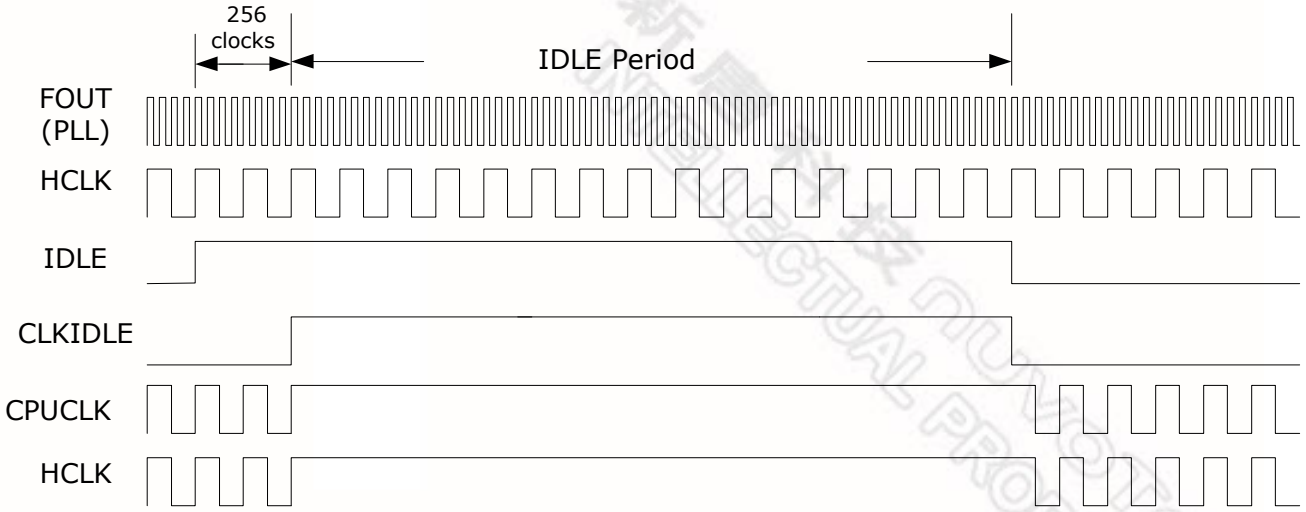
This chip provides three power management scenarios to reduce power consumption. The peripheral clocks can be enabled / disabled individually by controlling the corresponding bit in CLKSEL control register. Software can turn-off the unused modules' clock for power saving. It also provides **IDLE** and **Power-down** modes to reduce the power consumption.

IDLE MODE

If the **IDLE** bit in Power Management Control Register (PMCON) is set, the ARM CORE clock source will be halted after 256 cycles, and then the ARM core will stop. The AHB or APB clocks are still active except the clock to cache controller and ARM core. This ARM core will exit from this mode when a **nIRQ** or **nFIQ** signals from any peripheral, such as Keypad and Timer overflow interrupts. The memory controller can also be forced to enter idle state if both the **MIDLE** and **IDLE** bits are set.



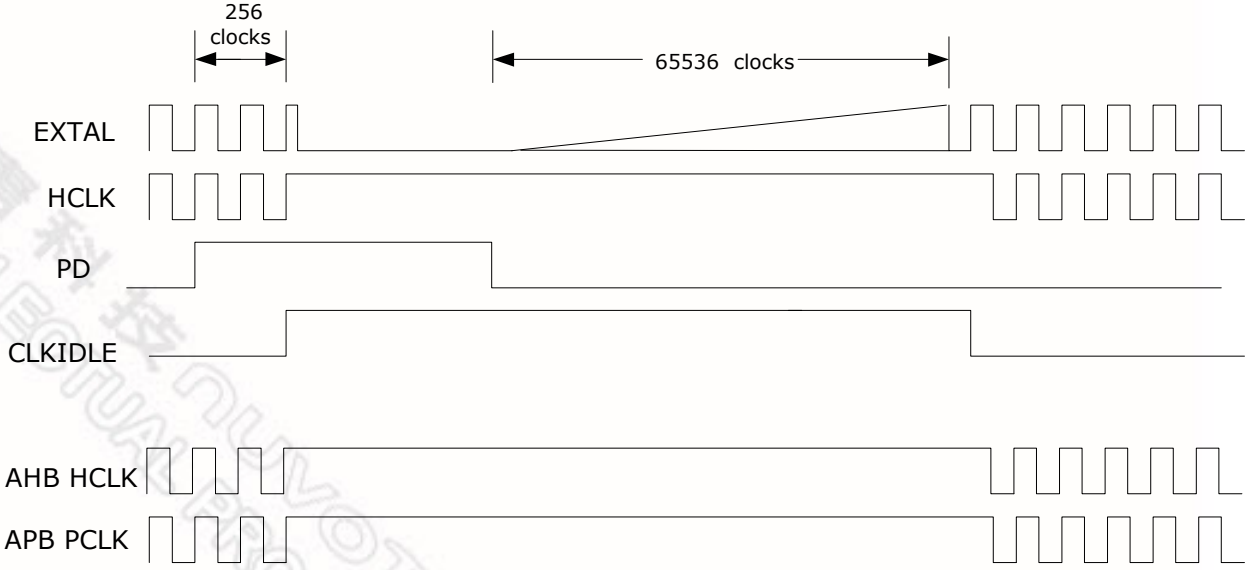
Case1. IDLE=1, PD=0, MIDLE=0



Case2. IDLE=1, PD=0, MIDLE=1

Power-Down Mode

The mode provides the minimum power consumption. When the system is not working or waiting an external event, software can write PD bit to turn off all the clocks includes system crystal oscillator and PLL to let ARM core to enter sleep mode after 256 clock cycles. In this state, all peripherals are also in sleep mode since the clock source is stopped. This system will exit from this mode when external interrupts (**nIRQ** signals) are detected; this chip provides external interrupts, USB device, RTC and Keypad to wakeup the clock.



Case3. IDLE=0, PD=1, MIDLE=0

7.3.2 Clock Control Registers Map

Register	Address	R/W	Description	Reset Value
CLK_BA = 0xB000_0200				
CLKEN	0xB000_0200	R/W	Clock Enable Register	0x0408_0834
CLKSEL	0xB000_0204	R/W	Clock Select Register	0x0000_0FFX
CLKDIV	0xB000_0208	R/W	Clock Divider Control Register	0x0400_0000
PLLCON0	0xB000_020C	R/W	PLL Control Register 0	0x0000_2B63
PMCON	0xB000_0214	R/W	Power Management Control Register	0x0000_0000
IRQWAKECON	0xB000_0218	R/W	IRQ Wakeup Control Register	0x0000_0000
IRQWAKEFLAG	0xB000_021C	R/W	IRQ Wakeup Flag Register	0x0000_0000
IPSRST	0xB000_0220	R/W	IP Software Reset Register	0x0000_0000
CLKEN1	0xB000_0224	R/W	Clock Enable Register	0x0000_0000
CLKDIV1	0xB000_0228	R/W	Clock Divider Control 1 Register	0x0000_0000

Clock Enable Register (CLKEN)

Register	Address	R/W	Description	Reset Value
CLKEN	0xB000_0200	R/W	Clock Enable Register	0x0408_0834

31	30	29	28	27	26	25	24
I2C1	I2C0	USI	Reserved		WDT	Reserved	
23	22	21	20	19	18	17	16
TIMER4	TIMER3	TIMER2	TIMER1	TIMER0	Reserved	Reserved	
15	14	13	12	11	10	9	8
Reserved			UART1	UART0	Reserved	USBH	USBD
7	6	5	4	3	2	1	0
EMC	Reserved	DMAC	FMI	Reserved			

Bits	Descriptions	
[31]	**I2C1**	**I2C Interface 1 Clock Enable Bit** 0 = Disable I2C-1 clock 1 = Enable I2C-1 clock
[30]	**I2C0**	**I2C Interface 0 Clock Enable Bit** 0 = Disable I2C-0 clock 1 = Enable I2C-0 clock
[29]	**USI**	**USI Clock Enable Bit** 0 = Disable USI clock 1 = Enable USI clock
[26]	**WDT**	**WDT Clock Enable Bit** 0 = Disable WDT counting clock 1 = Enable WDT counting clock
[23]	**TIMER4**	**Timer4 Clock Enable Bit** 0 = Disable Timer clock 1 = Enable Timer clock
[22]	**TIMER3**	**Timer3 Clock Enable Bit** 0 = Disable Timer clock 1 = Enable Timer clock
[21]	**TIMER2**	**Timer2 Clock Enable Bit** 0 = Disable Timer clock 1 = Enable Timer clock
[20]	**TIMER1**	**Timer1 Clock Enable Bit** 0 = Disable Timer clock 1 = Enable Timer clock

[19]	**TIMER0**	**Timer0 Clock Enable Bit** 0 = Disable Timer clock 1 = Enable Timer clock
[12]	**UART1**	**UART1 Clock Enable Bit** 0 = Disable UART1 clock 1 = Enable UART1 clock
[11]	**UART0**	**UART0 Clock Enable Bit** 0 = Disable UART0 clock 1 = Enable UART0 clock
[9]	**USBH**	**USB Clock Enable Bit** 0 = Disable USB clock 1 = Enable USB clock
[8]	**USBD**	**USB device Clock Enable Bit** 0 = Disable USB host clock 1 = Enable USB host clock
[7]	**EMC**	**EMC Clock Enable Bit** 0 = Disable EMC clock 1 = Enable EMC clock
[5]	**DMAC**	**DMAC Clock Enable Bit** 0 = Disable DMAC clock 1 = Enable DMAC clock
[4]	**FMI**	**FMI Clock Enable Bit** 0 = Disable FMI clock 1 = Enable FMI clock

Clock Select Register (CLKSEL)

Register	Address	R/W	Description	Reset Value
CLKSEL	0xB000_0204	R/W	Clock Select Register	0x0000_0FFX

31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							MSDSEL
15	14	13	12	11	10	9	8
MSDSEL				Reserved		UART1SEL	
7	6	5	4	3	2	1	0
Reserved		Reserved		RESERVED		CPUCKSEL	

Bits	Descriptions	
[16:12]	**MSSEL**	**MS Engine Clock Source Select Bit** [16:15] MSSEL[16:15] / Clock Source 0 0 PLL0 Clock 0 1 PLL1 Clock 1 0 EXTAL15M pin 1 1 EXTAL15M pin (Default) [14:12] Selected PLL0 or PLL1 source divided from 1 to 8.
[9:8]	**UART1SEL**	**UART1 Clock Source Select Bit** UART1SEL / Clock Source 0 0 PLL0 Clock 0 1 PLL1 Clock 1 0 EXTAL15M pin 1 1 EXTAL15M pin (Default)
[1:0]	**CPUCKSEL**	**CPU/AMBA Clock Source Select Bit** Default value is depended on power-on setting (Pin A17) CPUCKSEL / Clock Source 0 0 PLL0 Clock 0 1 PLL1 Clock 1 0 PLL0 /2 Clock 1 1 EXTAL15M pin

Clock Divider Control Register (CLKDIV)

Register	Address	R/W	Description	Reset Value
CLKDIV	0xB000_0208	R/W	Clock Divider Control Register	0x0400_0000

31	30	29	28	27	26	25	24
RESERVED				APBCKDIV		AHBCKDIV	
23	22	21	20	19	18	17	16
RESERVED				UART1DIV			
15	14	13	12	11	10	9	8
RESERVED				RESERVED			
7	6	5	4	3	2	1	0
RESERVED				CPUCKDIV			

Bits	Descriptions	
[27:26]	**APBCKDIV**	**AMBA APB Clock Divider Control Register** **APBCKDIV** / **Clock Frequency** 0 0 : Reserved 0 1 : AHBCLK/2 1 0 : AHBCLK/4 1 1 : AHBCLK/8
[25:24]	**AHBCKDIV**	**AMBA AHB Clock (AHBCLK) Divider Control Register** **AHBCKDIV** / **Clock Frequency** 0 0 : CPUCLK/1 0 1 : CPUCLK/2 1 0 : CPUCLK/4 1 1 : CPUCLK/8
[19:16]	**UART1DIV**	**UART1 Clock Source Divider Control Register** **UART1CK = UART1 clock / (UART1DIV +1)** Where (1) UART1DIV is 0~15 (2) UART1 clock is the clock source output by UART1SEL control reg.

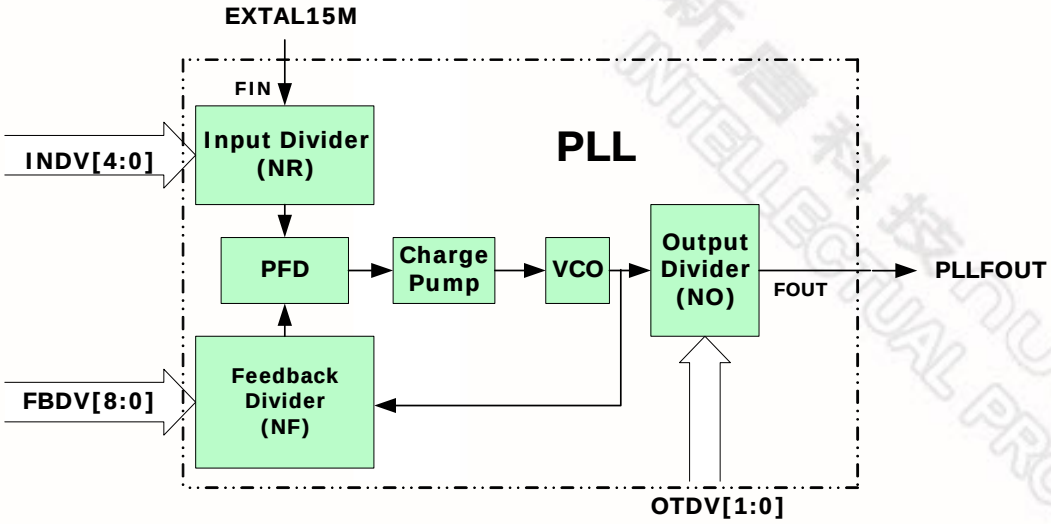
[3:0]	**CPUCKDIV**	**CPU Clock Source Divider Control Register** **CPUCLK = CCK clock / (CPUCKDIV +1)** Where (1) CPUCKDIV is 0~15 (2) CCK clock is the clock source output by CPUCKSEL control register

PLL Control Register 0（PLLCON0）

Register	Address	R/W	Description	Reset Value
PLLCON0	0xB000_020C	R/W	PLL Control Register 0	0x0000_2B63

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

Bits	Descriptions	
[16]	**PWDEN**	**Power Down Mode Enable** 0 = PLL is in normal mode 1 = PLL is in power down mode
[15:7]	**FBDV**	**PLL VCO Output Clock Feedback Divider** Feedback Divider divides the output clock from VCO of PLL.
[6:5]	**OTDV**	**PLL Output Clock Divider** OTDV / Divided by 0 0 1 0 1 2 1 0 2 1 1 4
[4:0]	**INDV**	**PLL Input Clock Divider** Input Divider divides the input reference clock into the PLL.



The formula of output clock of PLL is:

$$F_{OUT} = F_{IN} * \frac{NF}{NR} * \frac{1}{NO}$$

F_{OUT} : Output clock of **Output Divider**
F_{IN} : External clock into the **Input Divider**
NR : Input divider value (NR = INDV + 2)
NF : Feedback divider value (NF = FBDV + 2)
NO : Output divider value (NO = OTDV)

Example Case:
The input clock frequency of EXTAL15M pin is 15MHz

PLL Output Frequency	200MHz	166MHz	133MHz	100MHz
PLLCON Reg.	0x0000_4F24	0x0000_4124	0x0000_22A2	0x0000_4F64

PLL Output Frequency	66MHz	169.34MHz (44.1K*3840)	122.88MHz (48K*2560)	
PLLCON Reg.	0x0000_2B63	0x0000_4E25	0x0000_92E7	

Power Management Control Register (PMCON)

Register	Address	R/W	Description	Reset Value
PMCON	0xB000_0214	R/W	Power Management Control Register	0x0000_0000

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

Bits	Descriptions	
[3]	**RESET**	**Software Reset** This is a software reset control bit. Set logic 1 to generate an internal reset pulse. This bit is auto-clear to logic 0 at the end of the reset pulse.
[2]	**MIDLE**	**Memory Controller IDLE enable** Setting this bit HIGH to enable memory controller enter IDLE mode, the clock source of memory controller will be halted while ARM CORE enter IDLE mode. 1 = Memory controller will enter IDLE mode when IDLE bit is set. 0 = Memory controller still active when IDLE bit is set.
[1]	**PD**	**Power Down Enable** Setting this bit HIGH, this chip enters power saving mode. The clock source 15M crystal oscillator and PLL both will stop to generate clock. User can use nIRQ [7:0], USB Device, RTC, Keypad and external nRESET to wakeup chip. 1 = Power down mode enable 0 = Normal mode
[0]	**IDLE**	**CPU IDLE mode Enable** Setting this bit HIGH, ARM CPU Core enters power saving mode. The peripherals still working if the clock enable bit in **CONSEL** is set. Any nIRQ or nFIQ to ARM core to exit IDLE state. 1 = CPU IDLE mode enable 0 = Normal mode

IRQ Wakeup Control Register (IRQWAKECON)

Register	Address	R/W	Description	Reset Value
IRQWAKECON	0xB000_0218	R/W	IRQ Wakeup Control Register	0x0000_0000

31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED				IRQWAKEUPPOL0			
7	6	5	4	3	2	1	0
RESERVED				IRQWAKEUPEN0			

Bits	Descriptions	
[11:8]	**IRQWAKEUPPOL0**	**Wakeup Polarity for nIRQ[0]** 1 = nIRQx is high level wakeup 0 = nIRQx is low level wakeup Bit [3:1] are reserved.
[3:0]	**IRQWAKEUPEN0**	**Wakeup Enable for nIRQ[0]** 1 = nIRQx wakeup enable 0 = nIRQx wakeup disable Bit [3:1] are reserved.

IRQ Wakeup Flag Register (IRQWAKEFLAG)

Register	Address	R/W	Description	Reset Value
IRQWAKEFLAG	0xB000_021C	R/W	IRQ Wakeup Flag Register	0x0000_0000

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

Bits	Descriptions	
[0]	**IRQWAKEFLAG**	**Wakeup Flag for nIRQ[0]** After power down wakeup, software should check these flags to identify which IRQ is used to wakeup the system. **And clear the flags in IRQ interrupt service routine.** 1 = CPU is wakeup by nIRQx 0 = not wakeup

IP Software Reset Register (IPSRST)

Register	Address	R/W	Description	Reset Value
IPSRST	0xB000_0220	W	IP Software Reset Register	0x0000_0000

31	30	29	28	27	26	25	24
RESERVED	I2C	USI	RESERVED				
23	22	21	20	19	18	17	16
RESERVED				TIMER	Reserved	RESERVED	
15	14	13	12	11	10	9	8
RESERVED				UART	RESERVED	USBH	USBD
7	6	5	4	3	2	1	0
EMC	RESERVED	DMAC	FMI	Reserved			

Bits	Descriptions	
[30]	**I2C**	**I2C Interface Software Reset Control Bit** 0 = write 0 is no action for both I2C0 and I2C1 1 = write 1 , a reset pulse is generated to reset both I2C0 and I2C1, and This bit will be auto clear to zero.
[29]	**USI**	**USI Software Reset Control Bit** 0 = write 0 is no action for USI 1 = write 1 , a reset pulse is generated to reset USI, and This bit will be auto clear to zero.
[19]	**TIMER**	**Timer Software Reset Control Bit** 0 = write 0 is no action for all of TIMERs and WDT 1 = write 1 , a reset pulse is generated to reset all of TIMERs and WDT, and This bit will be auto clear to zero.
[11]	**UART**	**UART Software Reset Control Bit** 0 = write 0 is no action for all of UARTs 1 = write 1 , a reset pulse is generated to reset all of UARTs, and This bit will be auto clear to zero.
[9]	**USBH**	**USB Software Reset Control Bit** 0 = write 0 is no action for USB Host Controller 1 = write 1 , a reset pulse is generated to reset USB Host Controller, and This bit will be auto clear to zero.
[8]	**USBD**	**USB Device Software Reset Control Bit** 0 = write 0 is no action for USB Device Controller 1 = write 1 , a reset pulse is generated to reset USB Device Controller, and This bit will be auto clear to zero.

[7]	**EMC**	**EMC Software Reset Control Bit** 0 = write 0 is no action for EMC Controller 1 = write 1 , a reset pulse is generated to reset EMC Controller, and This bit will be auto clear to zero.
[5]	**DMAC**	**DMAC Software Reset Control Bit** 0 = write 0 is no action for DMA Controller 1 = write 1 , a reset pulse is generated to reset DMA Controller, and This bit will be auto clear to zero.
[4]	**FMI**	**FMI Software Reset Control Bit** 0 = write 0 is no action for FMI Controller 1 = write 1 , a reset pulse is generated to reset FMI Controller, and This bit will be auto clear to zero.

Clock Enable 1 Register (CLKEN1)

Register	Address	R/W	Description	Reset Value
CLKEN1	0xB000_0224	R/W	Clock Enable 1 Register	0x0000_0000

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

Bits	Descriptions	
[2]	**RMII**	**RMII Clock Enable Bit** 0 = Disable RMII clock 1 = Enable RMII clock
[1]	**SD**	**SD Clock Enable Bit** 0 = Disable SD clock 1 = Enable SD clock
[0]	**MS**	**MS Clock Enable Bit** 0 = Disable MS clock 1 = Enable MS clock

Clock Divider Control 1 Register (CLKDIV1)

Register	Address	R/W	Description	Reset Value
CLKDIV1	0xB000_0228	R/W	Clock Divider Control 1 Register	0x0000_0000

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

Bits	Descriptions	
[15:8]	**SD_DIV**	**SD divider** **SD_CLK = Source Clock / (SD_DIV +1)** Where Source Clock selection is controlled by MSDSEL of register CLKSEL.
[7:0]	**MS_DIV**	**MS divider** **MS_CLK = Source Clock / (MS_DIV +1)** Where Source Clock selection is controlled by MSDSEL of register CLKSEL.

7.4 External Bus Interface

7.4.1 Overview

This chip supports External Bus Interface (**EBI**), which controls the access to the external memory (ROM/FLASH, SDRAM) and External I/O devices. The **EBI** has chip select signals to select one ROM/FLASH bank, two SDRAM banks, and five External I/O banks with 22-bit address bus. It supports 8-bit and 16-bit external data bus width for each bank.

The EBI has the following functions：

- SDRAM controller
- EBI control register
- ROM/FLASH interface
- External I/O interface

7.4.2 Functional Description

7.4.2.1 SDRAM Controller

The SDRAM controller module contains configuration registers, timing control registers, common control register and other logic to provide 8 or 16 bits SDRAM interface with a single 8 or 16 bits SDRAM device or two 8-bit devices wired to give a 16-bit data path.
The SDRAM controller has the following features：

- Supports up to 1 external SDRAM devices
- Maximum size of each device is 128M bytes
- 8 or 16-bit data interface
- Programmable CAS Latency： 1,2 and 3
- Fixed Burst Length： 1
- Sequential burst type
- Write Burst Length mode is Burst
- Auto Refresh Mode and Self Refresh Mode
- Adjustable Refresh Rate
- Power up sequence

7.4.2.2 SDRAM Components Supported

Table: SDRAM Components supported

Size	Type	Banks	Row Addressing	Column Addressing
16M bits	2Mx8	2	RA0~RA10	CA0~CA8
	1Mx16	2	RA0~RA10	CA0~CA7
64M bits	8Mx8	4	RA0~RA11	CA0~CA8
	4Mx16	4	RA0~RA11	CA0~CA7
128M bits	16Mx8	4	RA0~RA11	CA0~CA9
	8Mx16	4	RA0~RA11	CA0~CA8
256M bits	32Mx8	4	RA0~RA12	CA0~CA9
	16Mx16	4	RA0~RA12	CA0~CA8
512M bits	64Mx8	4	RA0~RA12	CA0~CA9,CA11
	32Mx16	4	RA0~RA12	CA0~CA9

7.4.2.3 AHB Bus Address Mapping to SDRAM Bus

Note: * indicates the signal is not used; ** indicates the signal is fixed at logic 0 and is not used;
The HADDR prefixes have been omitted on the following tables.
MA14 ~ MA0 are the Address pins of the EBI interface;
MA14 and MA13 are also the bank selected signals of SDRAM.

SDRAM Data Bus Width: 16-bit

Total	Type	R x C	R / C	MA14 (BS1)	MA13 (BS0)	MA12	MA11	MA10	MA9	MA8	MA7	MA6	MA5	MA4	MA3	MA2	MA1	MA0
16M	2Mx8	11x9	R	**	10	**	10*	21	20	19	18	17	16	15	14	13	12	11
			C	**	**10**	**	10*	AP	24*	**9**	**8**	**7**	**6**	**5**	**4**	**3**	2	1
16M	1Mx16	11x8	R	**	9	**	9*	**10**	20	19	18	17	16	15	14	13	12	11
			C	**	9	**	9*	AP	24*	9*	**8**	**7**	**6**	**5**	**4**	**3**	2	1
64M	8Mx8	12x9	R	10	11	10*	22	21	20	19	18	17	16	15	14	13	12	**23**
			C	10	11	10*	22*	AP	24*	**9**	**8**	**7**	**6**	**5**	**4**	**3**	2	1
64M	4Mx16	12x8	R	10	9	10*	22	21	20	19	18	17	16	15	14	13	12	11
			C	10	9	10*	22*	AP	24*	23*	**8**	**7**	**6**	**5**	**4**	**3**	2	1
128M	16Mx8	12x10	R	10	11	10*	22	21	20	19	18	17	16	15	14	13	12	**23**
			C	10	11	10*	22*	AP	**24**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	2	1
128M	8Mx16	12x9	R	10	11	10*	22	21	20	19	18	17	16	15	14	13	12	**23**
			C	10	11	10*	22*	AP	24*	**9**	**8**	**7**	**6**	**5**	**4**	**3**	2	1
256M*	32Mx8	13x10	R	10	11	23	22	21	20	19	18	17	16	15	14	13	12	**24**
			C	10	11	23*	22*	AP	25	**9**	**8**	**7**	**6**	**5**	**4**	**3**	2	1
256M	16Mx16	13x9	R	10	11	23	22	21	20	19	18	17	16	15	14	13	12	**24**
			C	10	11	23*	22*	AP	25*	9	**8**	**7**	**6**	**5**	**4**	**3**	2	1
512M	64Mx8	13x11	R	10	11	23	22	21	20	19	18	17	16	15	14	13	12	**24**
			C	10	11	23*	26	AP	25	9	**8**	**7**	**6**	**5**	**4**	**3**	2	1
512M	32Mx16	13x10	R	10	11	23	22	21	20	19	18	17	16	15	14	13	12	**24**
			C	10	11	23*	22*	AP	25	9	**8**	**7**	**6**	**5**	**4**	**3**	2	1

SDRAM Data Bus Width: 8-bit

Total	Type	R x C	R/C	MA14 (BS1)	MA13 (BS0)	MA12	MA11	MA10	MA9	MA8	MA7	MA6	MA5	MA4	MA3	MA2	MA1	MA0
16M	2Mx8	11x9	R	**	9	**	9*	20	19	18	17	16	15	14	13	12	11	10
			C	**	**9**	**	9*	AP	23*	**8**	**7**	**6**	**5**	**4**	**3**	2	1	0
16M	1Mx16	11x8	R	**	8	**	8*	**9**	19	18	17	16	15	14	13	12	11	10
			C	**	8	**	8*	AP	23*	8*	**7**	**6**	**5**	**4**	**3**	2	1	0
64M	8Mx8	12x9	R	9	10	9*	21	20	19	18	17	16	15	14	13	12	11	**22**
			C	9	10	9*	21*	AP	23*	**8**	**7**	**6**	**5**	**4**	**3**	2	1	1
64M	4Mx16	12x8	R	9	8	9*	21	20	19	18	17	16	15	14	13	12	11	10
			C	9	8	9*	21*	AP	23*	22*	**7**	**6**	**5**	**4**	**3**	2	1	0
128M	16Mx8	12x10	R	9	10	9*	21	20	19	18	17	16	15	14	13	12	11	**22**
			C	9	10	9*	21*	AP	**23**	**8**	**7**	**6**	**5**	**4**	**3**	2	1	0
128M	8Mx16	12x9	R	9	10	9*	21	20	19	18	17	16	15	14	13	12	11	**22**
			C	9	10	9*	21*	AP	23*	**8**	**7**	**6**	**5**	**4**	**3**	2	1	0
256M	32Mx8	13x10	R	9	10	22	21	20	19	18	17	16	15	14	13	12	11	**23**
			C	9	10	22*	21*	AP	24	**8**	**7**	**6**	**5**	**4**	**3**	2	1	0
256M	16Mx16	13x9	R	9	10	22	21	20	19	18	17	16	15	14	13	12	11	**23**
			C	9	10	22*	21*	AP	24*	8	**7**	**6**	**5**	**4**	**3**	2	1	0
512M	64Mx8	13x11	R	9	10	22	21	20	19	18	17	16	15	14	13	12	11	**23**
			C	9	10	22*	25	AP	24	8	**7**	**6**	**5**	**4**	**3**	2	1	0
512M	32Mx16	13x10	R	9	10	22	21	20	19	18	17	16	15	14	13	12	11	**23**
			C	9	10	22*	21*	AP	24	8	**7**	**6**	**5**	**4**	**3**	2	1	0

7.4.2.4 SDRAM Power-Up Sequence

The default value of the mode register is not defined, therefore the mode register must be written after power up to operate the SDRAM. This chip supports the function of Power-Up Sequence, that is, after system power on, the SDRAM Controller automatically executes the commands needed for Power-Up sequence and set the mode register of each bank to default value. The default value is :

- Burst Length = 1
- Burst Type = Sequential (fixed)
- CAS Latency = 2
- Write Burst Length = Burst (fixed)

The value of mode register can be changed after power up sequence by setting the value of corresponding bank's configuration register "**LENGTH**" and "**LATENCY**" bits and set the **MRSET** bit enable to execute the Mode Register Set command.

7.4.3 EBI Register Mapping

Register	Offset	R/W	Description	Reset Value
(EBI_BA=0xB000_1000)				
EBICON	0xB000_1000	R/W	EBI control register	0x0001_0001
ROMCON	0xB000_1004	R/W	ROM/FLASH control register	0x0000_0FFX
SDCONF0	0xB000_1008	R/W	SDRAM bank 0 configuration register	0x0000_0800
SDTIME0	0xB000_1010	R/W	SDRAM bank 0 timing control register	0x0000_0000
EXT0CON	0xB000_1018	R/W	External I/O 0 control register	0x0000_0000
CKSKEW	0xB000_102C	R/W	Clock skew control register	0xXXXX_0048

7.4.4 EBI Register Details

EBI Control Register (EBICON)

Register	Address	R/W	Description	Reset Value
EBICON	0xB000_1000	R/W	EBI Control Register	0x0001_0001

31	30	29	28	27	26	25	24
RESERVED							EXBE0
23	22	21	20	19	18	17	16
Reserved					REFEN	REFMOD	CLKEN
15	14	13	12	11	10	9	8
REFRAT							
7	6	5	4	3	2	1	0
REFRAT					WAITVT		LITTLE

Bits	Descriptions	
[24]	**EXBE0**	**EXBE0: External IO Bank 0 Byte Enable** 0: nWBE[1:0] pin is byte write strobe signal 1: nWBE[1:0] pin is byte enable signals, nSWE will be used as write strobe signal to SRAM
[23:19]	**Reserved**	Write 0 for normal operation
[18]	**REFEN**	**Enable SDRAM refresh cycle for SDRAM bank0** This bit set will start the auto-refresh cycle to SDRAM. The refresh rate is according to REFRAT bits.
[17]	**REFMOD**	**The refresh mode of SDRAM for SDRAM bank** Defines the refresh mode type of external SDRAM bank 0 = Auto refresh mode 1 = Self refresh mode
[16]	**CLKEN**	**Clock enable for SDRAM** Enables the SDRAM clock enable (**CKE**) control signal 0 = Disable (power down mode) 1 = Enable （Default）
[15:3]	**REFRAT**	**Refresh count value for SDRAM** The refresh period is calculated as $period = \frac{value}{fMCLK}$ The SDRAM Controller automatically provides an auto refresh cycle for every refresh period programmed into the **REFRAT** bits when the **REFEN** bit of each bank is set.

[2:1]	**WAITVT**	**Valid time of nWAIT signal** This bit recognizes the **nWAIT** signal at the next "nth" **MCLK** rising edge after the **nOE** or **nWBE** active cycle. **WAITVT** bits determine the n. **WAITVT [2:1]** | **nth MCLK** 0 0 1 0 1 2 1 0 3 1 1 4
[0]	**LITTLE**	**Little Endian mode** This bit always set to a logic 1 (Read Only)

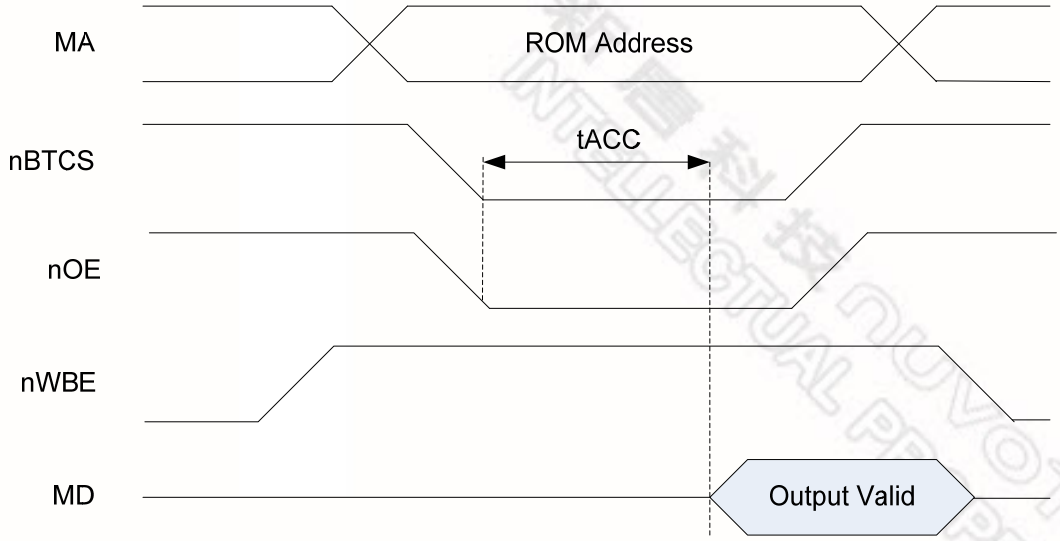
ROM/Flash Control Register（ROMCON）

Register	Address	R/W	Description	Reset Value
ROMCON	0xB000_1004	R/W	ROM/FLASH Control Register	0x0000_0FFX

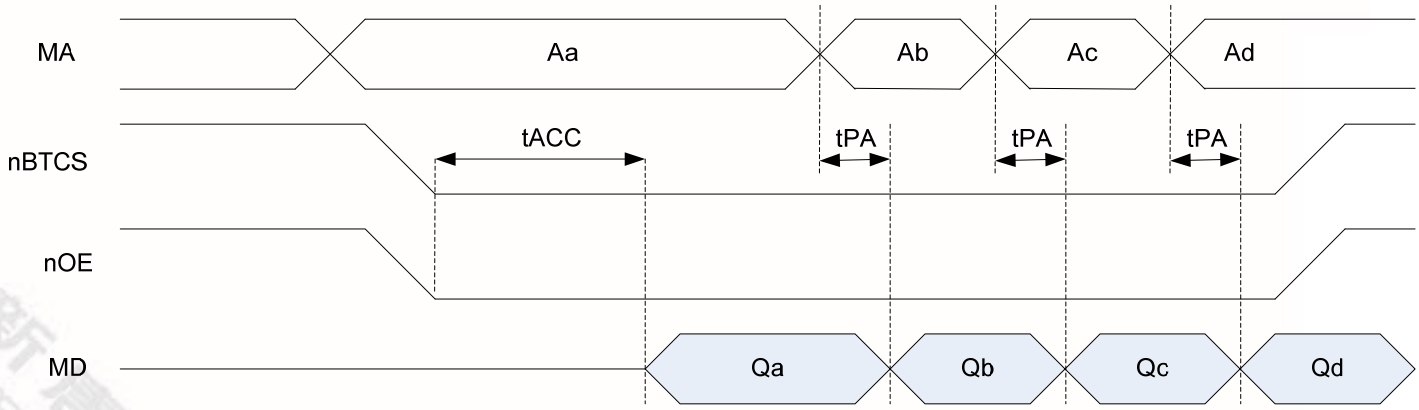
31	30	29	28	27	26	25	24
BASADDR							
23	22	21	20	19	18	17	16
BASADDR					SIZE		
15	14	13	12	11	10	9	8
SIZE	Reserved			tPA			
7	6	5	4	3	2	1	0
tACC				BTSIZE		PGMODE	

Bits	Descriptions	
[31:19]	**BASADDR**	**Base Address Pointer of ROM/Flash Bank** The start address is calculated as ROM/Flash bank base pointer << 18. The base address pointer together with the "**SIZE**" bits constitutes the whole address range of each bank.
[18:15]	**SIZE**	**Size of ROM/FLASH Memory** SIZE [18:15] / Byte 0 0 0 0 / 256K 0 0 1 0 / 512K 0 1 0 0 / 1M 0 1 1 0 / 2M 1 0 0 0 / 4M 1 0 1 0 / 8M Others / Reserved .
[11:8]	**tPA**	**Page Mode Access Cycle Time** tPA[11:8] / MCLK / tPA[11:8] / MCLK 0 0 0 0 / 1 / 1 0 0 0 / 10 0 0 0 1 / 2 / 1 0 0 1 / 12 0 0 1 0 / 3 / 1 0 1 0 / 14 0 0 1 1 / 4 / 1 0 1 1 / 16 0 1 0 0 / 5 / 1 1 0 0 / 18 0 1 0 1 / 6 / 1 1 0 1 / 20 0 1 1 0 / 7 / 1 1 1 0 / 22 0 1 1 1 / 8 / 1 1 1 1 / 24

[7:4]	**tACC**	**Access Cycle Time** **tACC[7:4]** / **MCLK** / **tACC[7:4]** / **MCLK** 0 0 0 0 / 3 / 1 0 0 0 / 10 0 0 0 1 / 3 / 1 0 0 1 / 12 0 0 1 0 / 3 / 1 0 1 0 / 14 0 0 1 1 / 4 / 1 0 1 1 / 16 0 1 0 0 / 5 / 1 1 0 0 / 18 0 1 0 1 / 6 / 1 1 0 1 / 20 0 1 1 0 / 7 / 1 1 1 0 / 22 0 1 1 1 / 8 / 1 1 1 1 / 24
[3:2]	**BTSIZE**	**Boot ROM/FLASH Data Bus Width** This ROM/Flash bank is designed for a boot ROM. **BASADDR** bits determine its start address. The external data bus width is determined by power-on setting when booting from external ROM. **BTSIZE [3:2]** / **Bus Width** 0 0 / 8-bit 0 1 / 16-bit 1 0 / RESERVED 1 1 / RESERVED
[1:0]	**PGMODE**	**Page Mode Configuration** **PGMODE [1:0]** / **Mode** 0 0 / Normal ROM 0 1 / 4 word page 1 0 / 8 word page 1 1 / 16 word page



ROM/FLASH Read Operation Timing



ROM/FLASH Page Read Operation Timing

SDRAM Configuration Register (SDCONF0)

The configuration registers enable software to set a number of operating parameters for the SDRAM controller. There are two configuration registers SDCONF0 for SDRAM bank 0 respectively. Each bank can have a different configuration.

Register	Address	R/W	Description	Reset Value
SDCONF0	0xB000_1008	R/W	SDRAM Bank 0 Configuration Register	0x0000_0800

31	30	29	28	27	26	25	24
BASADDR							
23	22	21	20	19	18	17	16
BASADDR					RESERVED		
15	14	13	12	11	10	9	8
MRSET	RESERVED	AUTOPR	LATENCY		RESERVED		
7	6	5	4	3	2	1	0
COMPBK	DBWD		COLUMN		SIZE		

Bits	Descriptions	
[31:19]	**BASADDR**	**Base Address Pointer of SDRAM Bank 0** The start address is calculated as SDRAM bank 0 base pointer << 18. The SDRAM base address pointer together with the "**SIZE**" bits constitutes the whole address range of each SDRAM bank.
[15]	**MRSET**	**SDRAM Mode Register Set Command for SDRAM Bank 0** This bit set will issue a mode register set command to SDRAM.
[13]	**AUTOPR**	**Auto Pre-charge Mode of SDRAM for SDRAM Bank 0** Enable the auto pre-charge function of external SDRAM bank 0 0 = Auto pre-charge 1 = No auto pre-charge
[12:11]	**LATENCY**	**The CAS Latency of SDRAM Bank 0** Defines the CAS latency of external SDRAM bank 0

LATENCY [12:11]		MCLK
0	0	1
0	1	2
1	0	3
1	1	REVERSED

[7]	**COMPBK**	**Number of Component Bank in SDRAM Bank 0** Indicates the number of component bank (2 or 4 banks) in external SDRAM bank 0. 0 = 2 banks 1 = 4 banks

<table>
<tr>
<td>[6:5]</td>
<td>DBWD</td>
<td>Data Bus Width for SDRAM Bank 0
Indicates the external data bus width connect with SDRAM bank 0
If DBWD = 00, the assigned SDRAM access signal is not generated i.e. disable.
<table>
<tr><th colspan="2">DBWD [6:5]</th><th>Bits</th></tr>
<tr><td>0</td><td>0</td><td>Bank disable</td></tr>
<tr><td>0</td><td>1</td><td>8-bit (byte)</td></tr>
<tr><td>1</td><td>0</td><td>16-bit (half-word)</td></tr>
<tr><td>1</td><td>1</td><td>RESERVED</td></tr>
</table>
</td>
</tr>
<tr>
<td>[4:3]</td>
<td>COLUMN</td>
<td>Number of Column Address bits in SDRAM Bank 0
Indicates the number of column address bits in external SDRAM bank 0.
<table>
<tr><th colspan="2">COLUMN [4:3]</th><th>Bits</th></tr>
<tr><td>0</td><td>0</td><td>8</td></tr>
<tr><td>0</td><td>1</td><td>9</td></tr>
<tr><td>1</td><td>0</td><td>10</td></tr>
<tr><td>1</td><td>1</td><td>11</td></tr>
</table>
</td>
</tr>
<tr>
<td>[2:0]</td>
<td>SIZE</td>
<td>Size of SDRAM Bank 0
Indicates the memory size of external SDRAM bank 0
<table>
<tr><th colspan="3">SIZE [2:0]</th><th>Size of SDRAM （Byte）</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>Bank disable</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>2M</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>4M</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>8M</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>16M</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>32M</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>64M</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>Reserved</td></tr>
</table>
</td>
</tr>
</table>

SDRAM Timing Control Register (SDTIME0)

Register	Address	R/W	Description	Reset Value
SDTIME0	0xB000_1010	R/W	SDRAM bank 0 timing control register	0x0000_0000

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

Bits	Descriptions	
[10:8]	tRCD	SDRAM Bank 0, /RAS to /CAS Delay
[7:6]	tRDL	SDRAM Bank 0, Last Data in to Pre-charge Command

tRCD [10:8]			MCLK
0	0	0	1
0	0	1	2
0	1	0	3
0	1	1	4
1	0	0	5
1	0	1	6
1	1	0	7
1	1	1	8

tRDL [7:6]		MCLK
0	0	1
0	1	2
1	0	3
1	1	4

<table>
<tr><td>[5:3]</td><td>tRP</td><td>SDRAM Bank 0, Row Pre-charge Time
<table>
<tr><th colspan="3">tRP [5:3]</th><th>MCLK</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>1</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>2</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>3</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>4</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>5</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>6</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>7</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>8</td></tr>
</table></td></tr>
<tr><td>[2:0]</td><td>tRAS</td><td>SDRAM Bank 0, Row Active Time
<table>
<tr><th colspan="3">tRAS [2:0]</th><th>MCLK</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>1</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>2</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>3</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>4</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>5</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>6</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>7</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>8</td></tr>
</table></td></tr>
</table>

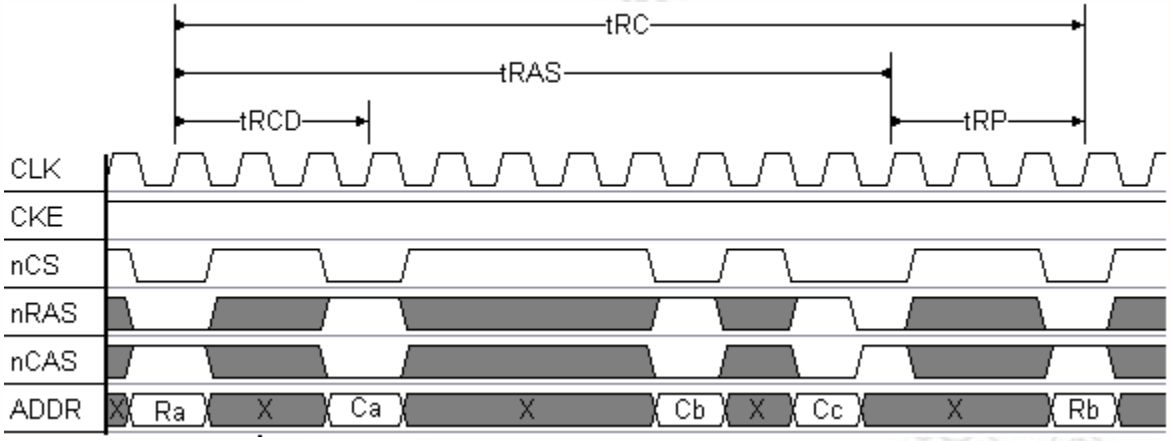


Fig. Access timing 1 of SDRAM

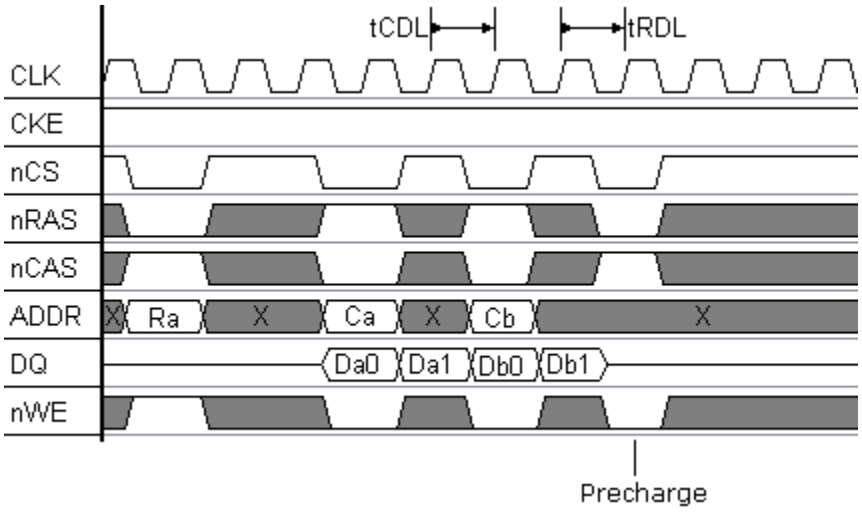


Fig. Access timing 2 of SDRAM

External I/O Control Registers (EXT0CON)

Register	Address	R/W	Description	Reset Value
EXT0CON	0xB000_1018	R/W	External I/O 0 control register	0x0000_0000

31	30	29	28	27	26	25	24
BASADDR							
23	22	21	20	19	18	17	16
BASADDR					SIZE		
15	14	13	12	11	10	9	8
ADRS	tACC				tCOH		
7	6	5	4	3	2	1	0
tACS			tCOS			DBWD	

Bits	Descriptions	
[31:19]	**BASADDR**	**Base Address Pointer of External I/O Bank 0** The start address of each external I/O bank is calculated as "**BASADDR"** base pointer << 18. Each external I/O bank base address pointer together with the "**SIZE**" bits constitutes the whole address range of each external I/O bank.
[18:16]	**SIZE**	**The Size of the External I/O Bank 0** **SIZE [18:16]** / **Byte** 0 0 0 – 256K 0 0 1 – 512K 0 1 0 – 1M 0 1 1 – 2M 1 0 0 – 4M 1 0 1 – 8M
[15]	**ADRS**	**Address Bus Alignment for External I/O Bank 0** When **ADRS** is set, EBI bus is alignment to byte address format, and ignores **DBWD [1:0]** setting.

[14:11] tACC

Access Cycles （nOE or nSWE active time）for External I/O Bank 0

tACC[14:11]				MCLK	tACC[14:11]				MCLK
0	0	0	0	Reversed	1	0	0	0	9
0	0	0	1	1	1	0	0	1	11
0	0	1	0	2	1	0	1	0	13
0	0	1	1	3	1	0	1	1	15
0	1	0	0	4	1	1	0	0	17
0	1	0	1	5	1	1	0	1	19
0	1	1	0	6	1	1	1	0	21
0	1	1	1	7	1	1	1	1	23

[10:8] tCOH

Chip Selection Hold-On Time on nOE or nWBE for External I/O Bank 0

tCOH [10:8]			MCLK
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6
1	1	1	7

[7:5] tACS

Address Set-up Before nECS for External I/O Bank 0

tACS [7:5]			MCLK
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6
1	1	1	7

<table>
<tr>
<td>[4:2]</td>
<td>tCOS</td>
<td>Chip Selection Set-up Time on nOE or nWBE for External I/O Bank 0
When the bank is configured, the access to its bank stretches chip selection time before the nOE or new signal is activated.
<table>
<tr><th colspan="3">tCOS [4:2]</th><th>MCLK</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>1</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>2</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>3</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>4</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>5</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>6</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>7</td></tr>
</table>
</td>
</tr>
<tr>
<td>[1:0]</td>
<td>DBWD</td>
<td>Programmable Data Bus Width for External I/O Bank 0
<table>
<tr><th colspan="2">DBWD [1:0]</th><th>Width of Data Bus</th></tr>
<tr><td>0</td><td>0</td><td>Disable bus</td></tr>
<tr><td>0</td><td>1</td><td>8-bit</td></tr>
<tr><td>1</td><td>0</td><td>16-bit</td></tr>
<tr><td>1</td><td>1</td><td>RESERVED</td></tr>
</table>
</td>
</tr>
</table>

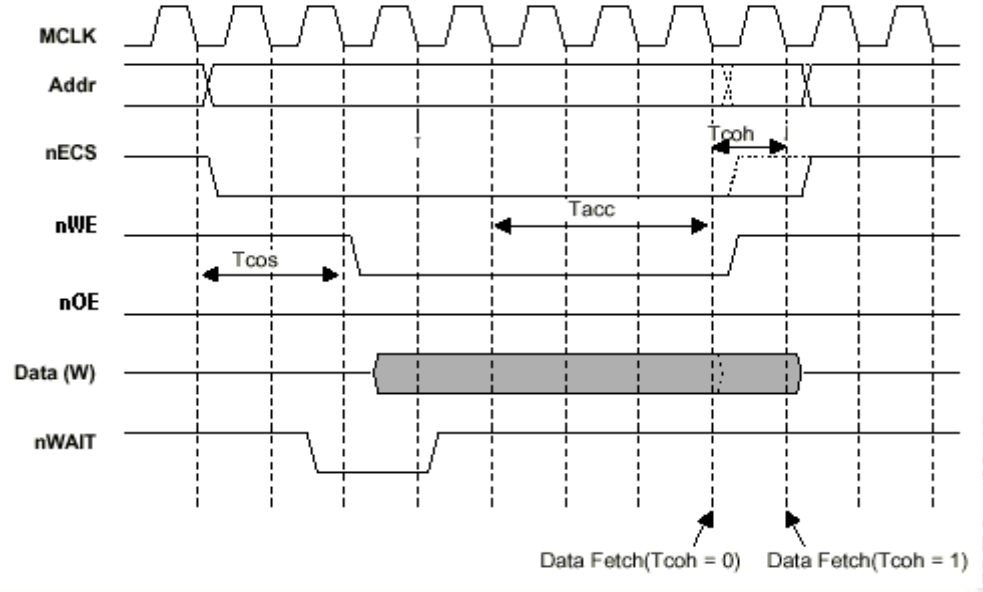


Fig. External I/O Write operation timing

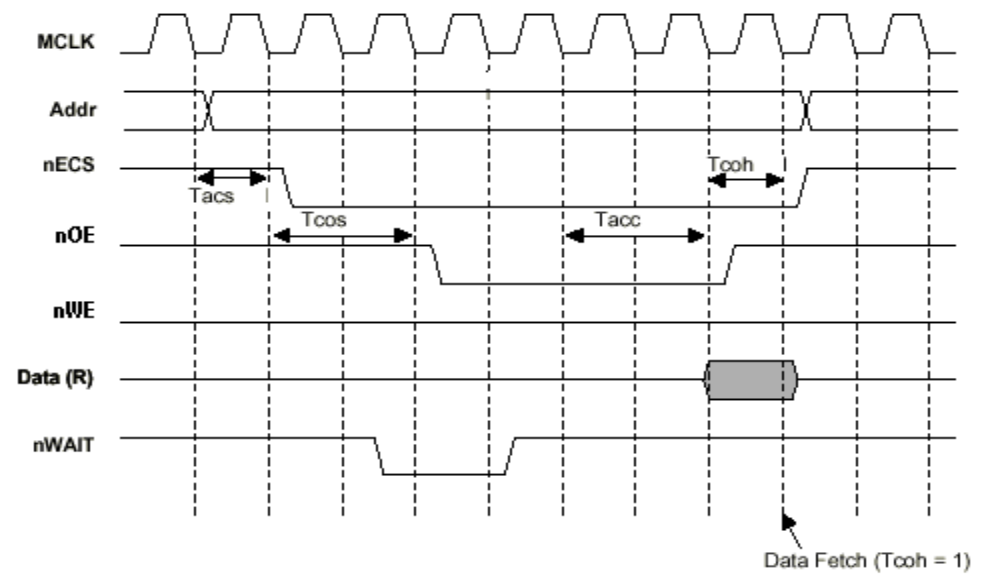


Fig. External I/O Read operation timing

Clock Skew Control Register (CKSKEW)

Register	Address	R/W	Description	Reset Value
CKSKEW	0xB000_102C	R/W	Clock Skew Control Register	0xXXXX_0048

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

Bits: [7:4] DLH_CLK_SKEW

Data Latch Clock Skew Adjustment

DLH_CLK_SKEW[7:4]				Gate Delay	DLH_CLK_SKEW[7:4]				Gate Delay
0	0	0	0	P-0	1	0	0	0	N-0
0	0	0	1	P-1	1	0	0	1	N-1
0	0	1	0	P-2	1	0	1	0	N-2
0	0	1	1	P-3	1	0	1	1	N-3
0	1	0	0	P-4	1	1	0	0	N-4
0	1	0	1	P-5	1	1	0	1	N-5
0	1	1	0	P-6	1	1	1	0	N-6
0	1	1	1	P-7	1	1	1	1	N-7

Note: P-x means Data latched Clock shift "X" gates delays by refer MCLKO positive edge;
N-x means Data latched Clock shift "X" gates delays by refer MCLKO negative edge.

Bits: [3:0] MCLK_O_D

MCLK Output Delay Adjustment

MCLK_O_D [3:0]				Gate Delay	MCLK_O_D [3:0]				Gate Delay
0	0	0	0	P-0	1	0	0	0	N-0
0	0	0	1	P-1	1	0	0	1	N-1
0	0	1	0	P-2	1	0	1	0	N-2
0	0	1	1	P-3	1	0	1	1	N-3
0	1	0	0	P-4	1	1	0	0	N-4
0	1	0	1	P-5	1	1	0	1	N-5
0	1	1	0	P-6	1	1	1	0	N-6
0	1	1	1	P-7	1	1	1	1	N-7

Note:
P-x means MCLKO shift "X" gates delay by refer HCLK positive edge;
N-x means MCLKO shift "X" gates delay by refer HCLK negative edge.
MCLK is the output pin of MCLKO, which is an internal signal on chip.

7.5 Ethernet MAC Controller

Overview

This chip provides an Ethernet MAC Controller (EMC) for WAN/LAN application. This EMC has its DMA controller, transmit FIFO, and receive FIFO.

The Ethernet MAC controller consists of IEEE 802.3/Ethernet protocol engine with internal CAM function for Ethernet MAC address recognition, Transmit-FIFO, Receive-FIFO, TX/RX state machine controller and status controller. The EMC only supports RMII (Reduced MII) interface to connect with PHY operating on 50MHz REF_CLK.

Features

- Supports IEEE Std. 802.3 CSMA/CD protocol.
- Supports both half and full duplex for 10M/100M bps operation.
- Supports RMII interface.
- Supports MII Management function.
- Supports pause and remote pause function for flow control.
- Supports long frame (more than 1518 bytes) and short frame (less than 64 bytes) reception.
- Supports 16 entries CAM function for Ethernet MAC address recognition.
- Supports internal loop back mode for diagnostic.
- Supports 256 bytes embedded transmit and receive FIFO.
- Supports DMA function.

EMC Descriptors

A link-list data structure named as descriptor is used to keep the control, status and data information of each frame. Through the descriptor, CPU and EMC exchange the information for frame reception and transmission.

Two different descriptors are defined in NUC946ADN. One named as Rx descriptor for frame reception and the other names as Tx descriptor for frame transmission. Each Rx descriptor consists of four words. There is much information kept in the descriptors and details are described as below.

7.5.1.1 Rx Buffer Descriptor

31	30	29	16	15	0
O	Rx Status			Receive Byte Count	
Receive Buffer Starting Address					BO
Reserved					
Next Rx Descriptor Starting Address					

Rx Descriptor Word 0

31	30	29	28	27	26	25	24
Owner		Reserved					
23	22	21	20	19	18	17	16
Reserved	RP	ALIE	RXGD	PTLE	Reserved	CRCE	RXINTR
15	14	13	12	11	10	9	8
RBC							
7	6	5	4	3	2	1	0
RBC							

Bits	Descriptions	
[31:30]	**Owner**	**Ownership** The ownership field defines which one, the CPU or EMC, is the owner of each Rx descriptor. Only the owner has right to modify the Rx descriptor and the others can read the Rx descriptor only. 00: The owner is CPU 01: Undefined 10: The owner is EMC 11: Undefined If the O=2'b10 indicates the EMC RxDMA is the owner of Rx descriptor and the Rx descriptor is available for frame reception. After the frame reception completed, if the frame needed NAT translation, EMC RxDMA modify ownership field to 2'b11. Otherwise, the ownership field will be modified to 2'b00. If the O=2'b00 indicates the CPU is the owner of Rx descriptor. After the CPU completes processing the frame, it modifies the ownership field to 2'b10 and releases the Rx descriptor to EMC RxDMA.
[29:23]	**Rx Status**	**Receive Status** This field keeps the status for frame reception. All status bits are updated by EMC. In the receive status, bits 29 to 23 are undefined and reserved for the future.
[22]	**RP**	**Runt Packet** The RP indicates the frame stored in the data buffer pointed by Rx descriptor is a short frame (frame length is less than 64 bytes). 1'b0: The frame is not a short frame. 1'b1: The frame is a short frame.
[21]	**ALIE**	**Alignment Error** The ALIE indicates the frame stored in the data buffer pointed by Rx descriptor is not a multiple of byte. 1'b0: The frame is a multiple of byte. 1'b1: The frame is not a multiple of byte.
[20]	**RXGD**	**Frame Reception Complete** The RXGD indicates the frame reception has completed and stored in the data buffer pointed by Rx descriptor. 1'b0: The frame reception not complete yet. 1'b1: The frame reception completed.
[19]	**PTLE**	**Packet Too Long** The PTLE indicates the frame stored in the data buffer pointed by Rx descriptor is a long frame (frame length is greater than 1518 bytes). 1'b0: The frame is not a long frame. 1'b1: The frame is a long frame.

[17]	**CRCE**	**CRC Error** The CRCE indicates the frame stored in the data buffer pointed by Rx descriptor incurred CRC error. 1'b0: The frame doesn't incur CRC error. 1'b1: The frame incurred CRC error.
[16]	**RXINTR**	**Receive Interrupt** The RXINTR indicates the frame stored in the data buffer pointed by Rx descriptor caused an interrupt condition. 1'b0: The frame doesn't cause an interrupt. 1'b1: The frame caused an interrupt.
[15:0]	**RBC**	**Receive Byte Count** The RBC indicates the byte count of the frame stored in the data buffer pointed by Rx descriptor. The four bytes CRC field is also included in the receive byte count. But if the SPCRC of register MCMDR is enabled, the four bytes CRC field will be excluded from the receive byte count.

Rx Descriptor Word 1

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

Bits	Descriptions	
[31:2]	**RXBSA**	**Receive Buffer Starting Address** The RXBSA indicates the starting address of the receive frame buffer. The RXBSA is used to be the bit 31 to 2 of memory address. In other words, the starting address of the receive frame buffer always located at word boundary.
[1:0]	**BO**	**Byte Offset** The BO indicates the byte offset from RXBSA where the received frame begins to store. If the BO is 2'b01, the starting address where the received frame begins to store is RXBSA+2'b01, and so on.

Rx Descriptor Word 2

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

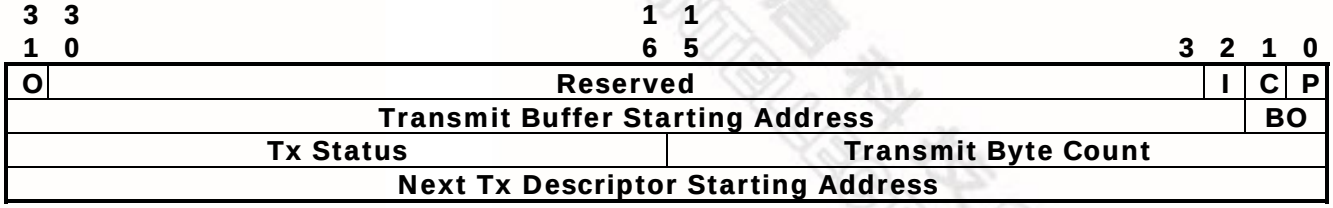
The Rx descriptor word 2 keeps obsolete information for MAC translation. Therefore, these information bits are undefined and should be ignored.

Rx Descriptor Word 3

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

Bits	Descriptions	
[31:0]	**NRXDSA**	**Next Rx Descriptor Starting Address** The Rx descriptor is a link-list data structure. Consequently, NRXDSA is used to keep the starting address of the next Rx descriptor. The bits [1:0] will be ignored by EMC. So, all Rx descriptor must locate at word boundary memory address.

7.5.1.2 Tx Buffer Descriptor



Tx Descriptor Word 0

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

Bits	Descriptions	
[31]	**Owner**	**Ownership** The ownership field defines which one, the CPU or EMC, is the owner of each Tx descriptor. Only the owner has right to modify the Tx descriptor and the other can read the Tx descriptor only. 0: The owner is CPU 1: The owner is EMC If the O=1'b1 indicates the EMC TxDMA is the owner of Tx descriptor and the Tx descriptor is available for frame transmission. After the frame transmission completed, EMC TxDMA modify ownership field to 1'b0 and return the ownership of Tx descriptor to CPU. If the O=1'b0 indicates the CPU is the owner of Tx descriptor. After the CPU prepares new frame to wait transmission, it modifies the ownership field to 1'b1 and releases the Tx descriptor to EMC TxDMA.
[2]	**IntEn**	**Transmit Interrupt Enable** The IntEn controls the interrupt trigger circuit after the frame transmission completed. If the IntEn is enabled, the EMC will trigger interrupt after frame transmission completed. Otherwise, the interrupt doesn't be triggered. 1'b0: Frame transmission interrupt is masked. 1'b1: Frame transmission interrupt is enabled.

[1]	**CRCApp**	**CRC Append** The CRCApp control the CRC append during frame transmission. If CRCApp is enabled, the 4-bytes CRC checksum will be appended to frame at the end of frame transmission. 1'b0: 4-bytes CRC appending is disabled. 1'b1: 4-bytes CRC appending is enabled.
[0]	**PadEN**	**Padding Enable** The PadEN control the PAD bits appending while the length of transmission frame is less than 60 bytes. If PadEN is enabled, EMC does the padding automatically. 1'b0: PAD bits appending is disabled. 1'b1: PAD bits appending is enabled.

Tx Descriptor Word 1

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

Bits	Descriptions	
[31:2]	**TXBSA**	**Transmit Buffer Starting Address** The TXBSA indicates the starting address of the transmit frame buffer. The TXBSA is used to be the bit 31 to 2 of memory address. In other words, the starting address of the transmit frame buffer always located at word boundary.
[1:0]	**BO**	**Byte Offset** The BO indicates the byte offset from TXBSA where the transmit frame begins to read. If the BO is 2'b01, the starting address where the transmit frame begins to read is TXBSA+2'b01, and so on.

Tx Descriptor Word 2

31	30	29	28	27	26	25	24
CCNT				Reserved	SQE	PAU	TXHA
23	22	21	20	19	18	17	16
LC	TXABT	NCS	EXDEF	TXCP	Reserved	DEF	TXINTR
15	14	13	12	11	10	9	8
TBC							
7	6	5	4	3	2	1	0
TBC							

Bits	Descriptions	
[31:28]	**CCNT**	**Collision Count** The CCNT indicates the how many collisions occurred consecutively during a packet transmission. If the packet incurred 16 consecutive collisions during transmission, the CCNT will be 4'h0 and bit TXABT will be set to 1.
[26]	**SQE**	**SQE Error** The SQE indicates the SQE error found at end of packet transmission on 10Mbps half-duplex mode. The SQE error check will only be done while both bit EnSQE of MCMDR is enabled and EMC is operating on 10Mbps half-duplex mode. 1'b0: No SQE error found at end of packet transmission. 1'b0: SQE error found at end of packet transmission.
[25]	**PAU**	**Transmission Paused** The PAU indicates the next normal packet transmission process will be paused temporally because EMC received a PAUSE control frame, or S/W set bit SDPZ of MCMDR and make EMC to transmit a PAUSE control frame out. 1'b0: Next normal packet transmission process will go on. 1'b1: Next normal packet transmission process will be paused.
[24]	**TXHA**	**P Transmission Halted** The TXHA indicates the next normal packet transmission process will be halted because the bit TXON of MCMDR is disabled be S/W. 1'b0: Next normal packet transmission process will go on. 1'b1: Next normal packet transmission process will be halted.
[23]	**LC**	**Late Collision** The LC indicates the collision occurred in the outside of 64 bytes collision window. This means after the 64 bytes of a frame has transmitted out to the network, the collision still occurred. The late collision check will only be done while EMC is operating on half-duplex mode. 1'b0: No collision occurred in the outside of 64 bytes collision window. 1'b1: Collision occurred in the outside of 64 bytes collision window.

[22]	**TXABT**	**Transmission Abort** The TXABT indicates the packet incurred 16 consecutive collisions during transmission, and then the transmission process for this packet is aborted. The transmission abort is only available while EMC is operating on half-duplex mode. 1'b0: Packet doesn't incur 16 consecutive collisions during transmission. 1'b1: Packet incurred 16 consecutive collisions during transmission.
[21]	**NCS**	**No Carrier Sense** The NCS indicates the MII I/F signal CRS doesn't active at the start of or during the packet transmission. The NCS is only available while EMC is operating on half-duplex mode. 1'b0: CRS signal actives correctly. 1'b1: CRS signal doesn't active at the start of or during the packet transmission.
[20]	**EXDEF**	**Defer Exceed** The EXDEF indicates the frame waiting for transmission has deferred over 0.32768ms on 100Mbps mode, or 3.2768ms on 10Mbps mode. The deferral exceed check will only be done while bit NDEF of MCMDR is disabled, and EMC is operating on half-duplex mode. 1'b0: Frame waiting for transmission has not deferred over 0.32768ms (100Mbps) or 3.2768ms (10Mbps). 1'b1: Frame waiting for transmission has deferred over 0.32768ms (100Mbps) or 3.2768ms (10Mbps).
[19]	**TXCP**	**Transmission Complete** The TXCP indicates the packet transmission has completed correctly. 1'b0: The packet transmission doesn't complete. 1'b1: The packet transmission has completed.
[17]	**DEF**	**Transmission Deferred** The DEF indicates the packet transmission has deferred once. The DEF is only available while EMC is operating on half-duplex mode. 1'b0: Packet transmission doesn't defer. 1'b1: Packet transmission has deferred once.
[16]	**TXINTR**	**Transmit Interrupt** The TXINTR indicates the packet transmission caused an interrupt condition. 1'b0: The packet transmission doesn't cause an interrupt. 1'b1: The packet transmission caused an interrupt.
[15:0]	**TBC**	**Transmit Byte Count** The TBC indicates the byte count of the frame stored in the data buffer pointed by Tx descriptor for transmission.

Tx Descriptor Word 3

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

Bits	Descriptions	
[31:0]	**NTXDSA**	**Next Tx Descriptor Starting Address** The Tx descriptor is a link-list data structure. Consequently, NTXDSA is used to keep the starting address of the next Tx descriptor. The bits [1:0] will be ignored by EMC. So, all Tx descriptor must locate at word boundary memory address.

7.5.2 EMC Register Mapping

The EMC implements many registers and the registers are separated into two types, the control registers and the status registers. The control registers are used by S/W to pass control information to EMC. The status registers are used to keep EMC operation status for S/W.

EMC Registers

Register	Address	R/W	Description	Reset Value
EMC_BA = 0xB000_3000				
Control Registers (44)				
CAMCMR	0xB000_3000	R/W	CAM Command Register	0x0000_0000
CAMEN	0xB000_3004	R/W	CAM Enable Register	0x0000_0000
CAM0M	0xB000_3008	R/W	CAM0 Most Significant Word Register	0x0000_0000
CAM0L	0xB000_300C	R/W	CAM0 Least Significant Word Register	0x0000_0000
CAM1M	0xB000_3010	R/W	CAM1 Most Significant Word Register	0x0000_0000
CAM1L	0xB000_3014	R/W	CAM1 Least Significant Word Register	0x0000_0000
CAM2M	0xB000_3018	R/W	CAM2 Most Significant Word Register	0x0000_0000
CAM2L	0xB000_301C	R/W	CAM2 Least Significant Word Register	0x0000_0000
CAM3M	0xB000_3020	R/W	CAM3 Most Significant Word Register	0x0000_0000
CAM3L	0xB000_3024	R/W	CAM3 Least Significant Word Register	0x0000_0000
CAM4M	0xB000_3028	R/W	CAM4 Most Significant Word Register	0x0000_0000
CAM4L	0xB000_302C	R/W	CAM4 Least Significant Word Register	0x0000_0000
CAM5M	0xB000_3030	R/W	CAM5 Most Significant Word Register	0x0000_0000
CAM5L	0xB000_3034	R/W	CAM5 Least Significant Word Register	0x0000_0000
CAM6M	0xB000_3038	R/W	CAM6 Most Significant Word Register	0x0000_0000
CAM6L	0xB000_303C	R/W	CAM6 Least Significant Word Register	0x0000_0000
CAM7M	0xB000_3040	R/W	CAM7 Most Significant Word Register	0x0000_0000
CAM7L	0xB000_3044	R/W	CAM7 Least Significant Word Register	0x0000_0000
CAM8M	0xB000_3048	R/W	CAM8 Most Significant Word Register	0x0000_0000
CAM8L	0xB000_304C	R/W	CAM8 Least Significant Word Register	0x0000_0000
CAM9M	0xB000_3050	R/W	CAM9 Most Significant Word Register	0x0000_0000
CAM9L	0xB000_3054	R/W	CAM9 Least Significant Word Register	0x0000_0000
CAM10M	0xB000_3058	R/W	CAM10 Most Significant Word Register	0x0000_0000
CAM10L	0xB000_305C	R/W	CAM10 Least Significant Word Register	0x0000_0000
CAM11M	0xB000_3060	R/W	CAM11 Most Significant Word Register	0x0000_0000
CAM11L	0xB000_3064	R/W	CAM11 Least Significant Word Register	0x0000_0000
CAM12M	0xB000_3068	R/W	CAM12 Most Significant Word Register	0x0000_0000
CAM12L	0xB000_306C	R/W	CAM12 Least Significant Word Register	0x0000_0000
CAM13M	0xB000_3070	R/W	CAM13 Most Significant Word Register	0x0000_0000
CAM13L	0xB000_3074	R/W	CAM13 Least Significant Word Register	0x0000_0000
CAM14M	0xB000_3078	R/W	CAM14 Most Significant Word Register	0x0000_0000
CAM14L	0xB000_307C	R/W	CAM14 Least Significant Word Register	0x0000_0000
CAM15M	0xB000_3080	R/W	CAM15 Most Significant Word Register	0x0000_0000
CAM15L	0xB000_3084	R/W	CAM15 Least Significant Word Register	0x0000_0000
TXDLSA	0xB000_3088	R/W	Transmit Descriptor Link List Start Address Register	0xFFFF_FFFC

RXDLSA	0xB000_308C	R/W	Receive Descriptor Link List Start Address Reg.	0xFFFF_FFFC
MCMDR	0xB000_3090	R/W	MAC Command Register	0x0000_0000
MIID	0xB000_3094	R/W	MII Management Data Register	0x0000_0000
MIIDA	0xB000_3098	R/W	MII Management Control and Address Register	0x0090_0000
FFTCR	0xB000_309C	R/W	FIFO Threshold Control Register	0x0000_0101
TSDR	0xB000_30A0	W	Transmit Start Demand Register	Undefined
RSDR	0xB000_30A4	W	Receive Start Demand Register	Undefined
DMARFC	0xB000_30A8	R/W	Maximum Receive Frame Control Register	0x0000_0800
MIEN	0xB000_30AC	R/W	MAC Interrupt Enable Register	0x0000_0000
Status Registers (11)				
MISTA	0xB000_30B0	R/W	MAC Interrupt Status Register	0x0000_0000
MGSTA	0xB000_30B4	R/W	MAC General Status Register	0x0000_0000
MPCNT	0xB000_30B8	R/W	Missed Packet Count Register	0x0000_7FFF
MRPC	0xB000_30BC	R	MAC Receive Pause Count Register	0x0000_0000
MRPCC	0xB000_30C0	R	MAC Receive Pause Current Count Register	0x0000_0000
MREPC	0xB000_30C4	R	MAC Remote Pause Count Register	0x0000_0000
DMARFS	0xB000_30C8	R/W	DMA Receive Frame Status Register	0x0000_0000
CTXDSA	0xB000_30CC	R	Current Transmit Descriptor Start Address Reg.	0x0000_0000
CTXBSA	0xB000_30D0	R	Current Transmit Buffer Start Address Register	0x0000_0000
CRXDSA	0xB000_30D4	R	Current Receive Descriptor Start Address Reg.	0x0000_0000
CRXBSA	0xB000_30D8	R	Current Receive Buffer Start Address Register	0x0000_0000

7.5.3 EMC Register Details

CAM Command Register (CAMCMR)

The EMC of NUC946ADN supports CAM function for destination MAC address recognition. The CAMCMR control the CAM comparison function, and unicast, multicast, and broadcast packet reception.

Register	Address	R/W	Description	Reset Value
CAMCMR	0xB000_3000	R/W	CAM Command Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved		RMII	ECMP	CCAM	ABP	AMP	AUP

Bits	Descriptions	
[5]	**RMII**	**Enable RMII Input Data Sampled by Negative Edge of REFCLK** 1'b0: PHY_CRSDV and PHY_RXD[1:0] are sampled by the positive edge of REFCLK 1'b1: PHY_CRSDV and PHY_RxD[1:0] are sampled by the negative edge of REFCLK
[4]	**ECMP**	**Enable CAM Compare** The ECMP controls the enable of CAM comparison function for destination MAC address recognition. If S/W wants to receive a packet with specific destination MAC address, configures the MAC address into anyone of 16 CAM entries, then enables that CAM entry and set ECMP to 1. 1'b0: Disable CAM comparison function for destination MAC address recognition. 1'b1: Enable CAM comparison function for destination MAC address recognition.
[3]	**CCAM**	**Complement CAM Compare** The CCAM controls the complement of the CAM comparison result. If the ECMP and CCAM are both enabled, the incoming packet with specific destination MAC address configured in CAM entry will be dropped. And the incoming packet with destination MAC address doesn't configured in any CAM entry will be received. 1'b0: The CAM comparison result doesn't be complemented. 1'b1: The CAM comparison result will be complemented.
[2]	**ABP**	**Accept Broadcast Packet** The ABP controls the broadcast packet reception. If ABP is enabled, EMC receives all incoming packet its destination MAC address is a broadcast address. 1'b0: EMC receives packet depends on the CAM comparison result. 1'b1: EMC receives all broadcast packets.

[1]	**AMP**	**Accept Multicast Packet** The AMP controls the multicast packet reception. If AMP is enabled, EMC receives all incoming packet its destination MAC address is a multicast address. 1'b0: EMC receives packet depends on the CAM comparison result. 1'b1: EMC receives all multicast packets.
[0]	**AUP**	**Accept Unicast Packet** The AUP controls the unicast packet reception. If AUP is enabled, EMC receives all incoming packet its destination MAC address is a unicast address. 1'b0: EMC receives packet depends on the CAM comparison result. 1'b1: EMC receives all unicast packets.

CAMCMR Setting and Comparison Result

The following table is the address recognition result in different CAMCMR configuration. The column Result shows the incoming packet type that can pass the address recognition in specific CAM configuration. The C, U, M and B represents the:

C: It indicates the destination MAC address of incoming packet has been configured in CAM entry.
U: It indicates the incoming packet is a unicast packet.
M: It indicates the incoming packet is a multicast packet.
B: It indicates the incoming packet is a broadcast packet.

ECMP	CCAM	AUP	AMP	ABP	Result
0	0	0	0	0	No Packet
0	0	0	0	1	B
0	0	0	1	0	M
0	0	0	1	1	M B
0	0	1	0	0	C U
0	0	1	0	1	C U B
0	0	1	1	0	C U M
0	0	1	1	1	C U M B
0	1	0	0	0	C U M B
0	1	0	0	1	C U M B
0	1	0	1	0	C U M B
0	1	0	1	1	C U M B
0	1	1	0	0	C U M B
0	1	1	0	1	C U M B
0	1	1	1	0	C U M B
0	1	1	1	1	C U M B
1	0	0	0	0	C
1	0	0	0	1	C B
1	0	0	1	0	C M
1	0	0	1	1	C N B
1	0	1	0	0	C U
1	0	1	0	1	C U B
1	0	1	1	0	C U M
1	0	1	1	1	C U M B
1	1	0	0	0	U M B
1	1	0	0	1	U M B
1	1	0	1	0	U M B
1	1	0	1	1	U M B
1	1	1	0	0	C U M B
1	1	1	0	1	C U M B
1	1	1	1	0	C U M B
1	1	1	1	1	C U M B

CAM Enable Register (CAMEN)

The CAMEN controls the validation of each CAM entry. Each CAM entry must be enabled first before it can participate in the destination MAC address recognition.

Register	Address	R/W	Description	Reset Value
CAMEN	0xB000_3004	R/W	CAM Enable Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
CAM15EN	CAM14EN	CAM13EN	CAM12EN	CAM11EN	CAM10EN	CAM9EN	CAM8EN
7	6	5	4	3	2	1	0
CAM7EN	CAM6EN	CAM5EN	CAM4EN	CAM3EN	CAM2EN	CAM1EN	CAM0EN

Bits	Descriptions	
[x]	**CAMxEN**	**CAM Entry x Enable** The CAMxEN controls the validation of CAM entry x. The x can be 0 to 15. The CAM entry 13, 14 and 15 are for PAUSE control frame transmission. If S/W wants to transmit a PAUSE control frame out to network, the enable bits of these three CAM entries all must be enabled first. 1'b0: CAM entry x is disabled. 1'b1: CAM entry x is enabled.

CAM Entry Registers (CAMxx)

In the EMC of NUC946ADN, there are 16 CAM entries. In these 16 CAM entries, 13 entries (entry 0~12) are to keep destination MAC address for packet recognition, and the other 3 entries (entry 13~15) are for PAUSE control frame transmission. Each CAM entry consists of 6 bytes. Consequently, 2 register ports are needed for each CAM entry.

For packet recognition, a register pair {CAMxM, CAMxL} represents a CAM entry and can keep a destination MAC address. The corresponding CAM enable bit CAMxEN of CAMEN register is also needed be enabled. The x can be the 0 to 12.

For PAUSE control frame transmission, first, S/W must configure destination MAC address of control frame into the register pair {CAM13M, CAM13L}, source MAC address into the register pair {CAM14M, CAM14L}, and configure length/type, op-code and operand of control frame into the register pair {CAM15M, CAM15L}. The bit CAM13EN, CAM14EN and CAM15EN of CAMEN register are also needed be enabled. Then, enable the bit SDPZ of MCMDR register.

Register	Address	R/W	Description	Reset Value
CAM0M	0xB000_3008	R/W	CAM0 Most Significant Word Register	0x0000_0000
CAM0L	0xB000_300C		CAM0 Least Significant Word Register	0x0000_0000
:	:		:	:
CAM15M	0xB000_3080		CAM15 Most Significant Word Register	0x0000_0000
CAM15L	0xB000_3084		CAM15 Least Significant Word Register	0x0000_0000

CAMxM

31	30	29	28	27	26	25	24
MAC Address Byte 5 (MSB)							
23	22	21	20	19	18	17	16
MAC Address Byte 4							
15	14	13	12	11	10	9	8
MAC Address Byte 3							
7	6	5	4	3	2	1	0
MAC Address Byte 2							

Bits	Descriptions	
[31:0]	**CAMxM**	**CAMx Most Significant Word** The CAMxM keeps the bit 47~16 of MAC address. The x can be the 0~14. The register pair {CAMxM, CAMxL} represents a CAM entry and can keep a MAC address. For example, if the MAC address 00-50-BA-33-BA-44 is kept in CAM entry 1, the register CAM1M is 32'h0050_BA33 and CAM1L is 32'hBA44_0000.

CAMxL

31	30	29	28	27	26	25	24
MAC Address Byte 1							
23	22	21	20	19	18	17	16
MAC Address Byte 0 (LSB)							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved							

Bits	Descriptions	
[31:0]	**CAMxL**	**CAMx Least Significant Word** The CAMxL keeps the bit 15~0 of MAC address. The x can be the 0~14. The register pair {CAMxM, CAMxL} represents a CAM entry and can keep a MAC address. For example, if the MAC address 00-50-BA-33-BA-44 is kept in CAM entry 1, the register CAM1M is 32'h0050_BA33 and CAM1L is 32'hBA44_0000.

CAM15M

31	30	29	28	27	26	25	24
Length/Type (MSB)							
23	22	21	20	19	18	17	16
Length/Type							
15	14	13	12	11	10	9	8
OP-Code (MSB)							
7	6	5	4	3	2	1	0
OP-Code							

Bits	Descriptions	
[31:16]	**Length/Type**	**Length/Type Field of PAUSE Control Frame** In the PAUSE control frame, a length/type field is defined and will be 16'h8808.
[15:0]	**OP-Code**	**OP Code Field of PAUSE Control Frame** In the PAUSE control frame, an op code field is defined and will be 16'h0001.

CAM15L

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

Bits	Descriptions	
[31:16]	**Operand**	**Pause Parameter** In the PAUSE control frame, an operand field is defined and controls how much time the destination Ethernet MAC Controller is paused. The unit of the operand is the slot time, the 512 bits time.

Transmit Descriptor Link List Start Address Register (TXDLSA)

The Tx descriptor defined in EMC is a link-list data structure. The TXDLSA keeps the starting address of this link-list. In other words, the TXDLSA keeps the starting address of the 1st Tx descriptor. S/W must configure TXDLSA before enable bit TXON of MCMDR register.

Register	Address	R/W	Description	Reset Value
TXDLSA	0xB000_3088	R/W	Transmit Descriptor Link List Start Address Register	0xFFFF_FFFC

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

Bits	Descriptions	
[31:0]	**TXDLSA**	**Transmit Descriptor Link-List Start Address** The TXDLSA keeps the start address of transmit descriptor link-list. If the S/W enables the bit TXON of MCMDR register, the content of TXDLSA will be loaded into the current transmit descriptor start address register (CTXDSA). The TXDLSA doesn't be updated by EMC. During the operation, EMC will ignore the bits [1:0] of TXDLSA. This means that each Tx descriptor always must locate at word boundary memory address.

Receive Descriptor Link List Start Address Register (RXDLSA)

The Rx descriptor defined in EMC is a link-list data structure. The RXDLSA keeps the starting address of this link-list. In other words, the RXDLSA keeps the starting address of the 1st Rx descriptor. S/W must configure RXDLSA before enable bit RXON of MCMDR register.

Register	Address	R/W	Description	Reset Value
RXDLSA	0xB000_308C	R/W	Receive Descriptor Link List Start Address Register	0xFFFF_FFFC

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

Bits	Descriptions	
[31:0]	**RTXDLSA**	**Receive Descriptor Link-List Start Address** The RXDLSA keeps the start address of receive descriptor link-list. If the S/W enables the bit RXON of MCMDR register, the content of RXDLSA will be loaded into the current receive descriptor start address register (CRXDSA). The RXDLSA doesn't be updated by EMC. During the operation, EMC will ignore the bits [1:0] of RXDLSA. This means that each Rx descriptor always must locate at word boundary memory address.

MAC Command Register (MCMDR)

The MCMDR provides the control information for EMC. Some command settings affect both frame transmission and reception, such as bit FDUP, the full/half duplex mode selection, or bit OPMOD, the 100/10M bps mode selection. Some command settings control frame transmission and reception separately, likes bit TXON and RXON.

Register	Address	R/W	Description	Reset Value
MCMDR	0xB000_3090	R/W	MAC Command Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							SWR
23	22	21	20	19	18	17	16
Reserved		LBK	OPMOD	EnMDC	FDUP	EnSQE	SDPZ
15	14	13	12	11	10	9	8
Reserved						NDEF	TXON
7	6	5	4	3	2	1	0
Reserved		SPCRC	AEP	ACP	ARP	ALP	RXON

Bits	Descriptions	
[24]	**SWR**	**Software Reset** The SWR implements a reset function to make the EMC return default state. The SWR is a self-clear bit. This means after the software reset finished, the SWR will be cleared automatically. Enable SWR can also reset all control and status registers, exclusive of these two bits EnRMII and OPMOD of MCMDR register. The EMC re-initial is needed after the software reset completed. 1'b0: Software reset completed. 1'b1: Enable software reset.
[21]	**LBK**	**Internal Loop Back Select** The LBK enables the EMC operating on internal loop-back mode. If the LBK is enabled, the packet transmitted out will be loop-backed to Rx. If the EMC is operating on internal loop-back mode, it also means the EMC is operating on full-duplex mode and the value of FDUP of MCMDR register is ignored. Beside, the LBK doesn't be affected by SWR bit. 1'b0: The EMC operates in normal mode. 1'b1: The EMC operates in internal loop-back mode.
[20]	**OPMOD**	**Operation Mode Select** The OPMOD defines the EMC is operating on 10M or 100M bps mode. The OPMOD doesn't be affected by SWR bit. 1'b0: The EMC operates on 10Mbps mode. 1'b1: The EMC operates on 100Mbps mode.

[19]	**EnMDC**	**Enable MDC Clock Generation** The EnMDC controls the MDC clock generation for MII Management Interface. If the EnMDC is set to 1, the MDC clock generation is enabled. Otherwise, the MDC clock generation is disabled. Consequently, if S/W wants to access the registers of external PHY through MII Management Interface, the EnMDC must be set to high. 1'b0: Disable MDC clock generation. 1'b1: Enable MDC clock generation.
[18]	**FDUP**	**Full Duplex Mode Select** The FDUP controls that EMC is operating on full or half duplex mode. 1'b0: The EMC operates on half duplex mode. 1'b1: The EMC operates on full duplex mode.
[17]	**EnSQE**	**Enable SQE Checking** The EnSQE controls the enable of SQE checking. The SQE checking is only available while EMC is operating on 10M bps and half duplex mode. In other words, the EnSQE cannot affect EMC operation, if the EMC is operating on 100M bps or full duplex mode. 1'b0: Disable SQE checking while EMC is operating on 10Mbps and half duplex mode. 1'b1: Enable SQE checking while EMC is operating on 10Mbps and half duplex mode.
[16]	**SDPZ**	**Send PAUSE Frame** The SDPZ controls the PAUSE control frame transmission. If S/W wants to send a PAUSE control frame out, the CAM entry 13, 14 and 15 must be configured first and the corresponding CAM enable bit of CAMEN register also must be set. Then, set SDPZ to 1 enables the PAUSE control frame transmission. The SDPZ is a self-clear bit. This means after the PAUSE control frame transmission has completed, the SDPZ will be cleared automatically. It is recommended that only enables SPDZ while EMC is operating on full duplex mode. 1'b0: The PAUSE control frame transmission has completed. 1'b1: Enable EMC to transmit a PAUSE control frame out.
[9]	**NDEF**	**No Defer** The NDEF controls the enable of deferral exceed counter. If NDEF is set to high, the deferral exceed counter is disabled. The NDEF is only useful while EMC is operating on half duplex mode. 1'b0: The deferral exceed counter is enabled. 1'b1: The deferral exceed counter is disabled.

[8]	**TXON**	**Frame Transmission ON** The TXON controls the normal packet transmission of EMC. If the TXON is set to high, the EMC starts the packet transmission process, including the Tx descriptor fetching, packet transmission and Tx descriptor modification. It is must to finish EMC initial sequence before enable TXON. Otherwise, the EMC operation is undefined. If the TXON is disabled during EMC is transmitting a packet out, the EMC stops the packet transmission process after the current packet transmission finished. 1'b0: The EMC stops packet transmission process. 1'b1: The EMC starts packet transmission process.
[5]	**SPCRC**	**Strip CRC Checksum** The SPCRC controls if the length of incoming packet is calculated with 4 bytes CRC checksum. If the SPCRC is set to high, 4 bytes CRC checksum is excluded from length calculation of incoming packet. 1'b0: The 4 bytes CRC checksum is included in packet length calculation. 1'b1: The 4 bytes CRC checksum is excluded in packet length calculation.
[4]	**AEP**	**Accept CRC Error Packet** The AEP controls the EMC accepts or drops the CRC error packet. If the AEP is set to high, the incoming packet with CRC error will be received by EMC as a good packet. 1'b0: The CRC error packet will be dropped by EMC. 1'b1: The CRC error packet will be accepted by EMC.
[3]	**ACP**	**Accept Control Packet** The ACP controls the control frame reception. If the ACP is set to high, the EMC will accept the control frame. Otherwise, the control frame will be dropped. It is recommended that S/W only enable AEP while EMC is operating on full duplex mode. 1'b0: The control frame will be dropped by EMC. 1'b1: The control frame will be accepted by EMC.
[2]	**ARP**	**Accept Runt Packet** The ARP controls the runt packet, which length is less than 64 bytes, reception. If the ARP is set to high, the EMC will accept the runt packet. Otherwise, the runt packet will be dropped. 1'b0: The runt packet will be dropped by EMC. 1'b1: The runt packet will be accepted by EMC.
[1]	**ALP**	**Accept Long Packet** The ALP controls the long packet, which packet length is greater than 1518 bytes, reception. If the ALP is set to high, the EMC will accept the long packet. Otherwise, the long packet will be dropped. 1'b0: The long packet will be dropped by EMC. 1'b1: The long packet will be accepted by EMC.

[0]	**RXON**	**Frame Reception ON** The RXON controls the normal packet reception of EMC. If the RXON is set to high, the EMC starts the packet reception process, including the Rx descriptor fetching, packet reception and Rx descriptor modification. It is must to finish EMC initial sequence before enable RXON. Otherwise, the EMC operation is undefined. If the RXON is disabled during EMC is receiving an incoming packet, the EMC stops the packet reception process after the current packet reception finished. 1'b0: The EMC stops packet reception process. 1'b1: The EMC starts packet reception process.

MII Management Data Register (MIID)

The EMC provides MII management function to access the control and status registers of the external PHY. The MIID register is used to store the data that will be written into the registers of external PHY for write command or the data that is read from the registers of external PHY for read command.

Register	Address	R/W	Description	Reset Value
MIID	0xB000_3094	R/W	MII Management Data Register	0x0000_0000

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

Bits	Descriptions	
[15:0]	**MIIData**	**MII Management Data** The MIIData is the 16 bits data that will be written into the registers of external PHY for MII Management write command or the data from the registers of external PHY for MII Management read command.

MII Management Control and Address Register (MIIDA)

The EMC provides MII management function to access the control and status registers of the external PHY. The MIIDA register is used to keep the MII management command information, like the register address, external PHY address, MDC clocking rate, read/write etc.

Register	Address	R/W	Description	Reset Value
MIIDA	0xB000_3098	R/W	MII Management Control and Address Register	0x0090_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
MDCCR				MDCON	PreSP	BUSY	Write
15	14	13	12	11	10	9	8
Reserved			PHYAD				
7	6	5	4	3	2	1	0
Reserved			PHYRAD				

Bits	Descriptions	
[23:20]	**MDCCR**	**MDC Clock Rating (Default Value: 4'h9)** The MDCCR controls the MDC clock rating for MII Management I/F. Depend on the IEEE Std. 802.3 clause 22.2.2.11, the minimum period for MDC shall be 400ns. In other words, the maximum frequency for MDC is 2.5MHz. The MDC is divided from the AHB bus clock, the HCLK. Consequently, for different HCLKs the different ratios are required to generate appropriate MDC clock. The following table shows relationship between HCLK and MDC clock in different MDCCR configurations. The T_{HCLK} indicates the period of HCLK.

MDCCR [23:20]	MDC Clock Period	MDC Clock Frequency
4'b0000	4 x T_{HCLK}	HCLK/4
4'b0001	6 x T_{HCLK}	HCLK/6
4'b0010	8 x T_{HCLK}	HCLK/8
4'b0011	12 x T_{HCLK}	HCLK/12
4'b0100	16 x T_{HCLK}	HCLK/16
4'b0101	20 x T_{HCLK}	HCLK/20
4'b0110	24 x T_{HCLK}	HCLK/24
4'b0111	28 x T_{HCLK}	HCLK/28
4'b1000	30 x T_{HCLK}	HCLK/30
4'b1001	32 x T_{HCLK}	HCLK/32
4'b1010	36 x T_{HCLK}	HCLK/36
4'b1011	40 x T_{HCLK}	HCLK/40
4'b1100	44 x T_{HCLK}	HCLK/44
4'b1101	48 x T_{HCLK}	HCLK/48
4'b1110	54 x T_{HCLK}	HCLK/54
4'b1111	60 x T_{HCLK}	HCLK/60

[19]	**MDCON**	**MDC Clock ON Always** The MDC controls the MDC clock generation. If the MDCON is set to high, the MDC clock actives always. Otherwise, the MDC will only active while S/W issues a MII management command. 1'b0: The MDC clock will only active while S/W issues a MII management command. 1'b1: The MDC clock actives always.
[18]	**PreSP**	**Preamble Suppress** The PreSP controls the preamble field generation of MII management frame. If the PreSP is set to high, the preamble field generation of MII management frame is skipped. 1'b0: Preamble field generation of MII management frame is not skipped. 1'b1: Preamble field generation of MII management frame is skipped.
[17]	**BUSY**	**Busy Bit** The BUSY controls the enable of the MII management frame generation. If S/W wants to access registers of external PHY, it set BUSY to high and EMC generates the MII management frame to external PHY through MII Management I/F. The BUSY is a self-clear bit. This means the BUSY will be cleared automatically after the MII management command finished. 1'b0: The MII management has finished. 1'b1: Enable EMC to generate a MII management command to external PHY.
[16]	**Write**	**Write Command** The Write defines the MII management command is a read or write. 1'b0: The MII management command is a read command. 1'b1: The MII management command is a write command.
[12:8]	**PHYAD**	**PHY Address** The PHYAD keeps the address to differentiate which external PHY is the target of the MII management command.
[4:0]	**PHYRAD**	**PHY Register Address** The PHYRAD keeps the address to indicate which register of external PHY is the target of the MII management command.

MII Management Function Frame Format

In IEEE Std. 802.3 clause 22.2.4, the MII management function is defined. The MII management function is used for the purpose of controlling the PHY and gathering status from the PHY. The MII management frame format is shown as follow.

	Management frame fields							
	PRE	ST	OP	PHYAD	REGAD	TA	DATA	IDLE
READ	1…1	01	10	AAAAA	RRRRR	Z0	DDDDDDDDDDDDDDDD	Z
WRITE	1…1	01	01	AAAAA	RRRRR	10	DDDDDDDDDDDDDDDD	Z

MII Management Function Configure Sequence

Read	Write
1. Set appropriate MDCCR. 2. Set PHYAD and PHYRAD. 3. Set Write to 1'b0 4. Set bit BUSY to 1'b1 to send a MII management frame out. 5. Wait BUSY to become 1'b0. 6. Read data from MIID register. 7. Finish the read command.	1. Write data to MIID register 2. Set appropriate MDCCR. 3. Set PHYAD and PHYRAD. 4. Set Write to 1'b1 5. Set bit BUSY to 1'b1 to send a MII management frame out. 6. Wait BUSY to become 1'b0. 7. Finish the write command.

FIFO Threshold Control Register (FFTCR)

The FFTCR defines the high and low threshold of internal FIFOs, including TxFIFO and RxFIFO. The threshold of internal FIFOs is related to EMC request generation and when the frame transmission starts. The FFTCR also defines the burst length of AHB bus cycle for system memory access.

Register	Address	R/W	Description	Reset Value
FFTCR	0xB000_309C	R/W	FIFO Threshold Control Register	0x0000_0101

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

Bits	Descriptions	
[21:20]	**Blength**	**DMA Burst Length** The Blength defines the burst length of AHB bus cycle while EMC accesses system memory. 2'b00: 4 words 2'b01: 8 words 2'b10: 16 words 2'b11: 16 words
[9:8]	**TxTHD**	**TxFIFO Low Threshold** Default Value: 2'b01 The TxTHD controls when TxDMA requests internal arbiter for data transfer between system memory and TxFIFO. The TxTHD defines not only the low threshold of TxFIFO, but also the high threshold. The high threshold is the twice of low threshold always. During the packet transmission, if the TxFIFO reaches the high threshold, the TxDMA stops generate request to transfer frame data from system memory to TxFIFO. If the frame data in TxFIFO is less than low threshold, TxDMA starts to transfer frame data from system memory to TxFIFO. The TxTHD also defines when the TxMAC starts to transmit frame out to network. The TxMAC starts to transmit the frame out while the TxFIFO first time reaches the high threshold during the transmission of the frame. If the frame data length is less than TxFIFO high threshold, the TxMAC starts to transmit the frame out after the frame data are all inside the TxFIFO. 2'b00: Undefined. 2'b01: TxFIFO low threshold is 64B and high threshold is 128B. 2'b10: TxFIFO low threshold is 80B and high threshold is 160B. 2'b11: TxFIFO low threshold is 96B and high threshold is 192B.

[1:0]	**RxTHD**	**RxFIFO High Threshold** Default Value: 2'b01 The RxTHD controls when RxDMA requests internal arbiter for data transfer between RxFIFO and system memory. The RxTHD defines not only the high threshold of RxFIFO, but also the low threshold. The low threshold is the half of high threshold always. During the packet reception, if the RxFIFO reaches the high threshold, the RxDMA starts to transfer frame data from RxFIFO to system memory. If the frame data in RxFIFO is less than low threshold, RxDMA stops to transfer the frame data to system memory. 2'b00: Depend on the burst length setting. If the burst length is 8 words, high threshold is 8 words, too. 2'b01: RxFIFO high threshold is 64B and low threshold is 32B. 2'b10: RxFIFO high threshold is 128B and low threshold is 64B. 2'b11: RxFIFO high threshold is 192B and low threshold is 96B.

Transmit Start Demand Register (TSDR)

If the Tx descriptor is not available for use of TxDMA after the TXON of MCMDR register is enabled, the FSM (Finite State Machine) of TxDMA enters the Halt state and the frame transmission is halted. After the S/W has prepared the new Tx descriptor for frame transmission, it must issue a write command to TSDR register to make TxDMA leave Halt state and contiguous frame transmission. The TSDR is a write only register and read from this register is undefined. The write to TSDR register has took effect only while TxDMA stayed at Halt state.

Register	Address	R/W	Description	Reset Value
TSDR	0xB000_30A0	W	Transmit Start Demand Register	Undefined

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

Bits	Descriptions	
[31:0]	**TSD**	**Transmit Start Demand**

Receive Start Demand Register (RSDR)

If the Rx descriptor is not available for use of RxDMA after the RXON of MCMDR register is enabled, the FSM (Finite State Machine) of RxDMA enters the Halt state and the frame reception is halted. After the S/W has prepared the new Rx descriptor for frame reception, it must issue a write command to RSDR register to make RxDMA leave Halt state and contiguous frame reception. The RSDR is a write only register and read from this register is undefined. The write to RSDR register has took effect only while RxDMA stayed at Halt state.

Register	Address	R/W	Description	Reset Value
RSDR	0xB000_30A4	W	Receive Start Demand Register	Undefined

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

Bits	Descriptions	
[31:0]	**RSD**	**Receive Start Demand**

Maximum Receive Frame Control Register (DMARFC)

The DMARFC defines the maximum frame length for a received frame that can be stored in the system memory. It is recommend that only use this register while S/W wants to receive a frame which length is greater than 1518 bytes.

Register	Address	R/W	Description	Reset Value
DMARFC	0xB000_30A8	R/W	Maximum Receive Frame Control Register	0x0000_0800

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

Bits	Descriptions	
[15:0]	**RXMS**	**Maximum Receive Frame Length** Default Value: 16'h0800 The RXMS defines the maximum frame length for received frame. If the frame length of received frame is greater than RXMS, and bit EnDFO of MIEN register is also enabled, the bit DFOI of MISTA register is set and the Rx interrupt is triggered. It is recommended that only use RXMS to qualify the length of received frame while S/W wants to receive a frame which length is greater than 1518 bytes.

MAC Interrupt Enable Register (MIEN)

The MIEN controls the enable of EMC interrupt status to generate interrupt. Two interrupts, RXINTR for frame reception and TXINTR for frame transmission, are generated from EMC to CPU.

Register	Address	R/W	Description	Reset Value
MIEN	0xB000_30AC	R/W	MAC Interrupt Enable Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							EnTxBErr
23	22	21	20	19	18	17	16
EnTDU	EnLC	EnTXABT	EnNCS	EnEXDEF	EnTXCP	EnTXEMP	EnTXINTR
15	14	13	12	11	10	9	8
Reserved	EnCFR	Reserved		EnRxBErr	EnRDU	EnDEN	EnDFO
7	6	5	4	3	2	1	0
EnMMP	EnRP	EnALIE	EnRXGD	EnPTLE	EnRXOV	EnCRCE	EnRXINTR

Bits	Descriptions	
[24]	**EnTxBErr**	**Enable Transmit Bus Error Interrupt** The EnTxBErr controls the TxBErr interrupt generation. If TxBErr of MISTA register is set, and both EnTxBErr and EnTXINTR are enabled, the EMC generates the Tx interrupt to CPU. If EnTxBErr or EnTXINTR is disabled, no Tx interrupt is generated to CPU even the TxBErr of MISTA register is set. 1'b0: TxBErr of MISTA register is masked from Tx interrupt generation. 1'b1: TxBErr of MISTA register can participate in Tx interrupt generation.
[23]	**EnTDU**	**Enable Transmit Descriptor Unavailable Interrupt** The EnTDU controls the TDU interrupt generation. If TDU of MISTA register is set, and both EnTDU and EnTXINTR are enabled, the EMC generates the Tx interrupt to CPU. If EnTDU or EnTXINTR is disabled, no Tx interrupt is generated to CPU even the TDU of MISTA register is set. 1'b0: TDU of MISTA register is masked from Tx interrupt generation. 1'b1: TDU of MISTA register can participate in Tx interrupt generation.
[22]	**EnLC**	**Enable Late Collision Interrupt** The EnLC controls the LC interrupt generation. If LC of MISTA register is set, and both EnLC and EnTXINTR are enabled, the EMC generates the Tx interrupt to CPU. If EnLC or EnTXINTR is disabled, no Tx interrupt is generated to CPU even the LC of MISTA register is set. 1'b0: LC of MISTA register is masked from Tx interrupt generation. 1'b1: LC of MISTA register can participate in Tx interrupt generation.

[21]	**EnTXABT**	**Enable Transmit Abort Interrupt** The EnTXABT controls the TXABT interrupt generation. If TXABT of MISTA register is set, and both EnTXABT and EnTXINTR are enabled, the EMC generates the Tx interrupt to CPU. If EnTXABT or EnTXINTR is disabled, no Tx interrupt is generated to CPU even the TXABT of MISTA register is set. 1'b0: TXABT of MISTA register is masked from Tx interrupt generation. 1'b1: TXABT of MISTA register can participate in Tx interrupt generation.
[20]	**EnNCS**	**Enable No Carrier Sense Interrupt** The EnNCS controls the NCS interrupt generation. If NCS of MISTA register is set, and both EnNCS and EnTXINTR are enabled, the EMC generates the Tx interrupt to CPU. If EnNCS or EnTXINTR is disabled, no Tx interrupt is generated to CPU even the NCS of MISTA register is set. 1'b0: NCS of MISTA register is masked from Tx interrupt generation. 1'b1: NCS of MISTA register can participate in Tx interrupt generation.
[19]	**EnEXDEF**	**Enable Defer Exceed Interrupt** The EnEXDEF controls the EXDEF interrupt generation. If EXDEF of MISTA register is set, and both EnEXDEF and EnTXINTR are enabled, the EMC generates the Tx interrupt to CPU. If EnEXDEF or EnTXINTR is disabled, no Tx interrupt is generated to CPU even the EXDEF of MISTA register is set. 1'b0: EXDEF of MISTA register is masked from Tx interrupt generation. 1'b1: EXDEF of MISTA register can participate in Tx interrupt generation.
[18]	**EnTXCP**	**Enable Transmit Completion Interrupt** The EnTXCP controls the TXCP interrupt generation. If TXCP of MISTA register is set, and both EnTXCP and EnTXINTR are enabled, the EMC generates the Tx interrupt to CPU. If EnTXCP or EnTXINTR is disabled, no Tx interrupt is generated to CPU even the TXCP of MISTA register is set. 1'b0: TXCP of MISTA register is masked from Tx interrupt generation. 1'b1: TXCP of MISTA register can participate in Tx interrupt generation.
[17]	**EnTXEMP**	**Enable Transmit FIFO Underflow Interrupt** The EnTXEMP controls the TXEMP interrupt generation. If TXEMP of MISTA register is set, and both EnTXEMP and EnTXINTR are enabled, the EMC generates the Tx interrupt to CPU. If EnTXEMP or EnTXINTR is disabled, no Tx interrupt is generated to CPU even the TXEMP of MISTA register is set. 1'b0: TXEMP of MISTA register is masked from Tx interrupt generation. 1'b1: TXEMP of MISTA register can participate in Tx interrupt generation.

[16]	**EnTXINTR**	**Enable Transmit Interrupt** The EnTXINTR controls the Tx interrupt generation. If EnTXINTR is enabled and TXINTR of MISTA register is high, EMC generates the Tx interrupt to CPU. If EnTXINTR is disabled, no Tx interrupt is generated to CPU even the status bits 17~24 of MISTA are set and the corresponding bits of MIEN are enabled. In other words, if S/W wants to receive Tx interrupt from EMC, this bit must be enabled. And, if S/W doesn't want to receive any Tx interrupt from EMC, disables this bit. 1'b0: TXINTR of MISTA register is masked and Tx interrupt generation is disabled. 1'b1: TXINTR of MISTA register is unmasked and Tx interrupt generation is enabled.
[14]	**EnCFR**	**Enable Control Frame Receive Interrupt** The EnCFR controls the CFR interrupt generation. If CFR of MISTA register is set, and both EnCFR and EnTXINTR are enabled, the EMC generates the Rx interrupt to CPU. If EnCFR or EnTXINTR is disabled, no Rx interrupt is generated to CPU even the CFR of MISTA register is set. 1'b0: CFR of MISTA register is masked from Rx interrupt generation. 1'b1: CFR of MISTA register can participate in Rx interrupt generation.
[11]	**EnRxBErr**	**Enable Receive Bus Error Interrupt** The EnRxBErr controls the RxBerr interrupt generation. If RxBErr of MISTA register is set, and both EnRxBErr and EnTXINTR are enabled, the EMC generates the Rx interrupt to CPU. If EnRxBErr or EnTXINTR is disabled, no Rx interrupt is generated to CPU even the RxBErr of MISTA register is set. 1'b0: RxBErr of MISTA register is masked from Rx interrupt generation. 1'b1: RxBErr of MISTA register can participate in Rx interrupt generation.
[10]	**EnRDU**	**Enable Receive Descriptor Unavailable Interrupt** The EnRDU controls the RDU interrupt generation. If RDU of MISTA register is set, and both EnRDU and EnTXINTR are enabled, the EMC generates the Rx interrupt to CPU. If EnRDU or EnTXINTR is disabled, no Rx interrupt is generated to CPU even the RDU of MISTA register is set. 1'b0: RDU of MISTA register is masked from Rx interrupt generation. 1'b1: RDU of MISTA register can participate in Rx interrupt generation.
[9]	**EnDEN**	**Enable DMA Early Notification Interrupt** The EnDEN controls the DENI interrupt generation. If DENI of MISTA register is set, and both EnDEN and EnTXINTR are enabled, the EMC generates the Rx interrupt to CPU. If EnDEN or EnTXINTR is disabled, no Rx interrupt is generated to CPU even the DENI of MISTA register is set. 1'b0: DENI of MISTA register is masked from Rx interrupt generation. 1'b1: DENI of MISTA register can participate in Rx interrupt generation.

[8]	**EnDFO**	**Enable Maximum Frame Length Interrupt** The EnDFO controls the DFOI interrupt generation. If DFOI of MISTA register is set, and both EnDFO and EnTXINTR are enabled, the EMC generates the Rx interrupt to CPU. If EnDFO or EnTXINTR is disabled, no Rx interrupt is generated to CPU even the DFOI of MISTA register is set. 1'b0: DFOI of MISTA register is masked from Rx interrupt generation. 1'b1: DFOI of MISTA register can participate in Rx interrupt generation.
[7]	**EnMMP**	**Enable More Missed Packet Interrupt** The EnMMP controls the MMP interrupt generation. If MMP of MISTA register is set, and both EnMMP and EnTXINTR are enabled, the EMC generates the Rx interrupt to CPU. If EnMMP or EnTXINTR is disabled, no Rx interrupt is generated to CPU even the MMP of MISTA register is set. 1'b0: MMP of MISTA register is masked from Rx interrupt generation. 1'b1: MMP of MISTA register can participate in Rx interrupt generation.
[6]	**EnRP**	**Enable Runt Packet Interrupt** The EnRP controls the RP interrupt generation. If RP of MISTA register is set, and both EnRP and EnTXINTR are enabled, the EMC generates the Rx interrupt to CPU. If EnRP or EnTXINTR is disabled, no Rx interrupt is generated to CPU even the RP of MISTA register is set. 1'b0: RP of MISTA register is masked from Rx interrupt generation. 1'b1: RP of MISTA register can participate in Rx interrupt generation.
[5]	**EnALIE**	**Enable Alignment Error Interrupt** The EnALIE controls the ALIE interrupt generation. If ALIE of MISTA register is set, and both EnALIE and EnTXINTR are enabled, the EMC generates the Rx interrupt to CPU. If EnALIE or EnTXINTR is disabled, no Rx interrupt is generated to CPU even the ALIE of MISTA register is set. 1'b0: ALIE of MISTA register is masked from Rx interrupt generation. 1'b1: ALIE of MISTA register can participate in Rx interrupt generation.
[4]	**EnRXGD**	**Enable Receive Good Interrupt** The EnRXGD controls the RXGD interrupt generation. If RXGD of MISTA register is set, and both EnRXGD and EnTXINTR are enabled, the EMC generates the Rx interrupt to CPU. If EnRXGD or EnTXINTR is disabled, no Rx interrupt is generated to CPU even the RXGD of MISTA register is set. 1'b0: RXGD of MISTA register is masked from Rx interrupt generation. 1'b1: RXGD of MISTA register can participate in Rx interrupt generation.
[3]	**EnPTLE**	**Enable Packet Too Long Interrupt** The EnPTLE controls the PTLE interrupt generation. If PTLE of MISTA register is set, and both EnPTLE and EnTXINTR are enabled, the EMC generates the Rx interrupt to CPU. If EnPTLE or EnTXINTR is disabled, no Rx interrupt is generated to CPU even the PTLE of MISTA register is set. 1'b0: PTLE of MISTA register is masked from Rx interrupt generation. 1'b1: PTLE of MISTA register can participate in Rx interrupt generation.

[2]	**EnRXOV**	**Enable Receive FIFO Overflow Interrupt** The EnRXOV controls the RXOV interrupt generation. If RXOV of MISTA register is set, and both EnRXOV and EnTXINTR are enabled, the EMC generates the Rx interrupt to CPU. If EnRXOV or EnTXINTR is disabled, no Rx interrupt is generated to CPU even the RXOV of MISTA register is set. 1'b0: RXOV of MISTA register is masked from Rx interrupt generation. 1'b1: RXOV of MISTA register can participate in Rx interrupt generation.
[1]	**EnCRCE**	**Enable CRC Error Interrupt** The EnCRCE controls the CRCE interrupt generation. If CRCE of MISTA register is set, and both EnCRCE and EnTXINTR are enabled, the EMC generates the Rx interrupt to CPU. If EnCRCE or EnTXINTR is disabled, no Rx interrupt is generated to CPU even the CRCE of MISTA register is set. 1'b0: CRCE of MISTA register is masked from Rx interrupt generation. 1'b1: CRCE of MISTA register can participate in Rx interrupt generation.
[0]	**EnRXINTR**	**Enable Receive Interrupt** The EnRXINTR controls the Rx interrupt generation. If EnRXINTR is enabled and RXINTR of MISTA register is high, EMC generates the Rx interrupt to CPU. If EnRXINTR is disabled, no Rx interrupt is generated to CPU even the status bits 1~14 of MISTA are set and the corresponding bits of MIEN are enabled. In other words, if S/W wants to receive Rx interrupt from EMC, this bit must be enabled. And, if S/W doesn't want to receive any Rx interrupt from EMC, disables this bit. 1'b0: RXINTR of MISTA register is masked and Rx interrupt generation is disabled. 1'b1: RXINTR of MISTA register is unmasked and Rx interrupt generation is enabled.

MAC Interrupt Status Register (MISTA)

The MISTA keeps much EMC statuses, like frame transmission and reception status, internal FIFO status and also NATA processing status. The statuses kept in MISTA will trigger the reception or transmission interrupt. The MISTA is a write clear register and write 1 to corresponding bit clears the status and also clears the interrupt.

Register	Address	R/W	Description	Reset Value
MISTA	0xB000_30B0	R/W	MAC Interrupt Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							TxBErr
23	22	21	20	19	18	17	16
TDU	LC	TXABT	NCS	EXDEF	TXCP	TXEMP	TXINTR
15	14	13	12	11	10	9	8
Reserved	CFR	Reserved		RxBErr	RDU	DENI	DFOI
7	6	5	4	3	2	1	0
MMP	RP	ALIE	RXGD	PTLE	RXOV	CRCE	RXINTR

Bits	Descriptions	
[24]	**TxBErr**	**Transmit Bus Error Interrupt** The TxBErr high indicates the memory controller replies ERROR response while EMC access system memory through TxDMA during packet transmission process. Reset EMC is recommended while TxBErr status is high. If the TxBErr is high and EnTxBErr of MIEN register is enabled, the TxINTR will be high. Write 1 to this bit clears the TxBErr status. 1'b0: No ERROR response is received. 1'b1: ERROR response is received.
[23]	**TDU**	**Transmit Descriptor Unavailable Interrupt** The TDU high indicates that there is no available Tx descriptor for packet transmission and TxDMA will stay at Halt state. Once, the TxDMA enters the Halt state, S/W must issues a write command to TSDR register to make TxDMA leave Halt state while new Tx descriptor is available. If the TDU is high and EnTDU of MIEN register is enabled, the TxINTR will be high. Write 1 to this bit clears the TDU status. 1'b0: Tx descriptor is available. 1'b1: Tx descriptor is unavailable.

[22]	**LC**	**Late Collision Interrupt** The LC high indicates the collision occurred in the outside of 64 bytes collision window. This means after the 64 bytes of a frame has transmitted out to the network, the collision still occurred. The late collision check will only be done while EMC is operating on half-duplex mode. If the LC is high and EnLC of MIEN register is enabled, the TxINTR will be high. Write 1 to this bit clears the LC status. 1'b0: No collision occurred in the outside of 64 bytes collision window. 1'b1: Collision occurred in the outside of 64 bytes collision window.
[21]	**TXABT**	**Transmit Abort Interrupt** The TXABT high indicates the packet incurred 16 consecutive collisions during transmission, and then the transmission process for this packet is aborted. The transmission abort is only available while EMC is operating on half-duplex mode. If the TXABT is high and EnTXABT of MIEN register is enabled, the TxINTR will be high. Write 1 to this bit clears the TXABT status. 1'b0: Packet doesn't incur 16 consecutive collisions during transmission. 1'b1: Packet incurred 16 consecutive collisions during transmission.
[20]	**NCS**	**No Carrier Sense Interrupt** The NCS high indicates the MII I/F signal CRS doesn't active at the start of or during the packet transmission. The NCS is only available while EMC is operating on half-duplex mode. If the NCS is high and EnNCS of MIEN register is enabled, the TxINTR will be high. Write 1 to this bit clears the NCS status. 1'b0: CRS signal actives correctly. 1'b1: CRS signal doesn't active at the start of or during the packet transmission.
[19]	**EXDEF**	**Defer Exceed Interrupt** The EXDEF high indicates the frame waiting for transmission has deferred over 0.32768ms on 100Mbps mode, or 3.2768ms on 10Mbps mode. The deferral exceed check will only be done while bit NDEF of MCMDR is disabled, and EMC is operating on half-duplex mode. If the EXDEF is high and EnEXDEF of MIEN register is enabled, the TxINTR will be high. Write 1 to this bit clears the EXDEF status. 1'b0: Frame waiting for transmission has not deferred over 0.32768ms (100Mbps) or 3.2768ms (10Mbps). 1'b1: Frame waiting for transmission has deferred over 0.32768ms (100Mbps) or 3.2768ms (10Mbps).
[18]	**TXCP**	**Transmit Completion Interrupt** The TXCP indicates the packet transmission has completed correctly. If the TXCP is high and EnTXCP of MIEN register is enabled, the TxINTR will be high. Write 1 to this bit clears the TXCP status. 1'b0: The packet transmission doesn't complete. 1'b1: The packet transmission has completed.

[17]	**TXEMP**	**Transmit FIFO Underflow Interrupt** The TXEMP high indicates the TxFIFO underflow occurred during packet transmission. While the TxFIFO underflow occurred, the EMC will retransmit the packet automatically without S/W intervention. If the TxFIFO underflow occurred often, it is recommended that modify TxFIFO threshold control, the TxTHD of FFTCR register, to higher level. If the TXEMP is high and EnTXEMP of MIEN register is enabled, the TxINTR will be high. Write 1 to this bit clears the TXEMP status. 1'b0: No TxFIFO underflow occurred during packet transmission. 1'b0: TxFIFO underflow occurred during packet transmission.
[16]	**TXINTR**	**Transmit Interrupt** The TXINTR indicates the Tx interrupt status. If TXINTR high and its corresponding enable bit, EnTXINTR of MISTA register, is also high indicates the EMC generates Tx interrupt to CPU. If TXINTR is high but EnTXINTR of MISTA is disabled, no Tx interrupt is generated. The TXINTR is a logic OR result of the bits 17~24 in MISTA register do logic AND with the corresponding bits in MIEN register. In other words, if one of the bits 17~24 in MISTA register is high and its corresponding enable bit in MIEN register is also enabled, the TXINTR will be high. Because the TXINTR is a logic OR result, clears bits 17~24 of MISTA register makes TXINTR be cleared, too. 1'b0: No status of bits 17~24 in MISTA is set or no enable of bits 17~24 in MIEN is turned on. 1'b1: At least one status of bits 17~24 in MISTA is set and its corresponding enable bit is turned on.
[14]	**CFR**	**Control Frame Receive Interrupt** The CFR high indicates EMC receives a flow control frame. The CFR only available while EMC is operating on full duplex mode. If the CFR is high and EnCFR of MIEN register is enabled, the RxINTR will be high. Write 1 to this bit clears the CFR status. 1'b0: The EMC doesn't receive the flow control frame. 1'b1: The EMC receives a flow control frame.
[11]	**RxBErr**	**Receive Bus Error Interrupt** The RxBErr high indicates the memory controller replies ERROR response while EMC access system memory through RxDMA during packet reception process. Reset EMC is recommended while RxBErr status is high. If the RxBErr is high and EnRxBErr of MIEN register is enabled, the RxINTR will be high. Write 1 to this bit clears the RxBErr status. 1'b0: No ERROR response is received. 1'b1: ERROR response is received.

[10]	**RDU**	**Receive Descriptor Unavailable Interrupt** The RDU high indicates that there is no available Rx descriptor for packet reception and RxDMA will stay at Halt state. Once, the RxDMA enters the Halt state, S/W must issues a write command to RSDR register to make RxDMA leave Halt state while new Rx descriptor is available. If the RDU is high and EnRDU of MIEN register is enabled, the RxINTR will be high. Write 1 to this bit clears the RDU status. 1'b0: Rx descriptor is available. 1'b1: Rx descriptor is unavailable.
[9]	**DENI**	**DMA Early Notification Interrupt** The DENI high indicates the EMC has received the Length/Type field of the incoming packet. If the DENI is high and EnDENI of MIEN register is enabled, the RxINTR will be high. Write 1 to this bit clears the DENI status. 1'b0: The Length/Type field of incoming packet has not received yet. 1'b1: The Length/Type field of incoming packet has received.
[8]	**DFOI**	**Maximum Frame Length Interrupt** The DFOI high indicates the length of the incoming packet has exceeded the length limitation configured in DMARFC register and the incoming packet is dropped. If the DFOI is high and EnDFO of MIEN register is enabled, the RxINTR will be high. Write 1 to this bit clears the DFOI status. 1'b0: The length of the incoming packet doesn't exceed the length limitation configured in DMARFC. 1'b1: The length of the incoming packet has exceeded the length limitation configured in DMARFC.
[7]	**MMP**	**More Missed Packet Interrupt** The MMP high indicates the MPCNT, Missed Packet Count, has overflow. If the MMP is high and EnMMP of MIEN register is enabled, the RxINTR will be high. Write 1 to this bit clears the MMP status. 1'b0: The MPCNT has not rolled over yet. 1'b1: The MPCNT has rolled over yet.
[6]	**RP**	**Runt Packet Interrupt** The RP high indicates the length of the incoming packet is less than 64 bytes and, the packet is dropped. If the ARP of MCMDR register is set, the short packet is regarded as a good packet and RP will not be set. If the RP is high and EnRP of MIEN register is enabled, the RxINTR will be high. Write 1 to this bit clears the RP status. 1'b0: The incoming frame is not a short frame or S/W wants to receive a short frame. 1'b1: The incoming frame is a short frame and dropped.
[5]	**ALIE**	**Alignment Error Interrupt** The ALIE high indicates the length of the incoming frame is not a multiple of byte. If the ALIE is high and EnALIE of MIEN register is enabled, the RxINTR will be high. Write 1 to this bit clears the ALIE status. 1'b0: The frame length is a multiple of byte. 1'b1: The frame length is not a multiple of byte.

[4]	**RXGD**	**Receive Good Interrupt** The RXGD high indicates the frame reception has completed. If the RXGD is high and EnRXGD of MIEN register is enabled, the RxINTR will be high. Write 1 to this bit clears the RXGD status. 1'b0: The frame reception has not complete yet. 1'b1: The frame reception has completed.
[3]	**PTLE**	**Packet Too Long Interrupt** The PTLE high indicates the length of the incoming packet is greater than 1518 bytes and the incoming packet is dropped. If the ALP of MCMDR register is set, the long packet will be regarded as a good packet and PTLE will not be set. If the PTLE is high and EnPTLE of MIEN register is enabled, the RxINTR will be high. Write 1 to this bit clears the PTLE status. 1'b0: The incoming frame is not a long frame or S/W wants to receive a long frame. 1'b1: The incoming frame is a long frame and dropped.
[2]	**RXOV**	**Receive FIFO Overflow Interrupt** The RXOV high indicates the RxFIFO overflow occurred during packet reception. While the RxFIFO overflow occurred, the EMC drops the current receiving packer. If the RxFIFO overflow occurred often, it is recommended that modify RxFIFO threshold control, the RxTHD of FFTCR register, to higher level. If the RXOV is high and EnRXOV of MIEN register is enabled, the RxINTR will be high. Write 1 to this bit clears the RXOV status. 1'b0: No RxFIFO overflow occurred during packet reception. 1'b0: RxFIFO overflow occurred during packet reception.
[1]	**CRCE**	**CRC Error Interrupt** The CRCE high indicates the incoming packet incurred the CRC error and the packet is dropped. If the AEP of MCMDR register is set, the CRC error packet will be regarded as a good packet and CRCE will not be set. If the CRCE is high and EnCRCE of MIEN register is enabled, the RxINTR will be high. Write 1 to this bit clears the CRCE status. 1'b0: The frame doesn't incur CRC error. 1'b1: The frame incurred CRC error.

[0]	**RXINTR**	**Receive Interrupt** The RXINTR indicates the Rx interrupt status. If RXINTR high and its corresponding enable bit, EnRXINTR of MISTA register, is also high indicates the EMC generates Rx interrupt to CPU. If RXINTR is high but EnRXINTR of MISTA is disabled, no Rx interrupt is generated. The RXINTR is a logic OR result of the bits 1~14 in MISTA register do logic AND with the corresponding bits in MIEN register. In other words, if one of the bits 1~14 in MISTA register is high and its corresponding enable bit in MIEN register is also enabled, the RXINTR will be high. Because the RXINTR is a logic OR result, clears bits 1~14 of MISTA register makes RXINTR be cleared, too. 1'b0: No status of bits 1~14 in MISTA is set or no enable of bits 1~14 in MIEN is turned on. 1'b1: At least one status of bits 1~14 in MISTA is set and its corresponding enable bit is turned on.

MAC General Status Register (MGSTA)

The MGSTA also keeps the statuses of EMC. But the statuses in the MGSTA will not trigger any interrupt. The MGSTA is a write clear register and write 1 to corresponding bit clears the status.

Register	Address	R/W	Description	Reset Value
MGSTA	0xB000_30B4	R/W	MAC General Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved				TXHA	SQE	PAU	DEF
7	6	5	4	3	2	1	0
CCNT				Reserved	RFFull	RXHA	CFR

Bits	Descriptions	
[11]	**TXHA**	**Transmission Halted** Default Value: 1'b0 The TXHA high indicates the next normal packet transmission process will be halted because the bit TXON of MCMDR is disabled be S/W. 1'b0: Next normal packet transmission process will go on. 1'b1: Next normal packet transmission process will be halted.
[10]	**SQE**	**Signal Quality Error** Default Value: 1'b0 The SQE high indicates the SQE error found at end of packet transmission on 10Mbps half-duplex mode. The SQE error check will only be done while both bit EnSQE of MCMDR is enabled and EMC is operating on 10Mbps half-duplex mode. 1'b0: No SQE error found at end of packet transmission. 1'b0: SQE error found at end of packet transmission.
[9]	**PAU**	**Transmission Paused** Default Value: 1'b0 The PAU high indicates the next normal packet transmission process will be paused temporally because EMC received a PAUSE control frame, or S/W set bit SDPZ of MCMDR and make EMC to transmit a PAUSE control frame out. 1'b0: Next normal packet transmission process will go on. 1'b1: Next normal packet transmission process will be paused.
[8]	**DEF**	**Deferred Transmission** Default Value: 1'b0 The DEF high indicates the packet transmission has deferred once. The DEF is only available while EMC is operating on half-duplex mode. 1'b0: Packet transmission doesn't defer. 1'b1: Packet transmission has deferred once.

[7:4]	**CCNT**	**Collision Count** Default Value: 4'h0 The CCNT indicates the how many collisions occurred consecutively during a packet transmission. If the packet incurred 16 consecutive collisions during transmission, the CCNT will be 4'h0 and bit TXABT will be set to 1.
[2]	**RFFull**	**RxFIFO Full** Default Value: 1'b0 The RFFull indicates the RxFIFO is full due to four 64-byte packets are kept in RxFIFO and the following incoming packet will be dropped. 1'b0: The RxFIFO is not full. 1'b1: The RxFIFO is full and the following incoming packet will be dropped.
[1]	**RXHA**	**Receive Halted** Default Value: 1'b0 The RXHA high indicates the next normal packet reception process will be halted because the bit RXON of MCMDR is disabled be S/W. 1'b0: Next normal packet reception process will go on. 1'b1: Next normal packet reception process will be halted.
[0]	**CFR**	**Control Frame Received** Default Value: 1'b0 The CFR high indicates EMC receives a flow control frame. The CFR only available while EMC is operating on full duplex mode. 1'b0: The EMC doesn't receive the flow control frame. 1'b1: The EMC receives a flow control frame.

Missed Packet Count Register (MPCNT)

The MPCNT keeps the number of packets that were dropped due to various types of receive errors. The MPCNT is a read clear register. In addition, S/W also can write an initial value to MPCNT and the missed packet counter will start counting from that initial value. If the missed packet counter is overflow, the MMP of MISTA will be set.

Register	Address	R/W	Description	Reset Value
MPCNT	0xB000_30B8	R/W	Missed Packet Count Register	0x0000_7FFF

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

Bits	Descriptions	
[15:0]	**MPC**	**Miss Packet Count** Default Value: 16'h7FFF The MPC indicates the number of packets that were dropped due to various types of receive errors. The following type of receiving error makes missed packet counter increase: ▪ Incoming packet is incurred RxFIFO overflow. ▪ Incoming packet is dropped due to RXON is disabled. Incoming packet is incurred CRC error.

MAC Receive Pause Count Register (MRPC)

The EMC of NUC946ADN supports the PAUSE control frame reception and recognition. If EMC received a PAUSE control frame, the operand field of the PAUSE control frame will be extracted and stored in the MRPC register. The MRPC register will keep the same while Tx of EMC is pausing due to the PAUSE control frame is received. The MRPC is read only and write to this register has no effect.

Register	Address	R/W	Description	Reset Value
MRPC	0xB000_30BC	R	MAC Receive Pause Count Register	0x0000_0000

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

Bits	Descriptions	
[15:0]	**MRPC**	**MAC Receive Pause Count** Default Value: 16'h0 The MRPC keeps the operand field of the PAUSE control frame. It indicates how many slot time (512 bit time) the Tx of EMC will be paused.

MAC Receive Pause Current Count Register (MRPCC)

The EMC of NUC946ADN supports the PAUSE control frame reception and recognition. If EMC received a PAUSE control frame, the operand field of the PAUSE control frame will be extracted and stored into a down count timer. The MRPCC shows the current value of that down count timer for S/W to know how long the Tx of EMC will be paused. The MRPCC is read only and write to this register has no effect.

Register	Address	R/W	Description	Reset Value
MRPCC	0xB000_30C0	R	MAC Receive Pause Current Count Register	0x0000_0000

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

Bits	Descriptions	
[15:0]	**MRPCC**	**MAC Receive Pause Current Count** Default Value: 16'h0 The MRPCC shows the current value of that down count timer. If a new PAUSE control frame is received before the timer count down to zero, the new operand of the PAUSE control frame will be stored into the down count timer and the timer starts count down from the new value.

MAC Remote Pause Count Register (MREPC)

The EMC of NUC946ADN supports the PAUSE control frame transmission. After the PAUSE control frame is transmitted out completely, a timer starts to count down from the value of operand of the transmitted PAUSE control frame. The MREPC shows the current value of this down count timer. The MREPC is read only and write to this register has no effect.

Register	Address	R/W	Description	Reset Value
MREPC	0xB000_30C4	R	MAC Remote Pause Count Register	0x0000_0000

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

Bits	Descriptions	
[15:0]	**MREPC**	**MAC Remote Pause Count** Default Value: 16'h0 The MREPC shows the current value of the down count timer that starts to count down from the value of operand of the transmitted PAUSE control frame.

DMA Receive Frame Status Register (DMARFS)

The DMARFS is used to keep the Length/Type field of each incoming Ethernet packet. This register is writing clear and writes 1 to corresponding bit clears the bit.

Register	Address	R/W	Description	Reset Value
DMARFS	0xB000_30C8	R/W	DMA Receive Frame Status Register	0x0000_0000

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

Bits	Descriptions	
[15:0]	**RXFLT**	**Receive Frame Length/Type** Default Value: 16'h0 The RXFLT keeps the Length/Type field of each incoming Ethernet packet. If the bit EnDEN of MIEN is enabled and the Length/Type field of incoming packet has received, the bit DENI of MISTA will be set and trigger interrupt. And, the content of Length/Type field will be stored in RXFLT.

Current Transmit Descriptor Start Address Register (CTXDSA)

Register	Address	R/W	Description	Reset Value
CTXDSA	0xB000_30CC	R	Current Transmit Descriptor Start Address Register	0x0000_0000

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

Bits	Descriptions	
[31:0]	**CTXDSA**	**Current Transmit Descriptor Start Address** Default Value: 32'h0 The CTXDSA keeps the start address of Tx descriptor that is used by TxDMA currently. The CTXDSA is read only and write to this register has no effect.

Current Transmit Buffer Start Address Register (CTXBSA)

Register	Address	R/W	Description	Reset Value
CTXBSA	0xB000_30D0	R	Current Transmit Buffer Start Address Register	0x0000_0000

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

Bits	Descriptions	
[31:0]	**CTXBSA**	**Current Transmit Buffer Start Address** Default Value: 32'h0 The CTXDSA keeps the start address of Tx frame buffer that is used by TxDMA currently. The CTXBSA is read only and write to this register has no effect.

Current Receive Descriptor Start Address Register (CRXDSA)

Register	Address	R/W	Description	Reset Value
CRXDSA	0xB000_30D4	R	Current Receive Descriptor Start Address Register	0x0000_0000

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

Bits	Descriptions	
[31:0]	**CRXDSA**	**Current Receive Descriptor Start Address** Default Value: 32'h0 The CRXDSA keeps the start address of Rx descriptor that is used by RxDMA currently. The CRXDSA is read only and write to this register has no effect.

Current Receive Buffer Start Address Register (CRXBSA)

Register	Address	R/W	Description	Reset Value
CRXBSA	0xB000_30D8	R	Current Receive Buffer Start Address Register	0x0000_0000

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

Bits	Descriptions	
[31:0]	**CRXBSA**	**Current Receive Buffer Start Address** Default Value: 32'h0 The CRXBSA keeps the start address of Rx frame buffer that is used by RxDMA currently. The CRXBSA is read only and write to this register has no effect.

7.5.4 Operation Notes

MII Management Interface

The operation mode between EMC and external PHY must be identically. Consequently, S/W has to access control register of external PHY through MII management interface to get operation information of PHY. To issue MII management command to access external PHY, the MIID and MIIDA registers can be used. And, while using MII management interface, the EnMDC of MCMDR register must be set to high.

EMC Initial

If S/W wants to enable EMC for packet transmission and reception, the TXON and RXON of MCMDR register must be enabled. But, before enabling TXON and RXON, the following issues must be noted.
For packet transmission, the Tx descriptor link list and Tx frame buffer must be prepared and TXDLSA must be configured.
For incoming packet destination MAC address recognition, the CAMCMR, CAMEN, CAMxM and CAMxL registers must be configured. For incoming packet's buffering, the Rx descriptor link list and Rx frame buffer must be prepared and RXDLSA register must be configured.
Besides, the interrupt status that S/W wants to know must be enabled through MIEN register.
Finally, the EMC operation mode control bits of MCMDR must be configured and TXON and RXON must be enabled.

MAC Interrupt Status Register (MISTA)

The MISTA register keeps the status of EMC operation. It is recommended that S/W must enable four interrupt statuses at least. They are TxBErr, RxBErr, TDU and RDU.
While EMC accesses memory, it reports the memory error through TxBErr or TxBErr status. If any of them actives, the reset EMC is recommended.
For packet transmission, a valid Tx descriptor is required, and for packet reception, a valid Rx one is. If EMC cannot find a valid Tx or Rx descriptor, it sets TDU or RDU to high respectively. After S/W releases a valid Tx or Rx descriptor to EMC, writing TSDR or RSDR register to enable packet transmission and reception again is needed.

Pause Control Frame Transmission

The EMC support the PAUSE control frame transmission for flow control while EMC is operating on full-duplex mode. The register CAM13M, CAM13L, CAM14M, CAM14L, CAM15M and CAM15L are designed for this purpose.
For PAUSE control frame transmission, first, S/W must configure destination MAC address of control frame into the register pair {CAM13M, CAM13L}, source MAC address into the register pair {CAM14M, CAM14L}, and configure length/type, op-code and operand of control frame into the register pair {CAM15M, CAM15L}. The bit CAM13EN, CAM14EN and CAM15EN of CAMEN register are also needed be enabled. Then, set bit SDPZ of MCMDR register to high to enable PAUSE control frame transmission. After the PAUSE control frame transmission completed, the SDPZ will be cleared automatically.

Internal Loop-back

If the LBK of MCMDR register is set, the EMC operates on internal loop-back mode. While EMC operates on internal loop-back mode, it also means EMC operates on full-duplex mode, and the value of FDUP of MCMDR register is ignored.

7.6 GDMA Controller

7.6.1 Overview & Features

The chip has a two-channel general DMA controller with or without descriptor fetch operation, called the GDMA. The two-channel GDMA performs the following data transfers without the CPU intervention:

- Memory-to-memory (memory to/from memory)
- Memory –to – IO
- IO- to -memory

The on-chip GDMA can be started by the software or external DMA request nXDREQ0/1. Software can also be used to restart the GDMA operation after it has been stopped. The CPU can recognize the completion of a GDMA operation by software polling or when it receives an internal GDMA interrupt. The GDMA controller can increment source or destination address, decrement them as well, and conduct 8-bit (byte) or 16-bit (half-word) data transfers.

7.6.2 GDMA Descriptor Functional Description

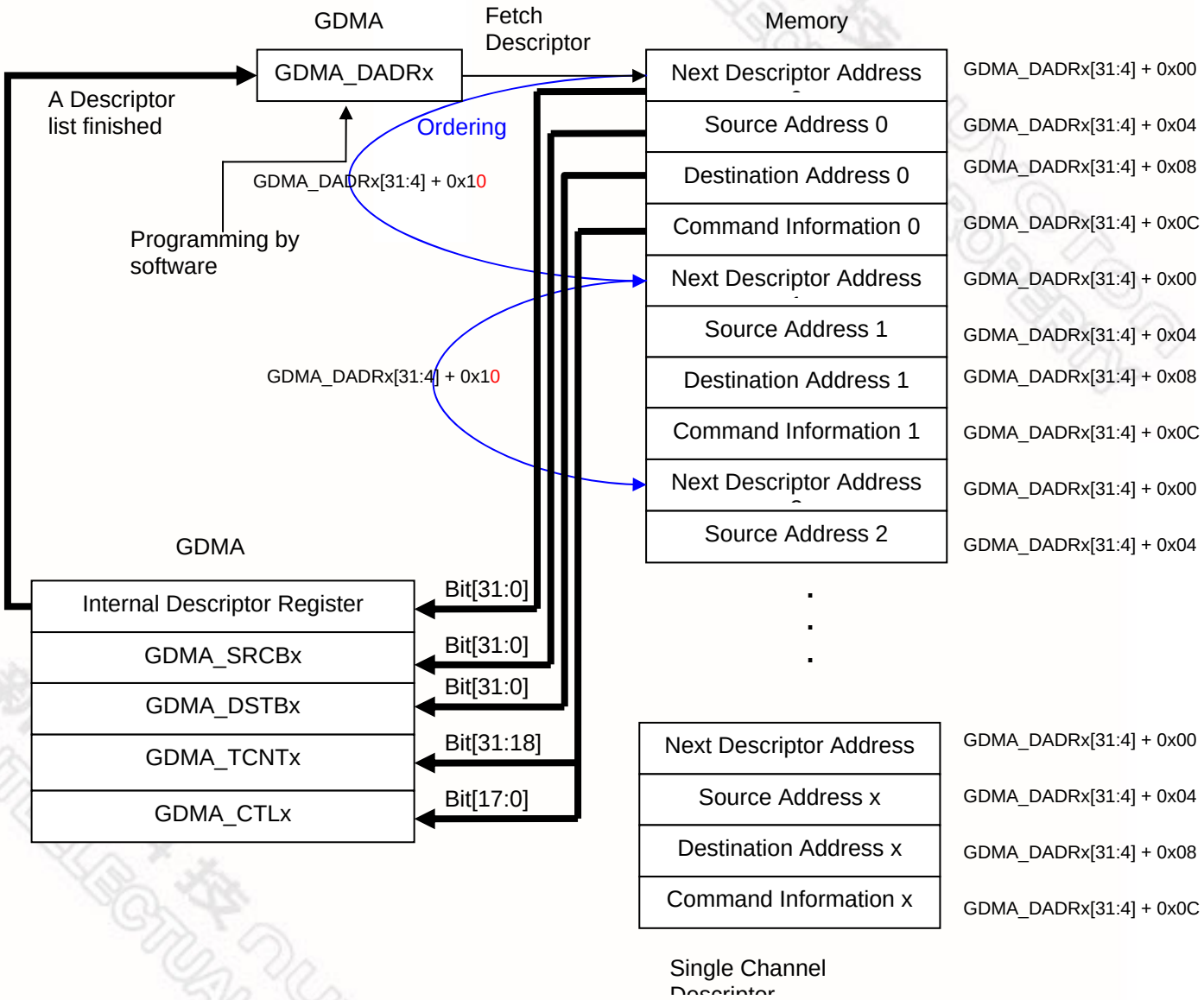
The descriptor-fetch function works when run-bit (bit-3) is set and non-dsptrmode-bit (bit-2) is cleared in Descriptor Register (GDMA_DADRx) and the GDMA_CTLx bit setting as following table. The Non-descriptor-fetch function works when software triggers the [softreq] bit (bit-16) and the [gdmaen] bit (bit-0) in GDMA_CTLx Register. If the [softreq] set to zero and the [GDMAMS] (bit2-3) set as 01 or 10 will start the I/O to memory function. Software can also be used to restart the GDMA operation after it has been stopped. The CPU can recognize the completion of a GDMA operation by software polling or when it receives an internal GDMA interrupt. The GDMA controller can increase source or destination address, decrease them as well, and conduct 8-bit (byte), 16-bit (half-word), or 32-bit (word) data transfers.

Operation Mode relevant to enable bit

Mode	Enable bit
Non-Descriptor Mode with SW Enable	GDMA_CTLx : gdmaen[0] softreq[16] gdmams[3:2]
Non-Descriptor Mode with I/O Enable	GDMA_CTLx : gdmaen[0] gdmams[3:2]
Descriptor Mode with SW Enable	GDMA_DADRx : run[3] non-dsptrmode[2]; GDMA_CTLx in Descriptor List : gdmaen[0] gdmams[3:2]
Descriptor Mode with I/O Enable	GDMA_DADRx : run[3] non-dsptrmode[2]; GDMA_CTLx in Descriptor List : gdmaen[0] gdmams[3:2]

7.6.2.1 Descriptor Fetch Function

The Illustration of Descriptor list fetches:



Descriptor-based function (GDMA_DADRx [NON_DSPTRMODE] = 0) operate in the following condition:

Memory to Memory

1. Software can write a value 0x04 to current GDMA_DADRx register to reset the register and disable Descriptor based function first.
2. Then software can program the bits of [Descriptor Address], [RUN], [NON_DSPTRMODE] and [ORDEN] to the GDMA_DADRx register to enable Descriptor based function. (The Descriptor can only work when the [RUN] [3] is set and [NON_DSPTRMODE] [2] bit is cleared properly.)
3. After sets current GDMA_DADRx register, the GDMA will fetch four-word information from memory immediately which contains the next Descriptor address, Source Address, Destination Address and Command information. (Command information consists of control and counter registers)

NOTE: GDMA will read the descriptor list from memory such the diagram above and write back to GDMA internal register (next GDMA_DADRx), GDMA_SRCBx, GDMA_DSTBx, GDMA_CTLx and GDMA_TCNTx registers. The most important one of write back is command information, which will separate some bits of command information into control and counter registers respectively. The first fourteen bits of the MSB of the Command information in Descriptor list will be written back to GDMA_TCNTx register, and the others bits of the Command information will be written back to GDMA_CTLx register. The control register part of the Command information will update the GDMA_CTLx register during every descriptor fetch. The allocation of command information is described at GDMA Register Descriptions.

The Allocation of Command Information in Descriptor List:

31	30	29	28	27	26	25	24
GDMA_TCNTx[13:6] ← Command Info[31:24]							
23	22	21	20	19	18	17	16
GDMA_TCNTx[5:0] ← Command Info[23:18]						BLOCK	SOFTREQ
15	14	13	12	11	10	9	8
TWS		RESERVED		D_INTS	D_INTS	RESERVED	
7	6	5	4	3	2	1	0
SAFIX	DAFIX	SADIR	DADIR	GDMAMS		BME	GDMAEN

4. GDMA will depend on the information to request a bus ownership and start the data transfer when GDMA has gotten a bus grant from the arbiter, otherwise, it will wait until get bus grant. The data transfer direction is dependent on the Control register.
5. The GDMA transfers data and releases bus at every burst transfer. The GDMA will stop transfer for current descriptor when the counter is decreased to zero. The current GDMA_DADRx will be updated by next GDMA_DADRx at end of each descriptor transfer.
6. The GDMA is running consecutively unless the next GDMA_DADRx[RUN] bit is zero or interrupt status bit of GDMA_INTCS register is cleared. The CPU can recognize the completion of a GDMA descriptor fetch operation by polling the current GDMA_DADRx[NON_DSPTRMODE] bit or set the GDMA_CTLx[D_INTS] to receive a interrupt from GDMA.(Note: The recommendation is the [NON_DSPTRMODE] bit in list is set at the same time)
7. When an error occurs in the descriptor operation, GDMA will clear [RUN] bit and stop channel operation immediately. Software can reset the channel, and sets the current GDMA_DADRx [RUN] register to

start again.

Memory to I/O and I/O to Memory

1. Software must set the [REQ_ATV], [ACK_ATV] and [GDMAMS] bits in GDMA_CTLx register corresponding to I/O pin with pull high or pull low properly first, and then set the current GDMA_DADRx to start the I/O to Memory with descriptor fetch transfer.
2. The descriptor lists stop transfer until the RUN bit was zero in descriptor list when external I/O request triggered once. The RUN bit can be set when external I/O request triggered again under the NON_DSPTRMODE bit was zero in descriptor list. The trigger period of the external I/O has a timing limitation whatever the GDMA was in single or burst mode, and the periodic trigger of the external I/O must be less than 38 MCLK.
3. Each GDMA lists can operate after clearing interrupt status. The descriptor lists stop transfer until the RUN bit was zero or interrupt status was set.
4. The next Descriptor address, Source Address, Destination Address and Command information must be set properly in every Descriptor list. Especially, every bit of the Command information will update the GDMA_CTLx and GDMA_TCNTx registers at every initiation of descriptor list.

NOTE: The [BLOCK] bit of GDMA_CTLx register is disabled when the descriptor mode of the I/O to memory is enabled.

NOTE: GDMA can change mode with following description:
Descriptor-fetch of each channel can be stopped until the current transfer list done. Software can change Descriptor mode to Non-Descritpor mode by writing 0x04 to GDMA_DADRx register during the current descriptor transfer operating.
Non-Descriptor fetch can be stopped until current transfer count finished when software programs the GDMA_CTLx register with gdmaen bit cleared or softreq cleared.

NOTE: Once software programs the current GDMA_DADRx register, GDMA will fetch the descriptor list from memory and fill the data to next GDMA_DADRx, current GDMA_SRCBx, current GDMA_DSTBx, current GDMA_CTLx and current GDMA_TCNTx registers automatically. The fourth word in descriptor list includes the information for GDMA_CTLx and GDMA_TCNTx registers.

NOTE: The descriptor fetch function only occurs when current GDMA_DADRx [RUN] bit is set and GDMA_DADRx [NON_DSPTRMODE] is cleared. The current GDMA_DADRx will be updated by next GDMA_DADRx at every descriptor stops.

7.6.2.2 Ordering function in Descriptor fetch mode

This function determines the source of next descriptor address. If [ORDEN] is set, the GDMA controller fetches the next descriptor from current GDMA_DADRx [Descriptor Address] + 16 bytes.
If this bit is cleared, GDMA fetches the next descriptor from the current GDMA_DADRx [Descriptor Address].
GDMA_DADRx [ORDEN] is only relevant to descriptor-fetch function (GDMA_DADRx [NON_DSPTRMODE] = 0).

7.6.2.3 Channel Reset

The Channel reset is turned on when the bit-0 of GDMA_DADRx is set. This function will clear all status and stop the descriptor based function relative to individual channel. The GDMA_DADRx register value is 0x05h when reset bit is set.

7.6.2.4 Non-Descriptor Fetch Function

The non-descriptor-fetch function will take place when current GDMA_DADRx [NON_DSPTRMODE] is set and the GDMA_DADRx register will have no any intention for the GDMA controller.
The default value of GDMA_DADRx is 0x04. Software can clear GDMA_DADRx with value 0x04 as well. In this mode, software should write a valid source address to the GDMA_SRCBx register, a destination address to the GDMA_DSTBx register, and a transfer count to the GDMA_TCNTx register. Next, the GDMA_CTLx of [gdmaen] and [softreq] bits must be set. A non-descriptor fetch is performed when bus granted. After transferring a number of bytes or words correspond with burst mode or not, the channel either waits for the next request or continues with the data transfer until the GDMA_CTCNTx reaches zero. When GDMA_CTCNTx reaches zero, the channel stops operation.
When an error occurs during the GDMA operation, the channel stops unless software clears the error condition and sets the GDMA_CTLx of [gdmaen] and [softreq] bits field to start again.

7.6.3 GDMA Register Map

R: read only, **W**: write only, **R/W**: both read and write, **C**: Only value 0 can be written

Register	Address	R/W	Description	Reset Value
GDMA_BA = 0xB000_4000				
Channel 0				
GDMA_CTL0	0xB000_4000	R/W	Channel 0 Control Register	0x0000_0000
GDMA_SRCB0	0xB000_4004	R/W	Channel 0 Source Base Address Register	0x0000_0000
GDMA_DSTB0	0xB000_4008	R/W	Channel 0 Destination Base Address Register	0x0000_0000
GDMA_TCNT0	0xB000_400C	R/W	Channel 0 Transfer Count Register	0x0000_0000
GDMA_CSRC0	0xB000_4010	R	Channel 0 Current Source Address Register	0x0000_0000
GDMA_CDST0	0xB000_4014	R	Channel 0 Current Destination Address Reg.	0x0000_0000
GDMA_CTCNT0	0xB000_4018	R	Channel 0 Current Transfer Count Register	0x0000_0000
GDMA_DADR0	0xB000_401C	R/W	Channel 0 Descriptor Address Register	0x0000_0004
Channel 1				
GDMA_CTL1	0xB000_4020	R/W	Channel 1 Control Register	0x0000_0000
GDMA_SRCB1	0xB000_4024	R/W	Channel 1 Source Base Address Register	0x0000_0000
GDMA_DSTB1	0xB000_4028	R/W	Channel 1 Destination Base Address Register	0x0000_0000
GDMA_TCNT1	0xB000_402C	R/W	Channel 1 Transfer Count Register	0x0000_0000
GDMA_CSRC1	0xB000_4030	R	Channel 1 Current Source Address Register	0x0000_0000
GDMA_CDST1	0xB000_4034	R	Channel 1 Current Destination Address Reg.	0x0000_0000
GDMA_CTCNT1	0xB000_4038	R	Channel 1 Current Transfer Count Register	0x0000_0000
GDMA_DADR1	0xB000_403C	R/W	Channel 1 Descriptor Address Register	0x0000_0004
GDMA_INTBUF0	0xB000_4080	R	GDMA Internal Buffer Word 0	0x0000_0000
GDMA_INTBUF1	0xB000_4084	R	GDMA Internal Buffer Word 1	0x0000_0000
GDMA_INTBUF2	0xB000_4088	R	GDMA Internal Buffer Word 2	0x0000_0000
GDMA_INTBUF3	0xB000_408C	R	GDMA Internal Buffer Word 3	0x0000_0000
GDMA_INTBUF4	0xB000_4090	R	GDMA Internal Buffer Word 4	0x0000_0000
GDMA_INTBUF5	0xB000_4094	R	GDMA Internal Buffer Word 5	0x0000_0000
GDMA_INTBUF6	0xB000_4098	R	GDMA Internal Buffer Word 6	0x0000_0000
GDMA_INTBUF7	0xB000_409C	R	GDMA Internal Buffer Word 7	0x0000_0000
GDMA_INTCS	0xB000_40A0	R/W	Interrupt Control and Status Register (2 Channels)	0x0000_0000

Channel 0/1 Control Register (GDMA_CTL0, GDMA_CTL1)

Register	Address	R/W	Description	Reset Value
GDMA_CTL0	0xB000_4000	R/W	Channel 0 Control Register	0x0000_0000
GDMA_CTL1	0xB000_4020	R/W	Channel 1 Control Register	0x0000_0000

The control registers has two formats for descriptor fetch and non-descriptor fetch function respectively. The functionality of each control bit is described in following table.

1. Non-Descriptor fetches Mode

31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED	SABNDERR	DABNDERR	RESERVED	AUTOIEN	RESERVED	BLOCK	SOFTREQ
15	14	13	12	11	10	9	8
RESERVED		TWS		SBMS	RESERVED		
7	6	5	4	3	2	1	0
SAFIX	DAFIX	SADIR	DADIR	GDMAMS		BME	GDMAEN

2. Descriptor fetches Mode

31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED	SABNDERR	DABNDERR	RESERVED			BLOCK	SOFTREQ
15	14	13	12	11	10	9	8
RESERVED		TWS		RESERVED	D_INTS	RESERVED	
7	6	5	4	3	2	1	0
SAFIX	DAFIX	SADIR	DADIR	GDMAMS		BME	GDMAEN

NOTE:

- The bit [REQ_ATV] and [ACK_ATV] must be set first before using I/O to Memory mode with Descriptor fetch transfer. These two bits cannot do any setup in command information within descriptor list configuration. The [SABNDERR], [DABNDERR], [GDMAERR] can also be read at descriptor fetch mode.
- Regardless of GDMA operate in descriptor mode or non-descriptor mode, when transfer width is 16-bit (half word) and the address with decrement function enable for starting source address or destination address or both are used should set the least two bit of addresses is 0xF.

Control Register of Non-Descriptor fetches Mode:

Bits	Descriptions	
[22]	**SABNDERR**	**Source Address Boundary Alignment Error Flag** If TWS [13:12]=10, GDMA_SRCB [1:0] should be 00 If TWS [13:12]=01, GDMA_SRCB [0] should be 0 Except the SADIR function enabled. The address boundary alignment should be depended on TWS [13:12]. 0 = the GDMA_SRCB is on the boundary alignment. 1 = the GDMA_SRCB not on the boundary alignment The SABNDERR register bits just can be read only.
[21]	**DABNDERR**	**Destination Address Boundary Alignment Error Flag** If TWS [13:12]=10, GDMA_DSTB [1:0] should be 00 If TWS [13:12]=01, GDMA_DSTB [0] should be 0 Except the SADIR function enabled. The address boundary alignment should be depended on TWS [13:12]. 0 = the GDMA_DSTB is on the boundary alignment. 1 = the GDMA_DSTB not on the boundary alignment The DABNDERR register bits just can be read only.
[19]	**AUTOIEN**	**Auto initialization Enable** 0 = Disables auto initialization 1 = Enables auto initialization, the GDMA_CSRC0/1, GDMA_CDST0/1, and GDMA_CTCNT0/1 registers are updated by the GDMA_SRC0/1, GDMA_DST0/1, and GDMA_TCNT0/1 registers automatically when transfer is complete. GDMA will start another transfer when SOFTREQ set again.
[17]	**BLOCK**	**Bus Lock** 0 = Unlocks the bus during the period of transfer 1 = locks the bus during the period of transfer
[16]	**SOFTREQ**	**Software Triggered GDMA Request** Software can request the GDMA transfer service by setting this bit to 1. This bit is automatically cleared by hardware when the transfer is completed. This bit is available only while GDMAMS [3:2] register bits are set on software mode (memory to memory and memory to I/O).
[13:12]	**TWS**	**Transfer Width Select** 00 = One byte (8 bits) is transferred for every GDMA operation 01 = One half-word (16 bits) is transferred for every GDMA operation 10 = One word (32 bits) is transferred for every GDMA operation 11 = Reserved The GDMA_SCRB and GDMA_DSTB should be alignment under the TWS selection

[11]	**SBMS**	**Single/Block Mode Select** 0 = Selects single mode. It requires an external GDMA request for every incurring GDMA operation. 1 = Selects block mode. It requires a single external GDMA request during the atomic GDMA operation. An atomic GDMA operation is defined as the sequence of GDMA operations until the transfer count register reaches zero.
[7]	**SAFIX**	**Source Address Fixed** 0 = Source address is changed during the GDMA operation 1 = Do not change the destination address during the GDMA operation. This feature can be used when data were transferred from a single source to multiple destinations.
[6]	**DAFIX**	**Destination Address Fixed** 0 = Destination address is changed during the GDMA operation 1 = Do not change the destination address during the GDMA operation. This feature can be used when data were transferred from multiple sources to a single destination.
[5]	**DADIR**	**Source Address Direction** 0 = Source address is incremented successively 1 = Source address is decremented successively
[4]	**DADIR**	**Destination Address Direction** 0 = Destination address is incremented successively 1 = Destination address is decremented successively
[3:2]	**GDMAMS**	**GDMA Mode Select** 00 = Software mode (memory-to-memory) 01 = External nXDREQ0 mode for external device (I/O to Memory) 10 = Reserved 11 = Reserved
[1]	**BME**	**Burst Mode Enable** 0 = Disables the 8-data burst mode 1 = Enables the 8-data burst mode If there are 8 words to be transferred, and the BME [1] =1, the GDMA_TCNTx should be 0x01. However, if BME [1] =0, the GDMA_TCNTx should be 0x08. It has to set BME [1] = 0 for I/O device access.
[0]	**GDMAEN**	**GDMA Enable** 0 = Disables the GDMA operation 1 = Enables the GDMA operation; this bit will be clear automatically when the transfer is complete on AUTOIEN [19] register bit is on Disable mode. Note: When operate in Non-Descriptor mode, this bit determines the Memory-to Memory, Memory-to-I/O and I/O-to-Memory operation or not. When operate in Descriptor mode, this bit is determined in descriptor list. Note: Channel reset will clear this bit.

Descriptor fetches mode of Control Register:

Bits	Descriptions	
[22]	**SABNDERR**	**Source Address Boundary Alignment Error Flag** If TWS [13:12]=10, GDMA_SRCB [1:0] should be 00 If TWS [13:12]=01, GDMA_SRCB [0] should be 0 Except the SADIR function enabled. The address boundary alignment should be depended on TWS [13:12]. 0 = the GDMA_SRCB is on the boundary alignment. 1 = the GDMA_SRCB not on the boundary alignment The SABNDERR register bits just can be read only.
[21]	**DABNDERR**	**Destination Address Boundary Alignment Error Flag** If TWS [13:12]=10, GDMA_DSTB [1:0] should be 00 If TWS [13:12]=01, GDMA_DSTB [0] should be 0 Except the DADIR function enabled. The address boundary alignment should be depended on TWS [13:12]. 0 = the GDMA_DSTB is on the boundary alignment. 1 = the GDMA_DSTB not on the boundary alignment The DABNDERR register bits just can be read only.
[17]	**BLOCK**	**Bus Lock** 0 = Unlocks the bus during the period of transfer 1 = locks the bus during the period of transfer
[13:12]	**TWS**	**Transfer Width Select** 00 = One byte (8 bits) is transferred for every GDMA operation 01 = One half-word (16 bits) is transferred for every GDMA operation 10 = One word (32 bits) is transferred for every GDMA operation 11 = Reserved The GDMA_SCRB and GDMA_DSTB should be alignment under the TWS selection
[10]	**D_INTS**	**Descriptor Fetch Mode Interrupt Select** 0 = The interrupt will take place at every end of descriptor fetch transfer. 1 = The interrupt only take place at the last descriptor fetch transfer. NOTE: this bit is only available in descriptor mode and lists intention.
[7]	**SAFIX**	**Source Address Fixed** 0 = Source address is changed during the GDMA operation 1 = Do not change the source address during the GDMA operation. This feature can be used when data were transferred from a single source to multiple destinations.
[6]	**DAFIX**	**Destination Address Fixed** 0 = Destination address is changed during the GDMA operation 1 = Do not change the destination address during the GDMA operation. This feature can be used when data were transferred from multiple sources to a single destination.

[5]	**SADIR**	**Source Address Direction** 0 = Source address is incremented successively 1 = Source address is decremented successively
[4]	**DADIR**	**Destination Address Direction** 0 = Destination address is incremented successively 1 = Destination address is decremented successively
[3:2]	**GDMAMS**	**GDMA Mode Select** 00 = Software mode (Memory-to-Memory) 01 = External nXDREQ0 mode for external device(I/O-to-Memory) 10 = Reserved 11 = Reserved
[1]	**BME**	**Burst Mode Enable** 0 = Disables the 8-data burst mode 1 = Enables the 8-data burst mode FF there are 8 words to be transferred, and BME [1]=1, the GDMA_TCNT should be 0x01; However, if BME [1] =0, the GDMA_TCNT should be 0x08. It has to set BME [1] = 0 for I/O device access.
[0]	**GDMAEN**	**GDMA Enable** 0 = Disables the GDMA operation 1 = Enables the GDMA operation; this bit will be clear automatically when the transfer is complete on AUTOIEN [19] register bit is on Disable mode. When operate in Non-Descriptor mode, this bit determines the Memory-to-Memory, Memory-to-I/O and I/O-to-Memory operation or not. When operate in Descriptor mode, this bit determines the I/O-to-Memory operation or not. Channel reset will clear this bit.

Channel 0 / 1 Source Base Address Register (GDMA_SRCB0, GDMA_SRCB1)

Register	Address	R / W	Description	Reset Value
GDMA_SRCB0	0xB000_4004	R/W	Channel 0 Source Base Address Register	0x0000_0000
GDMA_SRCB1	0xB000_4024	R/W	Channel 1 Source Base Address Register	0x0000_0000

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

Bits	Descriptions	
[31:0]	**SRC_BASE_ADDR**	**32-bit Source Base Address** The GDMA channel starts reading its data from the source address as defined in this source base address register.

Channel 0/1 Destination Base Address Register (GDMA_DSTB0, GDMA_DSTB1)

Register	Address	R/W	Description	Reset Value
GDMA_DSTB0	0xB000_4008	R/W	Channel 0 Destination Base Address Register	0x0000_0000
GDMA_DSTB1	0xB000_4028	R/W	Channel 1 Destination Base Address Register	0x0000_0000

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

Bits	Descriptions	
[31:0]	**DST_BASE_ADDR**	**32-bit Destination Base Address** The GDMA channel starts writing its data to the destination address as defined in this destination base address register. During a block transfer, the GDMA determines successive destination addresses by adding to or subtracting from the destination base address.

Channel 0/1 Transfer Count Register (GDMA_TCNT0, GDMA_TCNT1)

Register	Address	R/W	Description	Reset Value
GDMA_TCNT0	0xB000_400C	R/W	Channel 0 Transfer Count Register	0x0000_0000
GDMA_TCNT1	0xB000_402C	R/W	Channel 1 Transfer Count Register	0x0000_0000

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

Bits	Descriptions	
[23:0]	**TFR_CNT**	**Transfer Count** **Non-Descriptor Mode:24-bit TFR_CNT [23:0]** The TFR_CNT represents the required number of GDMA transfers. The maximum transfer count is 16M –1. **Descriptor Mode: 14-bit TFR_CNT [13:0]** The TFR_CNT represents the required number of GDMA transfers. The maximum transfer count is 16K –1.

Channel 0 / 1 Current Source Register (GDMA_CSRC0, GDMA_CSRC1)

Register	Address	R / W	Description	Reset Value
GDMA_CSRC0	0xB000_4010	R	Channel 0 Current Source Address Register	0x0000_0000
GDMA_CSRC1	0xB000_4030	R	Channel 1 Current Source Address Register	0x0000_0000

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

Bits	Descriptions	
[31:0]	**CURRENT_SRC_ADDR**	**32-bit Current Source Address** The CURRENT_SRC_ADDR indicates the source address where the GDMA transfer is just occurring. During a block transfer, the GDMA determines the successive source addresses by adding to or subtracting from the source base address. Depending on the settings you make to the control register, the current source address will remain the same or will be incremented or decremented.

Channel 0/1 Current Destination Register (GDMA_CDST0, GDMA_CDST1)

Register	Address	R/W	Description	Reset Value
GDMA_CDST0	0xB000_4014	R	Channel 0 Current Destination Address Register	0x0000_0000
GDMA_CDST1	0xB000_4034	R	Channel 1 Current Destination Address Register	0x0000_0000

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

Bits	Descriptions	
[31:0]	**CURRENT_DST_ADDR**	**32-bit Current Destination Address** The CURRENT_DST_ADDR indicates the destination address where the GDMA transfer is just occurring. During a block transfer, the GDMA determines the successive destination addresses by adding to or subtracting from the destination base address. Depending on the settings you make to the control register, the current destination address will remain the same or will be incremented or decremented.

Channel 0 / 1 Current Transfer Count Register (GDMA_CTCNT0, GDMA_CTCNT1)

Register	Address	R / W	Description	Reset Value
GDMA_CTCNT0	0xB000_4018	R	Channel 0 Current Transfer Count Register	0x0000_0000
GDMA_CTCNT1	0xB000_4038	R	Channel 1 Current Transfer Count Register	0x0000_0000

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

Bits	Descriptions	
[23:0]	**CURRENT_TFR_CNT**	**Current Transfer Count** The Current transfer count register indicates the number of transfer being performed. **Non-Descriptor Mode: 24-bit CURENT_TFR_CNT [23:0]** **Descriptor Mode : 14-bit CURENT_TFR_CNT [13:0]**

Channel 0/1 Descriptor Register (GDMA_DADR0/1)

	Address	R/W	Description	Reset Value
GDMA_DADR0	0xB000_401C	R/W	Channel 0 Control Register	0x0000_0004
GDMA_DADR1	0xB000_403C	R/W	Channel 1 Control Register	0x0000_0004

31	30	29	28	27	26	25	24
Descriptor Address[31:24]							
23	22	21	20	19	18	17	16
Descriptor Address[23:16]							
15	14	13	12	11	10	9	8
Descriptor Address[15:8]							
7	6	5	4	3	2	1	0
Descriptor Address[7:4]				RUN	NON_DSPTRMODE	ORDEN	RESET

Bits	Descriptions	
[31:4]	**Descriptor Address**	**Descriptor Address** Contains address of next descriptor.
[3]	**RUN**	**Run** The RUN bit can be cleared during descriptor data transfer, and set RUN bit to starts the stopped channel under [Descriptor Address] and [Non-DSPTRMODE] bits are set properly. When RUN bit is cleared and the NON_DSPTRMODE bit is set that non-descriptor fetch occurs whether a valid descriptor address is written to register GDMA_DADRx or not. This bit will reset automatically when each descriptor transfer stopped or the bit in descriptor list is zero. The Descriptor interrupt is determined by bit-10 of the GDMA_CTLx Register. 0 = Stops the channel. 1 = Starts the channel. Note: must co-operate to [NON_DSPTRMODE] to start the channel with Descriptor fetch function.
[2]	**NON_DSPTRMODE**	**Non-Descriptor-Fetch** When NON_DSPTRMODE is set, the channel is considered as a channel with no descriptors. In this mode, the GDMA does not initiate descriptor fetching and software can program the SCRBx, DSTBx, CTRx and TCNTx registers to transfer data until the TCNTx reaches zero. The GDMA_DADRx register is not used in non-descriptor mode. If NON_DSPTRMDOE is cleared under [RUN] and [Descriptor Address] are set properly, GDMA controller initiates descriptor-fetching. The descriptor fetch transfer stops when the counter for the current transfer reaches zero, [RUN] bit is cleared and [NON_DSPTRMODE] is set base on the bits of the descriptor list.

		0 = Descriptor-fetch transfer 1 = NON-descriptor-fetch transfer **Note: this bit = 1 will disable Descriptor function regardless of the RUN bit is 1 or not.**
[1]	**ORDEN**	**Enable Ordering Execution for Descriptor List** The GDMA_DADRx [ORDEN] determine which the next descriptor address will be fetched. If [ORDEN] is set, the GDMA controller fetches the next descriptor from Current GDMA_DADRx [Descriptor Address] + 16 bytes. If this bit is cleared, GDMA fetches the next descriptor address from the current GDMA_DADRx [Descriptor Address] register. GDMA_DADRx [ORDEN] is relevant only for descriptor-fetch function (GDMA_DADRx [NON_DSPTRMODE] = 0). 0 = Disable descriptor ordering. Fetch the next descriptor from register GDMA_DDADRx [Descriptor Address]. 1 = Enable descriptor ordering.
[0]	**RESET**	**Reset Channel** 0 = Disable channel reset. 1 = Enable channel status reset and disable descriptor based function.

Channel 0/1 GDMA Internal Buffer Register (GDMA_INTBUF0/1)

Software can set the [17-16] bit of GDMA_INTCS to select channels and watch the value which has read from memory.

Register	Address	R/W	Description	Reset Value
GDMA_INTBUF0	0xB000_4080	R	GDMA Internal Buffer Word 0	0x0000_0000
GDMA_INTBUF1	0xB000_4084	R	GDMA Internal Buffer Word 1	0x0000_0000
GDMA_INTBUF2	0xB000_4088	R	GDMA Internal Buffer Word 2	0x0000_0000
GDMA_INTBUF3	0xB000_408C	R	GDMA Internal Buffer Word 3	0x0000_0000
GDMA_INTBUF4	0xB000_4090	R	GDMA Internal Buffer Word 4	0x0000_0000
GDMA_INTBUF5	0xB000_4094	R	GDMA Internal Buffer Word 5	0x0000_0000
GDMA_INTBUF6	0xB000_4098	R	GDMA Internal Buffer Word 6	0x0000_0000
GDMA_INTBUF7	0xB000_409C	R	GDMA Internal Buffer Word 7	0x0000_0000

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

Bits	Descriptions	
[31:0]	DATA_BUFFER	**Internal Buffer Register** Each channel has its own internal buffer from Word 0 to Word 7. The [17-16] bit of GDMA_INTCS will determine the values of channels mapping to GDMA_INTBUF0~7. NOTE: The GDMA_INTBUF0~7 are available when burst mode used, otherwise, only the GDMA_INTBUF0 available.

Channel 0/1 GDMA Interrupt Control and Status Register (GDMA_INTCS)

Register	Address	R/W	Description	Reset Value
GDMA_INTCS	0xB000_40A0	R/W	Interrupt Control and Status Register (2 Channels)	0x0000_0000

31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED						BUF_RD_SEL	
15	14	13	12	11	10	9	8
RESERVED				TERR1F	TC1F	TERR0F	TC0F
7	6	5	4	3	2	1	0
RESERVED				TERR1EN	TC1EN	TERR0EN	TC0EN

Bits	Descriptions	
[17:16]	**BUF_RD_SEL**	**Internal Buffer Read Select** 00 = Read Internal Buffer for Channel 0 01 = Read Internal Buffer for Channel 1 10 = RESERVED 11 = RESERVED
[11]	**TERR1F**	**Channel 1 Transfer Error** O = No error occurs 1 = Hardware sets this bit on a GDMA transfer failure This bit will be cleared when write logic 1. Transfer error will generate GDMA interrupt
[10]	**TC1F**	**Channel 1 Terminal Count** 0 = Channel does not expire 1 = Channel expires; this bit is set only by GDMA hardware, and clear by software to write logic 1. TC1 is the GDMA interrupt flag. TC1 or GDMATERR1 will generate interrupt
[9]	**TERR0F**	**Channel 0 Transfer Error** O = No error occurs 1 = Hardware sets this bit on a GDMA transfer failure This bit will be cleared when write logic 1. Transfer error will generate GDMA interrupt

[8]	**TC0F**	**Channel 0 Terminal Count** 0 = Channel does not expire 1 = Channel expires; this bit is set only by GDMA hardware, and clear by software to write logic 1. TC0 is the GDMA interrupt flag. TC0 or GDMATERR0 will generate interrupt
[3]	**TEER1EN**	**Channel 1 Interrupt Enable for Transfer Error** 0 = Disable Interrupt 1 = Enable Interrupt
[2]	**TC1EN**	**Channel 1 Interrupt Enable for Terminal Count** 0 = Disable Interrupt 1 = Enable Interrupt
[1]	**TEER0EN**	**Channel 0 Interrupt Enable for Transfer Error** 0 = Disable Interrupt 1 = Enable Interrupt
[0]	**TC0EN**	**Channel 0 Interrupt Enable for Terminal Count** 0 = Disable Interrupt 1 = Enable Interrupt

7.7 USB Host Controller (USBH)

The Universal Serial Bus (USB) is a fast, bi-directional, isochronous, low-cost, dynamically attachable serial interface standard intended for USB devices. The USB is a 4-wire serial cable bus that supports serial data exchange between a Host Controller and a network of peripheral devices. The attached peripherals share USB bandwidth through a host-scheduled, token-based protocol. Peripherals may be attached, configured, used, and detached, while the host and other peripherals continue operation (i.e. hot plug and unplug is supported).
A major design goal of the USB standard was to allow flexible, plug-and-play networks of USB devices. In any USB network, there will be only one host, but there can be many devices and hubs.

The USB Host Controller includes the following features

- Fully compliant with USB Revision 2.0 specification.
- Enhanced Host Controller Interface (EHCI) Revision 1.0 compatible.
- Open Host Controller Interface (OHCI) Revision 1.0 compatible.
- Supports high-speed (480Mbps), full-speed (12Mbps) and low-speed (1.5Mbps) USB devices.
- Supports Control, Bulk, Interrupt, Isochronous and Split transfers.
- Integrated a port routing logic to route full/low speed device to OHCI controller.
- Built-in DMA for real-time data transfer.

7.7.1 Register Mapping

Register	Offset	R/W	Description	Reset Value
Capability Registers (USBH_BA = 0xB000_5000)				
EHCVNR	0xB000_5000	R	EHCI Version Number Register	0x0095_0020
EHCSPR	0xB000_5004	R	EHCI Structural Parameters Register	0x0000_0012
EHCCPR	0xB000_5008	R	EHCI Capability Parameters Register	0x0000_0000
Operational Registers				
UCMDR	0xB000_5020	R/W	USB Command Register	0x0008_0000
USTSR	0xB000_5024	R/W	USB Status Register	0x0000_1004
UIENR	0xB000_5028	R/W	USB Interrupt Enable Register	0x0000_0000
UFINDR	0xB000_502C	R/W	USB Frame Index Register	0x0000_0000
UPFLBAR	0xB000_5034	R/W	USB Periodic Frame List Base Address Register	0x0000_0000
UCALAR	0xB000_5038	R/W	USB Current Asynchronous List Address Register	0x0000_0000
UASSTR	0xB000_503C	R/W	USB Asynchronous Schedule Sleep Timer Register	0x0000_0BD6
UCFGR	0xB000_5060	R/W	USB Configure Flag Register	0x0000_0000
UPSCR0	0xB000_5064	R/W	USB Port 0 Status and Control Register	0x0000_2000
UPSCR1	0xB000_5068	R/W	USB Port 1 Status and Control Register	0x0000_2000
Miscellaneous Registers				
USBPCR0	0xB000_50C4	R/W	USB PHY 0 Control Register	0x0000_0060
USBPCR1	0xB000_50C8	R/W	USB PHY 1 Control Register	0x0000_0020
OHCI Registers (USBO_BA = 0xB000_7000)				
HcRev	0xB000_7000	R	Host Controller Revision Register	0x0000_0010
HcControl	0xB000_7004	R/W	Host Controller Control Register	0x0000_0000
HcComSts	0xB000_7008	R/W	Host Controller Command Status Register	0x0000_0000
HcIntSts	0xB000_700C	R/W	Host Controller Interrupt Status Register	0x0000_0000
HcIntEn	0xB000_7010	R/W	Host Controller Interrupt Enable Register	0x0000_0000
HcIntDis	0xB000_7014	R/W	Host Controller Interrupt Disable Register	0x0000_0000
HcHCCA	0xB000_7018	R/W	Host Controller Communication Area Register	0x0000_0000
HcPerCED	0xB000_701C	R/W	Host Controller Period Current ED Register	0x0000_0000
HcCtrHED	0xB000_7020	R/W	Host Controller Control Head ED Register	0x0000_0000
HcCtrCED	0xB000_7024	R/W	Host Controller Control Current ED Register	0x0000_0000
HcBlkHED	0xB000_7028	R/W	Host Controller Bulk Head ED Register	0x0000_0000
HcBlkCED	0xB000_702C	R/W	Host Controller Bulk Current ED Register	0x0000_0000
HcDoneH	0xB000_7030	R/W	Host Controller Done Head Register	0x0000_0000
HcFmIntv	0xB000_7034	R/W	Host Controller Frame Interval Register	0x0000_2EDF
HcFmRem	0xB000_7038	R	Host Controller Frame Remaining Register	0x0000_0000

HcFNum	0xB000_703C	R	Host Controller Frame Number Register	0x0000_0000
HcPerSt	0xB000_7040	R/W	Host Controller Periodic Start Register	0x0000_0000
HcLSTH	0xB000_7044	R/W	Host Controller Low Speed Threshold Register	0x0000_0628
HcRhDeA	0xB000_7048	R/W	Host Controller Root Hub Descriptor A Register	0x0100_0002
HcRhDeB	0xB000_704C	R/W	Host Controller Root Hub Descriptor B Register	0x0000_0000
HcRhSts	0xB000_7050	R/W	Host Controller Root Hub Status Register	0x0000_0000
HcRhPrt1	0xB000_7054	R/W	Host Controller Root Hub Port Status [1]	0x0000_0000
HcRhPrt2	0xB000_7058	R/W	Host Controller Root Hub Port Status [2]	0x0000_0000
OHCI USB Configuration Register				
OpModEn	0xB000_7204	R/W	USB Operational Mode Enable Register	0X0000_0000

7.7.2 Register Details

EHCI Version Number Register (EHCVNR)

Register	Address	R/W	Description	Reset Value
EHCVNR	0xB000_5000	R	EHCI Version Number Register	0x0095_0020

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

Bits	Descriptions	
[31:16]	**Version**	**Host Controller Interface Version Number** This is a two-byte register containing a BCD encoding of the EHCI revision number supported by this host controller. The most significant byte of this register represents a major revision and the least significant byte is the minor revision.
[7:0]	**CR_Length**	**Capability Registers Length** This register is used as an offset to add to register base to find the beginning of the Operational Register Space.

EHCI Structural Parameters Register (EHCSPR)

Register	Address	R/W	Description	Reset Value
EHCSPR	0xB000_5004	R	EHCI Structural Parameters Register	0x0000_0012

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

Bits	Descriptions	
[15:12]	**N_CC**	**Number of Companion Controller** This field indicates the number of companion controllers associated with this USB 2.0 host controller. A zero in this field indicates there are no companion host controllers. Port-ownership hand-off is not supported. Only high-speed devices are supported on the host controller root ports. A value larger than zero in this field indicates there are companion USB 1.1 host controller(s). Port-ownership hand-offs are supported. High, Full- and Low-speed devices are supported on the host controller root ports.
[11:8]	**N_PCC**	**Number of Ports per Companion Controller** This field indicates the number of ports supported per companion host controller. It is used to indicate the port routing configuration to system software. For example, if N_PORTS has a value of 6 and N_CC has a value of 2 then N_PCC could have a value of 3. The convention is that the first N_PCC ports are assumed to be routed to companion controller 1, the next N_PCC ports to companion controller 2, etc. In the previous example, the N_PCC could have been 4, where the first 4 are routed to companion controller 1 and the last two are routed to companion controller 2. The number in this field must be consistent with N_PORTS and N_CC.
[4]	**PPC**	**Port Power Control** This field indicates whether the host controller implementation includes port power control. A one in this bit indicates the ports have port power switches. A zero in this bit indicates the port do not have port power stitches. The value of this field affects the functionality of the *Port Power* field in each port status and control register.

[3:0]	**N_PORTS**	**Number of Physical Downstream Ports** This field specifies the number of physical downstream ports implemented on this host controller. The value of this field determines how many port registers are addressable in the Operational Register Space. Valid values are in the range of 1H to FH. A zero in this field is undefined.

EHCI Capability Parameters Register (EHCCPR)

Register	Address	R/W	Description	Reset Value
EHCCPR	0xB000_5008	R	EHCI Capability Parameters Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
EECP							
7	6	5	4	3	2	1	0
ISO_SCH_TH				Reserved	ASPC	PFList	64B

Bits	Descriptions	
[15:8]	**EECP**	**EHCI Extended Capabilities Pointer (EECP)** 8'h0: No extended capabilities are implemented.
[7:4]	**ISO_SCH_TH**	**Isochronous Scheduling Threshold**
[2]	**ASPC**	**Asynchronous Schedule Park Capability** 1'b0: This EHCI host controller doesn't support park feature of high-speed queue heads in the Asynchronous Schedule.
[1]	**PFList**	**Programmable Frame List Flag** 1'b0: System software must use a frame list length of 1024 elements with this EHCI host controller.
[0]	**64B**	**64-bit Addressing Capability** 1'b0: Data structure using 32-bit address memory pointers.

USB Command Register (UCMDR)

Register	Address	R/W	Description	Reset Value
UCMDR	0xB000_5020	R/W	USB Command Register	0x0008_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
INT_TH_CTL							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved	AsynADB	ASEN	PSEN	FLSize		HCRESET	RunStop

Bits	Descriptions	
[23:16]	**INT_TH_CTL**	**Interrupt Threshold Control (R/W)** This field is used by system software to select the maximum rate at which the host controller will issue interrupts. The only valid values are defined below. If software writes an invalid value to this register, the results are undefined. **Value Maximum Interrupt Interval** 00h Reserved 01h 1 micro-frame 02h 2 micro-frames 04h 4 micro-frames 08h 8 micro-frames (default, equates to 1 ms) 10h 16 micro-frames (2 ms) 20h 32 micro-frames (4 ms) 40h 64 micro-frames (8 ms) Any other value in this register yields undefined results. Software modifications to this bit while HCHalted bit is equal to zero results in undefined behavior.
[6]	**AsynADB**	**Interrupt on Async Advance Doorbell (R/W)** 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. Software must write a 1 to this bit to ring the doorbell. When the host controller has evicted all appropriate cached schedule state, it sets the *Interrupt on Async Advance* status bit in the USBSTS register. If the *Interrupt on Async Advance Enable* bit in the USBINTR register is a one then the host controller will assert an interrupt at the next interrupt threshold. The host controller sets this bit to a zero after it has set the *Interrupt on Async Advance* status bit in the USBSTS register to a one. Software should not write a one to this bit when the asynchronous schedule is disabled. Doing so will yield undefined results.

[5]	**ASEN**	**Asynchronous Schedule Enable (R/W)** This bit controls whether the host controller skips processing the Asynchronous Schedule. Values mean: 0b Do not process the Asynchronous Schedule 1b Use the ASYNCLISTADDR register to access the Asynchronous Schedule
[4]	**PSEN**	**Periodic Schedule Enable (R/W)** This bit controls whether the host controller skips processing the Periodic Schedule. Values mean: 0b Do not process the Periodic Schedule 1b Use the PERIODICLISTBASE register to access the Periodic Schedule
[3:2]	**FLSize**	**Frame List Size (R/W or RO)** This field is R/W only if *Programmable Frame List Flag* in the HCCPARAMS registers is set to a one. This field specifies the size of the frame list. The size the frame list controls which bits in the Frame Index Register should be used for the Frame List Current index. Values mean: 00b 1024 elements (4096 bytes) Default value 01b 512 elements (2048 bytes) 10b 256 elements (1024 bytes) – for resource-constrained environment 11b **Reserved**
[1]	**HCRESET**	**Host Controller Reset (HCRESET) (R/W)** This control bit is used by software to reset the host controller. The effects of this on Root Hub registers are similar to a Chip Hardware Reset. When software writes a one to this bit, the Host Controller resets its internal pipelines, timers, counters, state machines, etc. to their initial value. Any transaction currently in progress on USB is immediately terminated. A USB reset is not driven on downstream ports. All operational registers, including port registers and port state machines are set to their initial values. Port ownership reverts to the companion host controller(s), with the side effects. Software must reinitialize the host controller in order to return the host controller to an operational state. This bit is set to zero by the Host Controller when the reset process is complete. Software cannot terminate the reset process early by writing a zero to this register. Software should not set this bit to a one when the HCHalted bit in the USBSTS register is a zero. Attempting to reset an actively running host controller will result in undefined behavior.
[0]	**RunStop**	**Run/Stop (R/W)** 1=Run. 0=Stop. When set to a 1, the Host Controller proceeds with execution of the schedule. The Host Controller continues execution as long as this bit is set to a 1. When this bit is set to 0, the Host Controller completes the current and any actively pipelined transactions on the USB and then halts. The Host Controller must halt within 16 micro-frames after software clears the Run bit. The HC Halted bit in the status register indicates when the Host Controller has finished its pending pipelined transactions and has entered the stopped state. Software must not write a one to this field unless the host controller is in the Halted state (i.e. HCHalted in the USBSTS register is a one). Doing so will yield undefined results.

USB Status Register (USTSR)

Register	Address	R/W	Description	Reset Value
USTSR	0xB000_5024	R/W	USB Status Register	0x0000_1000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
ASSTS	PSSTS	RECLA	HCHalted	Reserved			
7	6	5	4	3	2	1	0
Reserved		IntAsynA	HSERR	FLROVER	PortCHG	UERRINT	USBINT

Bits	Descriptions	
[15]	**ASSTS**	**Asynchronous Schedule Status (RO)** The bit reports the current real status of the Asynchronous Schedule. If this bit is a zero then the status of them Asynchronous Schedule is disabled. If this bit is a one then the status of the Asynchronous Schedule is enabled. The Host Controller is not required to i*mmediately* disable or enable the Asynchronous Schedule when software transitions the *Asynchronous Schedule Enable* bit in the USBCMD register. When this bit and the *Asynchronous Schedule Enable* bit are the same value, the Asynchronous Schedule is either r enabled (1) or disabled (0).
[14]	**PSSTS**	**Periodic Schedule Status (RO)** The bit reports the current real status of the Periodic Schedule. If this bit is a zero then the status of the Periodic Schedule is disabled. If this bit is a one then the status of the Periodic Schedule is enabled. The Host Controller is not required to *immediately* disable or enable the Periodic Schedule when software transitions the *Periodic Schedule Enable* bit in the USBCMD register. When this bit and the *Periodic Schedule Enable* bit are the same value, the Periodic Schedule is either enabled (1) or disabled (0).
[13]	**RECLA**	**Reclamation (RO)** This is a read-only status bit, which is used to detect an empty asynchronous schedule.
[12]	**HCHalted**	**HCHalted (RO)** 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 (e.g. internal error).

[5]	**IntAsynA**	**Interrupt on Async Advance (R/WC)** 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 USBCMD register. This status bit indicates the assertion of that interrupt source.
[4]	**HSERR**	**Host System Error (R/WC)** The Host Controller sets this bit to 1 when a serious error occurs during a host system access involving the Host Controller module.
[3]	**FLROVER**	**Frame List Rollover (R/WC)** The Host Controller sets this bit to a one when the *Frame List Index* rolls over from its maximum value to zero. The exact value at which the rollover occurs depends on the frame list size. For example, if the frame list size (as programmed in the *Frame List Size* field of the USBCMD register) is 1024, the *Frame Index Register* rolls over every time FRINDEX [13] toggles. Similarly, if the size is 512, the Host Controller sets this bit to a one every time FRINDEX [12] toggles.
[2]	**PortCHG**	**Port Change Detect (R/WC)** 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 transition 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. This bit is allowed to be maintained in the Auxiliary power well. Alternatively, it is also acceptable that on a D3 to D0 transition of the EHCI HC device, this bit is loaded with the OR of all of the PORTSC change bits (including: Force port resume, over-current change, enable/disable change and connect status change).
[1]	**UERRINT**	**USB Error Interrupt (USBERRINT) (R/WC)** The Host Controller sets this bit to 1 when completion of a USB transaction results in an error condition (e.g., error counter underflow). If the TD on which the error interrupt occurred also had its IOC bit set, both this bit and USBINT bit are set.
[0]	**USBINT**	**USB Interrupt (USBINT) (R/WC)** The Host Controller sets this bit to 1 on the completion of a USB transaction, which results in the retirement of a Transfer Descriptor that had its IOC bit set. The Host Controller also sets this bit to 1 when a short packet is detected (actual number of bytes received was less than the expected number of bytes).

USB Interrupt Enable Register (UIENR)

Register	Address	R/W	Description	Reset Value
UIENR	0xB000_5028	R/W	USB Interrupt Enable Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved		AsynAEN	HSERREN	FLREN	PCHGEN	UERREN	USBIEN

Bits	Descriptions	
[5]	**AsynAEN**	**Interrupt on Async Advance Enable** When this bit is a one, and the *Interrupt on Async Advance* bit in the USBSTS register is a one, the host controller will issue an interrupt at the next interrupt threshold. The interrupt is acknowledged by software clearing the *Interrupt on Async Advance* bit.
[4]	**HSERREN**	**Host System Error Enable** When this bit is a one, and the Host System Error Status bit in the USBSTS register is a one, the host controller will issue an interrupt. The interrupt is acknowledged by software clearing the Host System Error bit.
[3]	**FLREN**	**Frame List Rollover Enable** When this bit is a one, and the Frame List Rollover bit in the USBSTS register is a one, the host controller will issue an interrupt. The interrupt is acknowledged by software clearing the Frame List Rollover bit.
[2]	**PCHGEN**	**Port Change Interrupt Enable** When this bit is a one, and the Port Change Detect bit in the USBSTS register is a one, the host controller will issue an interrupt. The interrupt is acknowledged by software clearing the Port Change Detect bit.
[1]	**UERREN**	**USB Error Interrupt Enable** When this bit is a one, and the USBERRINT bit in the USBSTS register is a one, the host t controller will issue an interrupt at the next interrupt threshold. The interrupt is acknowledged by software clearing the USBERRINT bit.
[0]	**USBIEN**	**USB Interrupt Enable** When this bit is a one, and the USBINT bit in the USBSTS register is a one, the host controller will issue an interrupt at the next interrupt threshold. The interrupt is acknowledged by software clearing the USBINT bit.

USB Frame Index Register (UFINDR)

Register	Address	R/W	Description	Reset Value
UFINDR	0xB000_502C	R/W	USB Frame Index Register	0x0000_0000

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

Bits	Descriptions	
[13:0]	**FrameIND**	**Frame Index** The value in this register increment at the end of each time frame (e.g. micro-frame). Bits [N: 3] are used for the Frame List current index. This means that each location of the frame list is accessed 8 times (frames or micro-frames) before moving to the next index. The following illustrates values of N based on the value of the Frame List Size field in the USBCMD register.

USB Periodic Frame List Base Address Register (UPFLBAR)

Register	Address	R/W	Description	Reset Value
UPFLBAR	0xB000_5034	R/W	USB Periodic Frame List Base Address Register	0x0000_0000

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

Bits	Descriptions	
[31:12]	**BADDR**	**Base Address (Low)** These bits correspond to memory address signals [31:12], respectively.

USB Current Asynchronous List Address Register (UCALAR)

Register	Address	R/W	Description	Reset Value
UCALAR	0xB000_5038	R/W	USB Current Asynchronous List Address Register	0x0000_0000

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

Bits	Descriptions	
[31:5]	**LPL**	**Link Pointer Low (LPL)** These bits correspond to memory address signals [31:5], respectively. This field may only reference a Queue Head (QH).

USB Asynchronous Schedule Sleep Timer Register

Register	Address	R/W	Description	Reset Value
UASSTR	0xB000_503C	R/W	USB Asynchronous Schedule Sleep Timer Register	0x0000_0BD6

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

Bits	Descriptions	
[11:0]	**ASSTMR**	**Asynchronous Schedule Sleep Timer** This field defines the AsyncSchedSleepTime of EHCI spec. The asynchronous schedule sleep timer is used to control how often the host controller fetches asynchronous schedule list from system memory while the asynchronous schedule is empty. The default value of this timer is 12'hBD6. Because this timer is implemented in UTMI clock (30MHz) domain, the default sleeping time will be about 100us.

USB Configure Flag Register (UCFGR)

Register	Address	R/W	Description	Reset Value
UCFGR	0xB000_5060	R/W	USB Configure Flag Register	0x0000_0000

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

Bits	Descriptions	
[0]	**CF**	**Configure Flag (CF)** Host software sets this bit as the last action in its process of configuring the Host Controller. This bit controls the default port-routing control logic. Bit values and side-effects are listed below. 0b Port routing control logic default-routes each port to an implementation dependent classic host controller. 1b Port routing control logic default-routes all ports to this host controller.

USB Port 0 Status and Control Register (UPSCR0)

Register	Address	R/W	Description	Reset Value
UPSCR0	0xB000_5064	R/W	USB Port 0 Status and Control Register	0x0000_2000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved		PO	PP	LStatus		Reserved	PRST
7	6	5	4	3	2	1	0
Suspend	FPResum	OCCHG	OCACT	PENCHG	PEN	CSCHG	CSTS

Bits	Descriptions	
[13]	**PO**	**Port Owner (R/W)** 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. System software uses this field to release ownership of the port to a selected host controller (in the event that the attached device is not a high-speed device). Software writes a one to this bit when the attached device is not a high-speed device. A one in this bit means that a companion host controller owns and controls the port.
[12]	**PP**	**Port Power (PP)** Host controller has port power control switches. This bit represents the Current setting of the switch (0 = off, 1 = on). When power is not available on a port (i.e. *PP* equals a 0), the port is nonfunctional and will not report attaches, detaches, etc. When an over-current condition is detected on a powered port and *PPC* is a one, the *PP* bit in each affected port may be transitioned by the host controller from a 1 to 0 (removing power from the port).

[11:10]	**LStatus**	**Line Status (RO)** These bits reflect the current logical levels of the D+ (bit 11) and D- (bit 10) signal lines. These bits are used for detection of low-speed USB devices prior to the port reset and enable sequence. This field is valid only when the port enable bit is zero and the current connect status bit is set to a one. The encoding of the bits are: **Bits[11:10] USB State Interpretation** 00b SE0 Not Low-speed device, perform EHCI reset 10b J-state Not Low-speed device, perform EHCI reset 01b K-state Low-speed device, release ownership of port 11b Undefined Not Low-speed device, perform EHCI reset. This value of this field is undefined if Port Power is zero.
[8]	**PRST**	**Port Reset (R/W)** 1=Port is in Reset. 0=Port is not in Reset. Default = 0. When software writes a one to this bit (from a zero), the bus reset sequence as defined in the USB Specification Revision 2.0 is started. Software writes a zero to this bit to terminate the bus reset sequence. Software must keep this bit at a one long enough to ensure the reset sequence, as specified in the USB Specification Revision 2.0, completes. Note: when software writes this bit to a one, it must also write a zero to the Port Enable bit. Note that when software writes a zero to this bit there may be a delay before the bit status changes to a zero. The bit status will not read as a zero until after the reset has completed. If the port is in high-speed mode after reset is complete, the host controller will automatically enable this port (e.g. set the Port Enable bit to a one). A host controller must terminate the reset and stabilize the state of the port within 2 milliseconds of software transitioning this bit from a one to a zero. For example: if the port detects that the attached device is high-speed during reset, then the host controller must have the port in the enabled state within 2ms of software writing this bit to a zero. The HCHalted bit in the USBSTS register should be a zero before software attempts to use this bit. The host controller may hold Port Reset asserted to a one when the HCHalted bit is a one. This field is zero if Port Power is zero.

[7]	**Suspend**	**Suspend (R/W)** 1=Port in suspend state. 0=Port not in suspend state. Default = 0. Port Enabled Bit and Suspend bit of this register define the port states as follows: **Bits [Port Enabled, Suspend] Port State** 0X Disable 10 Enable 11 Suspend When in suspend state, downstream propagation of data is blocked on this port, except for port reset. The blocking occurs at the end of the current transaction, if a transaction was in progress when this bit was written to 1. In the suspend state, the port is sensitive to resume detection. Note that the bit status does not change until the port is suspended and that there may be a delay in suspending a port if there is a transaction currently in progress on the USB. 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 bit to a zero (from a one). Software sets the Port Reset bit to a one (from a zero). If host software sets this bit to a one when the port is not enabled (i.e. Port enabled bit is a zero) the results are undefined. This field is zero if Port Power is zero.
[6]	**FPResum**	**Force Port Resume (R/W)** 1= Resume detected/driven on port. 0=No resume (Kstate) detected/driven on port. Default = 0. This functionality defined for manipulating this bit depends on the value of the Suspend bit. For example, if the port is not suspended (Suspend and Enabled bits are a one) and software transitions this bit to a one, then the effects on the bus are undefined. Software sets this bit to a 1 to drive 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. When this bit transitions to a one because a J-to-K transition is detected, the Port Change Detect bit in the USBSTS register is also set to a one. If software sets this bit to a one, the host controller must not set the Port Change Detect bit. Note that when the EHCI controller owns the port, the resume sequence follows the defined sequence documented in the USB Specification Revision 2.0. The resume signaling (Full-speed 'K') is driven on the port as long as this bit remains a one. Software must appropriately time the Resume and set this bit to a zero when the appropriate amount of time has elapsed. Writing a zero (from one) causes the port to return to high-speed mode (forcing the bus below the port into a high-speed idle). This bit will remain a one until the port has switched to the high-speed idle. The host controller must complete this transition within 2 milliseconds of software setting this bit to a zero. This field is zero if Port Power is zero.
[5]	**OCCHG**	**Over-current Change (R/WC)** Default = 0. 1=This bit gets set to a one when there is a change to Over-current Active. Software clears this bit by writing a one to this bit position.

[4]	**OCACT**	**Over-current Active (RO)** Default = 0. 1=This port currently has an over current condition. 0=This port does not have an over-current condition. This bit will automatically transition from a one to a zero when the over current condition is removed.
[3]	**PENCHG**	**Port Enable / Disable Change (R / WC)** 1=Port enabled/disabled status has changed. 0=No change. Default = 0. 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 point (See Chapter 11 of the USB Specification for the definition of a Port Error). Software clears this bit by writing a 1 to it. This field is zero if Port Power is zero.
[2]	**PEN**	**Port Enabled / Disabled (R / W)** 1=Enable. 0=Disable. Default = 0. 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. The host controller will only set this bit to a one when the reset sequence determines that the attached device is a high-speed device. Ports can be disabled by either a fault condition (disconnect event or other fault condition) or by host software. Note that the bit status does not change until the port state actually changes. There may be a delay in disabling or enabling a port due to other host controller and bus events. When the port is disabled (0b) downstream propagation of data is blocked on this port, except for reset. This field is zero if Port Power is zero.
[1]	**CSCHG**	**Connect Status Change (R / W)** 1=Change in Current Connect Status. 0=No change. Default = 0. Indicates a change has occurred in the port's Current Connect Status. The host controller sets this bit for all changes to the port device connect status, even if system software has not cleared an existing connect status change. For example, the insertion status changes twice before system software has cleared the changed condition, hub hardware will be "setting" an already-set bit (i.e., the bit will remain set).Software sets this bit to 0 by writing a 1 to it. This field is zero if Port Power is zero.
[0]	**CSTS**	**Current Connect Status (RO)** 1=Device is present on port. 0=No device is present. Default = 0. This value reflects the current state of the port, and may not correspond directly to the event that caused the Connect Status Change bit (Bit 1) to be set. This field is zero if Port Power is zero.

USB Port 1 Status and Control Register (UPSCR1)

Register	Address	R/W	Description	Reset Value
UPSCR1	0xB000_5068	R/W	USB Port 1 Status and Control Register	0x0000_2000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved				Reserved			
15	14	13	12	11	10	9	8
Reserved		PO	PP	LStatus		Reserved	PRST
7	6	5	4	3	2	1	0
Suspend	FPResum	OCCHG	OCACT	PENCHG	PEN	CSCHG	CSTS

Bits	Descriptions	
[13]	**PO**	**Port Owner (R/W)** 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. System software uses this field to release ownership of the port to a selected host controller (in the event that the attached device is not a high-speed device). Software writes a one to this bit when the attached device is not a high-speed device. A one in this bit means that a companion host controller owns and controls the port.
[12]	**PP**	**Port Power (PP)** Host controller has port power control switches. This bit represents the Current setting of the switch (0 = off, 1 = on). When power is not available on a port (i.e. *PP* equals a 0), the port is nonfunctional and will not report attaches, detaches, etc. When an over-current condition is detected on a powered port and *PPC* is a one, the *PP* bit in each affected port may be transitioned by the host controller from a 1 to 0 (removing power from the port).

[11:10]	**LStatus**	**Line Status (RO)** These bits reflect the current logical levels of the D+ (bit 11) and D- (bit 10) signal lines. These bits are used for detection of low-speed USB devices prior to the port reset and enable sequence. This field is valid only when the port enable bit is zero and the current connect status bit is set to a one. The encoding of the bits are: **Bits[11:10] USB State Interpretation** 00b SE0 Not Low-speed device, perform EHCI reset 10b J-state Not Low-speed device, perform EHCI reset 01b K-state Low-speed device, release ownership of port 11b Undefined Not Low-speed device, perform EHCI reset. This value of this field is undefined if Port Power is zero.
[8]	**PRST**	**Port Reset (R/W)** 1=Port is in Reset. 0=Port is not in Reset. Default = 0. When software writes a one to this bit (from a zero), the bus reset sequence as defined in the USB Specification Revision 2.0 is started. Software writes a zero to this bit to terminate the bus reset sequence. Software must keep this bit at a one long enough to ensure the reset sequence, as specified in the USB Specification Revision 2.0, completes. Note: when software writes this bit to a one, it must also write a zero to the Port Enable bit. Note that when software writes a zero to this bit there may be a delay before the bit status changes to a zero. The bit status will not read as a zero until after the reset has completed. If the port is in high-speed mode after reset is complete, the host controller will automatically enable this port (e.g. set the Port Enable bit to a one). A host controller must terminate the reset and stabilize the state of the port within 2 milliseconds of software transitioning this bit from a one to a zero. For example: if the port detects that the attached device is high-speed during reset, then the host controller must have the port in the enabled state within 2ms of software writing this bit to a zero. The HCHalted bit in the USBSTS register should be a zero before software attempts to use this bit. The host controller may hold Port Reset asserted to a one when the HCHalted bit is a one. This field is zero if Port Power is zero.

[7]	**Suspend**	**Suspend (R/W)** 1=Port in suspend state. 0=Port not in suspend state. Default = 0. Port Enabled Bit and Suspend bit of this register define the port states as follows: **Bits [Port Enabled, Suspend] Port State** 0X Disable 10 Enable 11 Suspend When in suspend state, downstream propagation of data is blocked on this port, except for port reset. The blocking occurs at the end of the current transaction, if a transaction was in progress when this bit was written to 1. In the suspend state, the port is sensitive to resume detection. Note that the bit status does not change until the port is suspended and that there may be a delay in suspending a port if there is a transaction currently in progress on the USB. 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 bit to a zero (from a one). Software sets the Port Reset bit to a one (from a zero). If host software sets this bit to a one when the port is not enabled (i.e. Port enabled bit is a zero) the results are undefined. This field is zero if Port Power is zero.
[6]	**FPResum**	**Force Port Resume (R/W)** 1= Resume detected/driven on port. 0=No resume (Kstate) detected/driven on port. Default = 0. This functionality defined for manipulating this bit depends on the value of the Suspend bit. For example, if the port is not suspended (Suspend and Enabled bits are a one) and software transitions this bit to a one, then the effects on the bus are undefined. Software sets this bit to a 1 to drive 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. When this bit transitions to a one because a J-to-K transition is detected, the Port Change Detect bit in the USBSTS register is also set to a one. If software sets this bit to a one, the host controller must not set the Port Change Detect bit. Note that when the EHCI controller owns the port, the resume sequence follows the defined sequence documented in the USB Specification Revision 2.0. The resume signaling (Full-speed 'K') is driven on the port as long as this bit remains a one. Software must appropriately time the Resume and set this bit to a zero when the appropriate amount of time has elapsed. Writing a zero (from one) causes the port to return to high-speed mode (forcing the bus below the port into a high-speed idle). This bit will remain a one until the port has switched to the high-speed idle. The host controller must complete this transition within 2 milliseconds of software setting this bit to a zero. This field is zero if Port Power is zero.
[5]	**OCCHG**	**Over-current Change (R/WC)** Default = 0. 1=This bit gets set to a one when there is a change to Over-current Active. Software clears this bit by writing a one to this bit position.

[4]	**OCACT**	**Over-current Active (RO)** Default = 0. 1=This port currently has an over current condition. 0=This port does not have an over-current condition. This bit will automatically transition from a one to a zero when the over current condition is removed.
[3]	**PENCHG**	**Port Enable/Disable Change (R/WC)** 1=Port enabled/disabled status has changed. 0=No change. Default = 0. 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 point (See Chapter 11 of the USB Specification for the definition of a Port Error). Software clears this bit by writing a 1 to it. This field is zero if Port Power is zero.
[2]	**PEN**	**Port Enabled/Disabled (R/W)** 1=Enable. 0=Disable. Default = 0. 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. The host controller will only set this bit to a one when the reset sequence determines that the attached device is a high-speed device. Ports can be disabled by either a fault condition (disconnect event or other fault condition) or by host software. Note that the bit status does not change until the port state actually changes. There may be a delay in disabling or enabling a port due to other host controller and bus events. When the port is disabled (0b) downstream propagation of data is blocked on this port, except for reset. This field is zero if Port Power is zero.
[1]	**CSCHG**	**Connect Status Change (R/W)** 1=Change in Current Connect Status. 0=No change. Default = 0. Indicates a change has occurred in the port's Current Connect Status. The host controller sets this bit for all changes to the port device connect status, even if system software has not cleared an existing connect status change. For example, the insertion status changes twice before system software has cleared the changed condition, hub hardware will be "setting" an already-set bit (i.e., the bit will remain set).Software sets this bit to 0 by writing a 1 to it. This field is zero if Port Power is zero.
[0]	**CSTS**	**Current Connect Status (RO)** 1=Device is present on port. 0=No device is present. Default = 0. This value reflects the current state of the port, and may not correspond directly to the event that caused the Connect Status Change bit (Bit 1) to be set. This field is zero if Port Power is zero.

USB PHY 0 Control Register (USBPCR0)

Register	Address	R/W	Description	Reset Value
USBPCR0	0xB000_50C4	R/W	USB PHY 0 Control Register	0x0000_0060

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved				ClkValid	Reserved		Suspend
7	6	5	4	3	2	1	0
CLK48	REFCLK	CLK_SEL		XO_ON	SIDDQ	Reserved	

Bits	Descriptions	
[11]	**ClkValid**	**UTMI Clock Valid** This bit is a flag to indicate if the UTMI clock from USB 2.0 PHY is ready. S/W program must prevent to write other control registers before this UTMI clock valid flag is active. 1'b0: UTMI clock is not valid 1'b1: UTMI clock is valid
[8]	**Suspend**	**Suspend Assertion** This bit controls the suspend mode of USB PHY 0. While PHY was suspended, all circuits of PHY were powered down and outputs are tri-stated. This bit is 1'b0 in default. This means the USB PHY 0 is suspended in default. It is necessary to set this bit 1'b1 to make USB PHY 0 leave suspend mode before doing configuration of USB host. 1'b0: USB PHY 0 was suspended. 1'b1: USB PHY 0 was not suspended.
[7]	**CLK48**	**Digital Logic Clock Select** This bit controls the input signal clk48m_sel of USB PHY 0. This signal selects Power-Save mode. 1'b0: Non-Power-Save mode. The PLL and the phase interpolator are powered up. The digital logic uses a 480MHz clock. Non-Power-Save mode is valid in all modes and speeds of operation. 1'b1: Power-Save mode. The PLL and the phase interpolator are powered down. The digital logic uses a 48MHz clock. Power-Save mode is valid for only FS-Only operation. The CLK_SEL should be set to 2'b10 (48MHz) when this bit is high.
[6]	**REFCLK**	**Reference Clock Source Select** This bit has to set to 1.

[5:4]	**CKL_SEL**	**Reference Clock Frequency Select** This field has to set to 2'b10;
[3]	**XO_ON**	**Force XO Block on During a Suspend** This bit controls the input signal xo_on of USB PHY 0. 1'b0: If all ports are suspended, the XO block is powered up, and the test_clk48m signal is available. 1'b1: This bit is inactive, and the XO block is powered down when all ports are suspended.
[2]	**SIDDQ**	**IDDQ Test Enable** This bit controls the input signal siddq of USB PHY 0. This signal powers down all analog blocks. 1'b0: The analog blocks are in normal operation. 1'b1: The analog blocks are powered down.

USB PHY 1 Control Register (USBPCR1)

Register	Address	R/W	Description	Reset Value
USBPCR1	0xB000_50C8	R/W	USB PHY 1 Control Register	0x0000_0020

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved				XO_SEL	Reserved		Suspend
7	6	5	4	3	2	1	0
CLK48	REFCLK	CLK_SEL		XO_ON	SIDDQ	Reserved	

Bits	Descriptions	
[11]	**XO_SEL**	**Clock Select for XO Block** This bit defines the clock source of PHY1's XO block is from external clock or a crystal. 1'b0: The XO block uses a 48MHz external clock supplied from PHY 0 1'b1: The XO block uses the clock from a crystal
[8]	**Suspend**	**Suspend Assertion** This bit controls the suspend mode of USB PHY 1. While PHY was suspended, all circuits of PHY were powered down and outputs are tri-stated. This bit is 1'b0 in default. This means the USB PHY 1 is suspended in default. It is necessary to set this bit 1'b1 to make USB PHY 1 leave suspend mode before doing configuration of USB host. 1'b0: USB PHY 1 was suspended. 1'b1: USB PHY 1 was not suspended.
[7]	**CLK48**	**Digital Logic Clock Select** This bit controls the input signal clk48m_sel of USB PHY 1. This signal selects Power-Save mode. 1'b0: Non-Power-Save mode. The PLL and the phase interpolator are powered up. The digital logic uses a 480MHz clock. Non-Power-Save mode is valid in all modes and speeds of operation. 1'b1: Power-Save mode. The PLL and the phase interpolator are powered down. The digital logic uses a 48MHz clock. Power-Save mode is valid for only FS-Only operation. The CLK_SEL should be set to 2'b10 (48MHz) when this bit is high.
[6]	**REFCLK**	**Reference Clock Source Select** This bit has to set to 0.

[5:4]	**CKL_SEL**	**Reference Clock Frequency Select** This field has to set to 2'b10.
[3]	**XO_ON**	**Force XO Block on During a Suspend** This bit controls the input signal xo_on of USB PHY 1. 1'b0: If all ports are suspended, the XO block is powered up, and the test_clk48m signal is available. 1'b1: This bit is inactive, and the XO block is powered down when all ports are suspended.
[2]	**SIDDQ**	**IDDQ Test Enable** This bit controls the input signal siddq of USB PHY 1. This signal powers down all analog blocks. 1'b0: The analog blocks are in normal operation. 1'b1: The analog blocks are powered down.

Host Controller Revision Register (HcRev)

Register	Address	R/W	Description	Reset Value
HcRev	0xB000_7000	R	Host Controller Revision Register	0x0000_0010

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

Bits	Descriptions	
[7:0]	**Rev**	**Revision** Indicates the Open HCI Specification revision number implemented by the Hardware. Host Controller supports 1.0 specification. (X.Y = XYh)

Host Controller Control Register (HcControl)

Register	Address	R/W	Description	Reset Value
HcControl	0xB000_7004	R/W	Host Controller Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved					RWakeEn	RWake	IntRoute
7	6	5	4	3	2	1	0
HcFunc		BlkEn	CtrlEn	ISOEn	PeriEn	CtrlBlkRatio	

Bits	Descriptions	
[10]	**RWakeEn**	**Remote Wakeup Connected Enable** If a remote wakeup signal is supported, this bit enables that operation. Since there is no remote wakeup signal supported, this bit is ignored.
[9]	**RWake**	**Remote Wakeup Connected** This bit indicated whether the HC supports a remote wakeup signal. This implementation does not support any such signal. The bit is hard-coded to '0.'
[8]	**IntRoute**	**Interrupt Routing** This bit is used for interrupt routing: 0: Interrupts routed to normal interrupt mechanism (INT). 1: Interrupts routed to SMI.
[7:6]	**HcFunc**	**Host Controller Functional State** This field sets the Host Controller state. The Controller may force a state change from USBSUSPEND to USBRESUME after detecting resume signaling from a downstream port. States are: 00: USBRESET 01: USBRESUME 10: USBOPERATIONAL 11: USBSUSPEND
[5]	**BlkEn**	**Bulk List Enable** When set this bit enables processing of the Bulk list.
[4]	**CtrlEn**	**Control List Enable** When set this bit enables processing of the Control list.
[3]	**ISOEn**	**Isochronous List Enable** When clear, this bit disables the Isochronous List when the Periodic List is enabled (so Interrupt EDs may be serviced). While processing the Periodic List, the Host Controller will check this bit when it finds an isochronous ED.

[2]	**PeriEn**	**Periodic List Enable** When set, this bit enables processing of the Periodic (interrupt and isochronous) list. The Host Controller checks this bit prior to attempting any periodic transfers in a frame.
[1:0]	**CtrlBlkRatio**	**Control Bulk Service Ratio** Specifies the number of Control Endpoints serviced for every Bulk Endpoint. Encoding is N-1 where N is the number of Control Endpoints (i.e. '00' = 1 Control Endpoint; '11' = 3 Control Endpoints)

Host Controller Command Status Register (HcComSts)

Register	Address	R/W	Description	Reset Value
HcComSts	0xB000_7008	R/W	Host Controller Command Status Register	0x0000_0000

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

Bits	Descriptions	
[17:16]	**SchOverRun**	Schedule Overrun Count This field is increment every time the **SchedulingOverrun** bit in *HcInterruptStatus* is set. The count wraps from '11' to '00.'
[3]	**OCReq**	**Ownership Chang Request** When set by software, this bit sets the **OwnershipChange** field in *HcInterruptStatus*. The bit is cleared by software.
[2]	**BlkFill**	**Bulk List Filled** Set to indicate there is an active ED on the Bulk List. The bit may be set by either software or the Host Controller and cleared by the Host Controller each time it begins processing the head of the Bulk List.
[1]	**CtrlFill**	**Control List Filled** Set to indicate there is an active ED on the Control List. It may be set by either software or the Host Controller and cleared by the Host Controller each time it begins processing the head of the Control List.
[0]	**HCReset**	**Host Controller Reset** This bit is set to initiate the software reset. This bit is cleared by the Host Controller, upon completed of the reset operation.

Host Controller Interrupt Status Register (HcIntSts)

Register	Address	R/W	Description	Reset Value
HcIntSts	0xB000_700C	R/W	Host Controller Interrupt Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved	OC	Reserved					
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved	RHSC	FNOF	UnRecErr	Resume	SOF	WBDnHD	SchOR

Bits	Descriptions	
[30]	**OC**	**Ownership Change** This bit is set when the **OwnershipChangeRequest** bit of *HcCommandStatus* is set.
[6]	**RHSC**	**Root Hub Status Change** This bit is set when the content of *HcRhStatus* or the content of any *HcRhPortStatus* register has changed.
[5]	**FNOF**	**Frame Number Overflow** Set when bit 15 of **FrameNumber** changes value.
[4]	**UnRecErr**	**Unrecoverable Error** This event is not implemented and is hard-coded to '0.' Writes are ignored.
[3]	**Resume**	**Resume Detected** Set when Host Controller detects resume signaling on a downstream port.
[2]	**SOF**	**Start Of Frame** Set when the Frame Management block signals a 'Start of Frame' event.
[1]	**WBDnHD**	**Write Back Done Head** Set after the Host Controller has written *HcDoneHead* to *HccaDoneHead*.
[0]	**SchOR**	**Scheduling Overrun** Set when the List Processor determines a Schedule Overrun has occurred.

Host Controller Interrupt Enable Register (HcIntEn)

Register	Address	R/W	Description	Reset Value
HcIntEn	0xB000_7010	R/W	Host Controller Interrupt Enable Register	0x0000_0000

31	30	29	28	27	26	25	24
IntEn	OCEn	Reserved					
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved	RHSCEn	FNOFEn	URErrEn	ResuEn	SOFEn	WBDHEn	SchOREn

Bits	Descriptions	
[31]	**IntEn**	**Master Interrupt Enable** This bit is a global interrupt enable. A write of '1' allows interrupts to be enabled via the specific enable bits listed above.
[30]	**OCEn**	**Ownership Change Enable** 0: Ignore 1: Enables interrupt generation due to Ownership Change.
[6]	**RHSCEn**	**Root Hub Status Change Enable** 0: Ignore 1: Enables interrupt generation due to Root Hub Status Change.
[5]	**FNOFEn**	**Frame Number Overflow Enable** 0: Ignore 1: Enables interrupt generation due to Frame Number Overflow.
[4]	**URErrEn**	**Unrecoverable Error Enable** This event is not implemented. All writes to this bit are ignored.
[3]	**ResuEn**	**Resume Detected Enable** 0: Ignore 1: Enables interrupt generation due to Resume Detected.
[2]	**SOFEn**	**Start Of Frame Enable** 0: Ignore 1: Enables interrupt generation due to Start of Frame.
[1]	**WBDHEn**	**Write Back Done Head Enable** 0: Ignore 1: Enables interrupt generation due to Write-back Done Head.
[0]	**SchOREn**	**Scheduling Overrun Enable** 0: Ignore 1: Enables interrupt generation due to Scheduling Overrun.

Host Controller Interrupt Disable Register (HcIntDis)

Register	Address	R/W	Description	Reset Value
HcIntDis	0xB000_7014	R/W	Host Controller Interrupt Disable Register	0x0000_0000

31	30	29	28	27	26	25	24
IntDis	OCDis	Reserved					
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved	RHSCDis	FNOFDis	URErrDis	ResuDis	SOFDis	WBDHDis	SchORDis

Bits	Descriptions	
[31]	**IntDis**	**Master Interrupt Disable** Global interrupt disable. A write of '1' disables all interrupts.
[30]	**OCDis**	**Ownership Change Disable** 0: Ignore 1: Disables interrupt generation due to Ownership Change.
[6]	**RHSCDis**	**Root Hub Status Change Disable** 0: Ignore 1: Disables interrupt generation due to Root Hub Status Change.
[5]	**FNOFDis**	**Frame Number Overflow Disable** 0: Ignore 1: Disables interrupt generation due to Frame Number Overflow.
[4]	**URErrDis**	**Unrecoverable Error Disable** This event is not implemented. All writes to this bit are ignored.
[3]	**ResuDis**	**Resume Detected Disable** 0: Ignore 1: Disables interrupt generation due to Resume Detected.
[2]	**SOFDis**	**Start Of Frame Disable** 0: Ignore 1: Disables interrupt generation due to Start of Frame.
[1]	**WBDHDis**	**Write Back Done Head Disable** 0: Ignore 1: Disables interrupt generation due to Write-back Done Head.
[0]	**SchORDis**	**Scheduling Overrun Disable** 0: Ignore 1: Disables interrupt generation due to Scheduling Overrun.

Host Controller Communication Area Register (HcHCCA)

Register	Address	R/W	Description	Reset Value
HcHCCA	0xB000_7018	R/W	Host Controller Communication Area Register	0x0000_0000

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

Bits	Descriptions	
[31:7]	**HCCA**	**Host Controller Communication Area** Pointer to HCCA base address.

Host Controller Period Current ED Register (HcPerCED)

Register	Address	R/W	Description	Reset Value
HcPerCED	0xB000_701C	R/W	Host Controller Period Current ED Register	0x0000_0000

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

Bits	Descriptions	
[31:4]	**PeriCED**	**Periodic Current ED** Pointer to the current Periodic List ED.

Host Controller Control Head ED Register (HcCtrHED)

Register	Address	R/W	Description	Reset Value
HcCtrHED	0xB000_7020	R/W	Host Controller Control Head ED Register	0x0000_0000

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

Bits	Descriptions	
[31:4]	**CtrlHED**	**Control Head ED** Pointer to the Control List Head ED.

Host Controller Control Current ED Register (HcCtrCED)

Register	Address	R/W	Description	Reset Value
HcCtrCED	0xB000_7024	R/W	Host Controller Control Current ED Register	0x0000_0000

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

Bits	Descriptions	
[31:4]	**CtrlCED**	**Control Current Head ED** Pointer to the current Control List Head ED.

Host Controller Bulk Head ED Register (HcBlkHED)

Register	Address	R/W	Description	Reset Value
HcBlkHED	0xB000_7028	R/W	Host Controller Bulk Head ED Register	0x0000_0000

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

Bits	Descriptions	
[31:4]	**BlkHED**	**Bulk Head ED** Pointer to the Bulk List Head ED.

Host Controller Bulk Current Head ED Register (HcBlkCED)

Register	Address	R/W	Description	Reset Value
HcBlkCED	0xB000_702C	R/W	Host Controller Bulk Current ED Register	0x0000_0000

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

Bits	Descriptions	
[31:4]	**BlkCED**	**Bulk Current Head ED** Pointer to the current Bulk List Head ED.

Host Controller Done Head Register (HcDoneH)

Register	Address	R/W	Description	Reset Value
HcDoneH	0xB000_7030	R/W	Host Controller Done Head Register	0x0000_0000

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

Bits	Descriptions	
[31:4]	**DoneH**	**Done Head** Pointer to the current Done List Head ED.

Host Controller Frame Interval Register (HcFmIntv)

Register	Address	R/W	Description	Reset Value
HcFmIntv	0xB000_7034	R/W	Host Controller Frame Interval Register	0x0000_2EDF

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

Bits	Descriptions	
[31]	**FmIntvT**	**Frame Interval Toggle** This bit is toggled by HCD when it loads a new value into **FrameInterval**.
[30: 16]	**FSDPktCnt**	**FS Largest Data Packet** This field specifies a value that is loaded into the Largest Data Packet Counter at the beginning of each frame.
[13:0]	**FmInterval**	**Frame Interval** This field specifies the length of a frame as (bit times - 1). For 12,000 bit times in a frame, a value of 11,999 is stored here.

Host Controller Frame Remaining Register (HcFmRem)

Register	Address	R/W	Description	Reset Value
HcFmRem	0xB000_7038	R	Host Controller Frame Remaining Register	0x0000_0000

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

Bits	Descriptions	
[31]	**FmRemT**	**Frame Remaining Toggle** Loaded with **FrameIntervalToggle** when **FrameRemaining** is loaded.
[13:0]	**FmRemain**	**Frame Remaining** When the Host Controller is in the USBOPERATIONAL state, this 14-bit field decrements each 12 MHz clock period. When the count reaches 0, (end of frame) the counter reloads with **FrameInterval**. In addition, the counter loads when the Host Controller transitions into USBOPERATIONAL.

Host Controller Frame Number Register (HcFNum)

Register	Address	R/W	Description	Reset Value
HcFNum	0xB000_703C	R	Host Controller Frame Number Register	0x0000_0000

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

Bits	Descriptions	
[15:0]	**FmNum**	**Frame Number** This 16-bit incrementing counter field is incremented coincident with the loading of **FrameRemaining**. The count rolls over from 'FFFFh' to '0h.'

Host Controller Periodic Start Register (HcPerSt)

Register	Address	R/W	Description	Reset Value
HcPerSt	0xB000_7040	R/W	Host Controller Periodic Start Register	0x0000_0000

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

Bits	Descriptions	
[13:0]	**PeriStart**	**Periodic Start** This field contains a value used by the List Processor to determine where in a frame the Periodic List processing must begin.

Host Controller Root Hub Descriptor A Register (HcRhDeA)

Register	Address	R/W	Description	Reset Value
HcRhDeA	0xB000_7048	R/W	Host Controller Root Hub Descriptor A Register	0x0100_0002

31	30	29	28	27	26	25	24
PwrGDT							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved			NOCP	OCPM	DevType	NPS	PSM
7	6	5	4	3	2	1	0
DPortNum							

Bits	Descriptions	
[31:24]	**PwrGDT**	**Power On to Power Good Time** This field value is represented as the number of 2 ms intervals, which ensuring that the power switching is effective within 2 ms. Only bits [25:24] are implemented as R/W. The remaining bits are read only as '0'. It is not expected that these bits be written to anything other than 1h, but limited adjustment is provided. This field should be written to support system implementation. This field should always be written to a non-zero value.
[12]	**NOCP**	**No Over Current Protection** This bit should be written to support the external system port over-current implementation. 0 = Over-current status is reported 1 = Over-current status is not reported
[11]	**OCPM**	**Over Current Protection Mode** This bit should be written 0 and is only valid when **NOCP** bit is cleared. 0 = Global Over-Current 1 = Individual Over-Current
[10]	**DevType**	**Device Type**
[9]	**NPS**	**No Power Switching** This bit should be written to support the external system port power switching implementation. 0 = Ports are power switched. 1 = Ports are always powered on.
[8]	**PSM**	**Power Switching Mode** This bit is only valid when **NoPowerSwitching** is cleared. This bit should be written '0'. 0 = Global Switching 1 = Individual Switching
[7:0]	**DPortNum**	**Number Downstream Ports**

Host Controller Root Hub Descriptor B Register (HcRhDeB)

Register	Address	R/W	Description	Reset Value
HcRhDeB	0xB000_704C	R/W	Host Controller Root Hub Descriptor B Register	0x0000_0000

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

Bits	Descriptions	
[31:16]	**PPCM**	**Port Power Control Mask** Global-power switching. This field is only valid if **NoPowerSwitching** is cleared and **PowerSwitchingMode** is set (individual port switching). When set, the port only responds to individual port power switching commands (**Set**/**ClearPortPower**). When cleared, the port only responds to global power switching commands (**Set**/**ClearGlobalPower**). 0 = Device not removable 1 = Global-power mask Port Bit relationship - Unimplemented ports are reserved, read/write '0'. 0 : Reserved 1 : Port 1 2 : Port 2 ... 15 : Port 15
[15:0]	**DevRemove**	**Device Removable** 0 = Device not removable 1 = Device removable Port Bit relationship 0 : Reserved 1 : Port 1 2 : Port 2 ... 15 : Port 15 Unimplemented ports are reserved, read/write '0'.

Host Controller Root Hub Status Register (HcRhSts)

Register	Address	R/W	Description	Reset Value
HcRhSts	0xB000_7050	R/W	Host Controller Root Hub Status Register	0x0000_0000

31	30	29	28	27	26	25	24
RWEClr	Reserved						
23	22	21	20	19	18	17	16
Reserved						OCIC	LPSC
15	14	13	12	11	10	9	8
DRWEn	Reserved						
7	6	5	4	3	2	1	0
Reserved						OC	LPS

Bits	Descriptions	
[31]	**RWEClr**	**Clear Remote Wakeup Enable** Writing a '1' to this bit clears **DeviceRemoteWakeupEnable**. Writing a '1' has no effect.
[17]	**OCIC**	**Over Current Indicator Change** This bit is set when **OverCurrentIndicator** changes. Writing a '1' clears this bit. Writing a '0' has no effect.
[16]	**LPSC**	(Read) **LocalPowerStatusChange** Not supported. Always read '0'. (Write) **SetGlobalPower** Write a '1' issues a **SetGlobalPower** command to the ports. Writing a '0' has no effect.
[15]	**DRWEn**	(Read) **DeviceRemoteWakeupEnable** This bit enables ports' **ConnectStatusChange** as a remote wakeup event. 0 = disabled 1 = enabled (Write) **SetRemoteWakeupEnable** Writing a '1' sets **DeviceRemoteWakeupEnable**. Writing a '0' has no effect.
[1]	**OC**	**Over Current Indicator** This bit reflects the state of the OVRCUR pin. This field is only valid if **NoOverCurrentProtection** and **OverCurrentProtectionMode** are cleared. 0 = No over-current condition 1 = Over-current condition
[0]	**LPS**	(Read) **LocalPowerStatus** Not Supported. Always read '0'. (Write) **ClearGlobalPower** Writing a '1' issues a **ClearGlobalPower** command to the ports. Writing a '0' has no effect.

Host Controller Root Hub Port Status (HcRhPrt [1: 2])

Register	Address	R/W	Description	Reset Value
HcRhPrt1	0xB000_7054	R/W	Host Controller Root Hub Port Status [1]	0x0000_0000
HcRhPrt2	0xB000_7058	R/W	Host Controller Root Hub Port Status [2]	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved			PRSC	POCIC	PSSC	PESC	CSC
15	14	13	12	11	10	9	8
Reserved						LSDev	PPS
7	6	5	4	3	2	1	0
Reserved			PR	POC	PS	PE	CC

Bits	Descriptions	
[20]	**PRSC**	**Port Reset Status Change** This bit indicates that the port reset signal has completed. 0 = Port reset is not complete. 1 = Port reset is complete.
[19]	**POCIC**	**Port Over Current Indicator Change** This bit is set when **OverCurrentIndicator** changes. Writing a '1' clears this bit. Writing a '0' has no effect.
[18]	**PSSC**	**Port Suspend Status Change** This bit indicates the completion of the selective resume sequence for the port. 0 = Port is not resumed. 1 = Port resume is complete.
[17]	**PESC**	**Port Enable Status Change** This bit indicates that the port has been disabled due to a hardware event (cleared **PortEnableStatus**). 0 = Port has not been disabled. 1 = PortEnableStatus has been cleared.
[16]	**CSC**	**Connect Status Change** This bit indicates connect or disconnect event has been detected. Writing a '1' clears this bit. Writing a '0' has no effect. 0 = No connect/disconnect event. 1 = Hardware detection of connect/disconnect event. Note: If DeviceRemoveable is set, this bit resets to '1'.

[9]	**LSDev**	(Read) **LowSpeedDeviceAttached** This bit defines the speed (and bud idle) of the attached device. It is only valid when **CurrentConnectStatus** is set. 0 = Full Speed device 1 = Low Speed device (Write) **ClearPortPower** Writing a '1' clears **PortPowerStatus**. Writing a '0' has no effect
[8]	**PPS**	(Read) **PortPowerStatus** This bit reflects the power state of the port regardless of the power switching mode. 0 = Port power is off. 1 = Port power is on. Note: If **NoPowerSwitching** is set, this bit is always read as '1'. (Write) **SetPortPower** Writing a '1' sets **PortPowerStatus**. Writing a '0' has no effect.
[4]	**PR**	(Read) **PortResetStatus** 0 = Port reset signal is not active. 1 = Port reset signal is active. (Write) **SetPortReset** Writing a '1' sets **PortResetStatus**. Writing a '0' has no effect.
[3]	**POC**	(Read) **PortOverCurrentIndicator** This bit reflects the state of the OVRCUR pin dedicated to this port. This field is only valid if **NoOverCurrentProtection** is cleared and **OverCurrentProtectionMode** is set. 0 = No over-current condition 1 = Over-current condition (Write) **ClearPortSuspend** Writing a '1' initiates the selective resume sequence for the port. Writing a '0' has no effect.
[2]	**PS**	(Read) **PortSuspendStatus** 0 = Port is not suspended 1 = Port is selectively suspended (Write) **SetPortSuspend** Writing a '1' sets **PortSuspendStatus**. Writing a '0' has no effect.
[1]	**PE**	(Read) **PortEnableStatus** 0 = Port disabled. 1 = Port enabled. (Write) **SetPortEnable** Writing a '1' sets **PortEnableStatus**. Writing a '0' has no effect.
[0]	**CC**	(Read) **CurrentConnectStatus** 0 = No device connected. 1 = Device connected. NOTE: If **DeviceRemoveable** is set (not removable) this bit is always '1'. (Write) **ClearPortEnable** Writing '1' a clears **PortEnableStatus**. Writing a '0' has no effect.

USB Operational Mode Enable Register (OpModEn)

Register	Address	R/W	Description	Reset Value
OpModEn	0xB000_7204	R/W	USB Operational Mode Enable Register	0X0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							SIEPDis
7	6	5	4	3	2	1	0
Reserved				OCALow	Reserved	ABORT	DBR16

Bits	Descriptions	
[8]	**SIEPDis**	**SIE Pipeline Disable** When set, waits for all USB bus activity to complete prior to returning completion status to the List Processor. This is a failsafe mechanism to avoid potential problems with the clk_dr transition between 1.5 MHz and 12 MHz.
[3]	**OCALow**	**Over Current Active Low** This bit controls the polarity of over current flag from external power IC. 0: Over current flag is high active 1: Over current flag is low active
[1]	**ABORT**	**AHB Bus ERROR Response** This bit indicates there is an ERROR response received in AHB bus. 0: No ERROR response received 1: ERROR response received
[0]	**DBR16**	**Data Buffer Region 16** When set, the size of the data buffer region is 16 bytes. Otherwise, the size is 32 bytes.

7.8 USB 2.0 Device Controller

The NUC946ADN USB Device Controller is compliant to the USB Specification version 2.0. It also supports the software control for device remote-wakeup and 6 configurable endpoints in addition to Control Endpoint. Each of these endpoints can be Isochronous, Bulk or Interrupt and they can be either of IN or OUT direction with maximum packet size up to 1024 bytes. Three different modes of operation (Auto validation mode, manual validation mode and Fly mode) are supported for IN-endpoint.

7.8.1 USB Device Register Group Summary

Register Groups	Description
Main Control Registers	These set of registers control the global enable of interrupts and maintain the status of the interrupts
USB Control Registers	These set of registers control the USB related events to/from the USB host and hold the status of the USB events.
Control Endpoint Registers	These set of registers direct the control endpoint in handling the USB requests from the host and hold the status information of the transactions.
Non control Endpoint Registers	These set of registers configure, control and exhibit the status of the non-control endpoints' operation
DMA Registers	These registers are responsible for the DMA related operations

7.8.2 USB Device Control Registers Map

Register	Address	R/W	Description	Reset Value
USBD_BA = 0xB000_6000				
IRQ_STAT	0xB000_6000	R	Interrupt Register	0x0000_0000
IRQ_ENB_L	0xB000_6008	R/W	Interrupt Enable Low Register	0x0000_0001
USB_IRQ_STAT	0xB000_6010	R/W	USB Interrupt Status register	0x0000_0000
USB_IRQ_ENB	0xB000_6014	R/W	USB Interrupt Enable register	0x0000_0040
USB_OPER	0xB000_6018	R/W	USB operational register	0x0000_0002

USB_FRAME_CNT	0xB000_601C	R	USB frame count register	0x0000_0000
USB_ADDR	0xB000_6020	R/W	USB address register	0x0000_0000
CEP_DATA_BUF	0xB000_6028	R/W	Control-ep Data Buffer	0x0000_0000
CEP_CTRL_STAT	0xB000_602C	R/W	Control-ep Control and Status	0x0000_0000
CEP_IRQ_ENB	0xB000_6030	R/W	Control-ep Interrupt Enable	0x0000_0000
CEP_IRQ_STAT	0xB000_6034	R/W	Control-ep Interrupt Status	0x0000_1000
IN_TRNSFR_CNT	0xB000_6038	R/W	In-transfer data count	0x0000_0000
OUT_TRNSFR_CNT	0xB000_603C	R	Out-transfer data count	0x0000_0000
CEP_CNT	0xB000_6040	R	Control-ep data count	0x0000_0000
SETUP1_0	0xB000_6044	R	Setupbyte1 & byte0	0x0000_0000
SETUP3_2	0xB000_6048	R	Setupbyte3 & byte2	0x0000_0000
SETUP5_4	0xB000_604C	R	Setupbyte5 & byte4	0x0000_0000
SETUP7_6	0xB000_6050	R	Setupbyte7 & byte6	0x0000_0000
CEP_START_ADDR	0xB000_6054	R/W	Control EP's RAM start address	0x0000_0000
CEP_END_ADDR	0xB000_6058	R/W	Control EP's RAM end address	0x0000_0000
DMA_CTRL_STS	0xB000_605C	R/W	DMA control and status register	0x0000_0000
DMA_CNT	0xB000_6060	R/W	DMA count register	0x0000_0000
EPA_DATA_BUF	0xB000_6064	R/W	Endpoint A data register	0x0000_0000
EPA_IRQ_STAT	0xB000_6068	R/W	Endpoint A Interrupt status register	0x0000_0002
EPA_IRQ_ENB	0xB000_606C	R/W	Endpoint A Interrupt enable register	0x0000_0000
EPA_DATA_CNT	0xB000_6070	R	Data count available in endpoint A buffer	0x0000_0000
EPA_RSP_SC	0xB000_6074	R/W	Endpoint A response register set/clear	0x0000_0000
EPA_MPS	0xB000_6078	R/W	Endpoint A maximum packet size register	0x0000_0000
EPA_CNT	0xB000_607C	R/W	Endpoint A transfer count register	0x0000_0000
EPA_CFG	0xB000_6080	R/W	Endpoint A configuration register	0x0000_0012
EPA_START_ADDR	0xB000_6084	R/W	Endpoint A RAM start address	0x0000_0000
EPA_END_ADDR	0xB000_6088	R/W	Endpoint A RAM end address	0x0000_0000
EPB_DATA_BUF	0xB000_608C	R/W	Endpoint B data register	0x0000_0000
EPB_IRQ_STAT	0xB000_6090	R/W	Endpoint B Interrupt status register	0x0000_0002
EPB_IRQ_ENB	0xB000_6094	R/W	Endpoint B Interrupt enable register	0x0000_0000
EPB_DATA_CNT	0xB000_6098	R	Data count available in endpoint B buffer	0x0000_0000

EPB_RSP_SC	0xB000_609C	R/W	Endpoint B response register set/clear	0x0000_0000
EPB_MPS	0xB000_60A0	R/W	Endpoint B maximum packet size register	0x0000_0000
EPB_TRF_CNT	0xB000_60A4	R/W	Endpoint B transfer count register	0x0000_0000
EPB_CFG	0xB000_60A8	R/W	Endpoint B configuration register	0x0000_0022
EPB_START_ADDR	0xB000_60AC	R/W	Endpoint B RAM start address	0x0000_0000
EPB_END_ADDR	0xB000_60B0	R/W	Endpoint B RAM end address	0x0000_0000
EPC_DATA_BUF	0xB000_60B4	R/W	Endpoint C data register	0x0000_0000
EPC_IRQ_STAT	0xB000_60B8	R/W	Endpoint C Interrupt status register	0x0000_0002
EPC_IRQ_ENB	0xB000_60BC	R/W	Endpoint C Interrupt enable register	0x0000_0000
EPC_DATA_CNT	0xB000_60C0	R	Data count available in endpoint C buffer	0x0000_0000
EPC_RSP_SC	0xB000_60C4	R/W	Endpoint C response register set/clear	0x0000_0000
EPC_MPS	0xB000_60C8	R/W	Endpoint C maximum packet size register	0x0000_0000
EPC_TRF_CNT	0xB000_60CC	R/W	Endpoint C transfer count register	0x0000_0000
EPC_CFG	0xB000_60D0	R/W	Endpoint C configuration register	0x0000_0032
EPC_START_ADDR	0xB000_60D4	R/W	Endpoint C RAM start address	0x0000_0000
EPC_END_ADDR	0xB000_60D8	R/W	Endpoint C RAM end address	0x0000_0000
EPD_DATA_BUF	0xB000_60DC	R/W	Endpoint D data register	0x0000_0000
EPD_IRQ_STAT	0xB000_60E0	R/W	Endpoint D Interrupt status register	0x0000_0002
EPD_IRQ_ENB	0xB000_60E4	R/W	Endpoint D Interrupt enable register	0x0000_0000
EPD_DATA_CNT	0xB000_60E8	R	Data count available in endpoint D buffer	0x0000_0000
EPD_RSP_SC	0xB000_60EC	R/W	Endpoint D response register set/clear	0x0000_0000
EPD_MPS	0xB000_60F0	R/W	Endpoint D maximum packet size register	0x0000_0000
EPD_TRF_CNT	0xB000_60F4	R/W	Endpoint D transfer count register	0x0000_0000
EPD_CFG	0xB000_60F8	R/W	Endpoint D configuration register	0x0000_0042
EPD_START_ADDR	0xB000_60FC	R/W	Endpoint D RAM start address	0x0000_0000
EPD_END_ADDR	0xB000_6100	R/W	Endpoint D RAM end address	0x0000_0000
EPE_DATA_BUF	0xB000_6104	R/W	Endpoint E data register	0x0000_0000
EPE_IRQ_STAT	0xB000_6108	R/W	Endpoint E Interrupt status register	0x0000_0002
EPE_IRQ_ENB	0xB000_610C	R/W	Endpoint E Interrupt enable register	0x0000_0000
EPE_DATA_CNT	0xB000_6110	R	Data count available in endpoint E buffer	0x0000_0000
EPE_RSP_SC	0xB000_6114	R/W	Endpoint E response register set/clear	0x0000_0000

EPE_MPS	0xB000_6118	R/W	Endpoint E maximum packet size register	0x0000_0000
EPE_TRF_CNT	0xB000_611C	R/W	Endpoint E transfer count register	0x0000_0000
EPE_CFG	0xB000_6120	R/W	Endpoint E configuration register	0x0000_0052
EPE_START_ADDR	0xB000_6124	R/W	Endpoint E RAM start address	0x0000_0000
EPE_END_ADDR	0xB000_6128	R/W	Endpoint E RAM end address	0x0000_0000
EPF_DATA_BUF	0xB000_612C	R/W	Endpoint F data register	0x0000_0000
EPF_IRQ_STAT	0xB000_6130	R/W	Endpoint F Interrupt status register	0x0000_0002
EPF_IRQ_ENB	0xB000_6134	R/W	Endpoint F Interrupt enable register	0x0000_0000
EPF_DATA_CNT	0xB000_6138	R	Data count available in endpoint F buffer	0x0000_0000
EPF_RSP_SC	0xB000_613C	R/W	Endpoint F response register set/clear	0x0000_0000
EPF_MPS	0xB000_6140	R/W	Endpoint F maximum packet size register	0x0000_0000
EPF_TRF_CNT	0xB000_6144	R/W	Endpoint F transfer count register	0x0000_0000
EPF_CFG	0xB000_6148	R/W	Endpoint F configuration register	0x0000_0062
EPF_START_ADDR	0xB000_614C	R/W	Endpoint F RAM start address	0x0000_0000
EPF_END_ADDR	0xB000_6150	R/W	Endpoint F RAM end address	0x0000_0000
USB_DMA_ADDR	0xB000_6700	R/W	AHB_DMA address register	0x0000_0000
USB_PHY_CTL	0xB000_6704	R/W	USB PHY control register	0x0000_0060

7.8.3 USB Device Control Registers

Interrupt Register (IRQ)

Register	Address	R/W	Description	Default Value
IRQ	0xB000_6000	R	Interrupt Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
EPF_INT	EPE_INT	EPD_INT	EPC_INT	EPB_INT	EPA_INT	CEP_INT	USB_INT

Bits	Descriptions	
[7]	**EPF_INT**	**This bit conveys the interrupt for Endpoints F.** When set, the corresponding Endpoint F's interrupt status register should be read to determine the cause of the interrupt.
[6]	**EPE_INT**	**This bit conveys the interrupt for Endpoints E.** When set, the corresponding Endpoint E's interrupt status register should be read to determine the cause of the interrupt.
[5]	**EPD_INT**	**This bit conveys the interrupt for Endpoints D.** When set, the corresponding Endpoint D's interrupt status register should be read to determine the cause of the interrupt.
[4]	**EPC_INT**	**This bit conveys the interrupt for Endpoints C.** When set, the corresponding Endpoint C's interrupt status register should be read to determine the cause of the interrupt.
[3]	**EPB_INT**	**This bit conveys the interrupt for Endpoints B.** When set, the corresponding Endpoint B's interrupt status register should be read to determine the cause of the interrupt.
[2]	**EPA_INT**	**This bit conveys the interrupt for Endpoints A.** When set, the corresponding Endpoint A's interrupt status register should be read to determine the cause of the interrupt.
[1]	**CEP_INT**	**Control Endpoint Interrupt.** This bit conveys the interrupt status for control endpoint. When set, Control-ep's interrupt status register should be read to determine the cause of the interrupt.

[0]	**USB_INT**	**USB Interrupt.** The interrupt status for USB specific events endpoint. When set, USB interrupt status register should be read to determine the cause of the interrupt.

Interrupt Enable Low Register (IRQ_ENB_L)

Register	Address	R/W	Description	Default Value
IRQ_ENB_L	0xB000_6008	R/W	Interrupt Enable Low Register	0x0000_0001

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
EPF_IE	EPE_IE	EPD_IE	EPC_IE	EPB_IE	EPA_IE	CEP_IE	USB_IE

Bits	Descriptions	
[7]	**EPF_IE**	**Interrupt Enable for Endpoint F.** When set, this bit enables a local interrupt to be generated when an interrupt is pending for the endpoint F
[6]	**EPE_IE**	**Interrupt Enable for Endpoint E.** When set, this bit enables a local interrupt to be generated when an interrupt is pending for the endpoint E
[5]	**EPD_IE**	**Interrupt Enable for Endpoint D.** When set, this bit enables a local interrupt to be generated when an interrupt is pending for the endpoint D
[4]	**EPC_IE**	**Interrupt Enable for Endpoint C.** When set, this bit enables a local interrupt to be generated when an interrupt is pending for the endpoint C
[3]	**EPB_IE**	**Interrupt Enable for Endpoint B.** When set, this bit enables a local interrupt to be generated when an interrupt is pending for the endpoint B
[2]	**EPA_IE**	**Interrupt Enable for Endpoint A.** When set, this bit enables a local interrupt to be generated when an interrupt is pending for the endpoint A.
[1]	**CEP_IE**	**Control Endpoint Interrupt Enable.** When set, this bit enables a local interrupt to be generated when an interrupt is pending for the control endpoint.
[0]	**USB_IE**	**USB Interrupt Enable.** When set, this bit enables a local interrupt to be generated when a USB event occurs on the bus.

USB Interrupt Status Register (USB_IRQ_STAT)

Register	Address	R/W	Description	Default Value
USB_IRQ_STAT	0xB000_6010	R/W	USB Interrupt Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved	TCLKOK_IS	DMACOM_IS	HISPD_IS	SUS_IS	RUM_IS	RST_IS	SOF_IS

Bits	Descriptions	
[6]	**TCLKOK_IS**	**Usable Clock Interrupt.** This bit is set when usable clock is available from the transceiver. Writing '1" clears this bit.
[5]	**DMACOM_IS**	**DMA Completion Interrupt.** This bit is set when the DMA transfer is over. Writing '1" clears this bit.
[4]	**HISPD_IS**	**High Speed Settle.** This bit is set when the valid high-speed reset protocol is over and the device has settled is high-speed. Writing '1" clears this bit.
[3]	**SUS_IS**	**Suspend Request.** This bit is set as default and it has to be cleared by writing '1' before the USB reset. This bit is also set when a USB Suspend request is detected from the host. Writing '1' clears this bit.
[2]	**RUM_IS**	**Resume.** When set, this bit indicates that a device resume has occurred. Writing a '1' clears this bit.
[1]	**RST_IS**	**Reset Status.** When set, this bit indicates that either the USB root port reset is end. Writing a '1' clears this bit.
[0]	**SOF_IS**	**SOF.** This bit indicates when a start-of-frame packet has been received. Writing a '1' clears this bit.

USB Interrupt Enable Register (USB_IRQ_ENB)

Register	Address	R/W	Description	Default Value
USB_IRQ_ENB	0xB000_6014	R/W	USB Interrupt Enable Register	0x0000_0040

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved	TCLKOK_IE	DMACOM_IE	HISPD_IE	SUS_IE	RUM_IE	RST_IE	SOF_IE

Bits	Descriptions	
[6]	**TCLKOK_IE**	**Usable Clock Interrupt.** This bit enables the usable clock interrupt.
[5]	**DMACOM_IE**	**DMA Completion Interrupt.** This bit enables the DMA completion interrupt
[4]	**HISPD_IE**	**High Speed Settle.** This bit enables the high-speed settle interrupt.
[3]	**SUS_IE**	**Suspend Request**. This bit enables the Suspend interrupt.
[2]	**RUM_IE**	**Resume.** This bit enables the Resume interrupt.
[1]	**RST_IE**	**Reset Status.** This bit enables the USB-Reset interrupt.
[0]	**SOF_IE**	**SOF Interrupt.** This bit enables the SOF interrupt.

USB Operational Register (USB_OPER)

Register	Address	R/W	Description	Default Value
USB_OPER	0xB000_6018	R/W	USB Operational Register	0x0000_0002

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

Bits	Descriptions	
[2]	**CUR_SPD**	**USB Current Speed.** When set, this bit indicates that the DEVICE CONTROLLER has settled in High Speed and a zero indicates that the device has settled in Full Speed
[1]	**SET_HISPD**	**USB High Speed.** When set to one, this bit indicates the DEVICE CONTROLLER to initiate a chirp-sequence during reset protocol, if it set to zero, it indicates the DEVICE CONTROLLER to suppress the chirp-sequence during reset protocol, thereby allowing the DEVICE CONTROLLER to settle in full-speed, even though it is connected to a USB2.0 Host.
[0]	**GEN_RUM**	**Generate Resume.** Writing a 1 to this bit causes a Resume sequence to be initiated to the host if device remote wakeup is enabled. This bit is self-clearing.

USB Frame Count Register (USB_FRAME_CNT)

Register	Address	R/W	Description	Default Value
USB_FRAME_CNT	0xB000_601C	R	USB Frame Count Register	0x0000_0000

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

Bits	Descriptions	
[13:3]	**FRAME_CNT**	**FRAME COUNTER.** This field contains the frame count from the most recent start-of-frame packet.
[2:0]	**MFRAME_CNT**	**MICRO FRAME COUNTER.** This field contains the micro-frame number for the frame number in the frame counter field.

USB Address Register (USB_ADDR)

Register	Address	R/W	Description	Default Value
USB_ADDR	0xB000_6020	R/W	USB Address Register	0x0000_0000

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

Bits	Descriptions	
[6:0]	**ADDR**	This field contains the current USB address of the device. This field is cleared when a root port reset is detected.

Control-ep Data Buffer (CEP_DATA_BUF)

Register	Address	R/W	Description	Default Value
CEP_DATA_BUF	0xB000_6028	RW	Control-ep Data Buffer	0x0000_0000

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

Bits	Descriptions	
[15:0]	**DATA_BUF**	**Control-ep Data Buffer.** Bits [15:8] of this register provide the high order byte and bits [7:0] of this register provide the lower order byte for the buffer transaction (read or write).

Control-ep Control and Status (CEP_CTRL_STAT)

Register	Address	R/W	Description	Default Value
CEP_CTRL_STAT	0xB000_602C	RW	Control-ep Control and Status	0x0000_0000

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

Bits	Descriptions	
[3]	**FLUSH**	**CEP-FLUSH Bit.** Writing 1 to this bit cause the packet buffer and its corresponding CEP_AVL_CNT register to be cleared. This bit is self-cleaning.
[2]	**ZEROLEN**	**ZEROLEN Bit.** This bit is valid for auto validation mode only. When this bit is set, DEVICE CONTROLLER can send a zero length packet to the host during Data stage to an IN token. This bit gets cleared once the zero length data packet is sent. So, the local CPU need not write again to clear this bit.
[1]	**STLALL**	**STALL.** This bit is a read/write bit. When this stall bit is set, the control endpoint sends a stall handshake in response to any in or out token thereafter. This is typically used for response to invalid/unsupported requests. When this bit is being set the NAK clear bit has to be cleared at the same time since the NAK clear bit has highest priority than STALL. It is automatically cleared on receipt of a next setup-token. So, the local CPU need not write again to clear this bit. NOTE: ONLY when cpu write data[1:0] is 2'b10 or 2'b00, this bit can be updated.

[0]	**NAK_CLEAR**	**NAK_CLEAR.** This is a read/write bit. This bit plays a crucial role in any control transfer. It bit is set to one by the DEVICE CONTROLLER, whenever a setup token is received. The local CPU can take its own time to finish off any house-keeping work based on the request and then clear this bit. Unless the bit is being cleared by the local CPU by writing zero, the DEVICE CONTROLLER will be responding with NAKs for the subsequent status phase. This mechanism holds the host from moving to the next request, until the local CPU is also ready to process the next request. NOTE: ONLY when CPU write data[1:0] is 2'b10 or 2'b00, this bit can be updated.

Control Endpoint Interrupt Enable (CEP_IRQ_ENABLE)

Register	Address	R/W	Description	Default Value
CEP_IRQ_ENABLE	0xB000_6030	R/W	Control Endpoint Interrupt Enable	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved			EMPTY_IE	FULL_IE	STACOM_IE	ERR_IE	STALL_IE
7	6	5	4	3	2	1	0
NAK_IE	DATA_RxED_IE	DATA_TxED_IE	PING_IE	IN_TK_IE	OUT_TK_IE	SETUP_PK_IE	SETUP_TK_IE

Bits	Descriptions	
[12]	**EMPTY_IE**	**Buffer Empty Interrupt.** This bit enables the buffer empty interrupt.
[11]	**FULL_IE**	**Buffer Full Interrupt.** This bit enables the buffer full interrupt.
[10]	**STACOM_IE**	**Status Completion Interrupt.** This bit enables the Status Completion interrupt.
[9]	**ERR_IE**	**USB Error Interrupt.** This bit enables the USB Error interrupt.
[8]	**STALL_IE**	**STALL Sent Interrupt.** This bit enables the STALL sent interrupt
[7]	**NAK_IE**	**NAK Sent Interrupt.** This bit enables the NAK sent interrupt.
[6]	**DATA_RxED_IE**	**Data Packet Received Interrupt.** This bit enables the data received interrupt.
[5]	**DATA_TxED_IE**	**Data Packet Transmitted Interrupt.** This bit enables the data packet transmitted interrupt.
[4]	**PING_IE**	**Ping Token Interrupt**. This bit enables the ping token interrupt.

[3]	**IN_TK_IE**	**In Token Interrupt.** This bit enables the in token interrupt
[2]	**OUT_TK_IE**	**Out Token Interrupt.** This bit enables the out token interrupt.
[1]	**SETUP_PK_IE**	**Setup Packet Interrupt.** This bit enables the setup packet interrupt.
[0]	**SETUP_TK_IE**	**Setup Token Interrupt Enable.** This bit enables the setup token interrupt.

Control-Endpoint Interrupt Status (CEP_IRQ_STAT)

Register	Address	R/W	Description	Default Value
CEP_IRQ_STAT	0xB000_6034	R/W	Control-ep Interrupt Status	0x0000_1000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved			EMPTY_IS	FULL_IS	STACOM_IS	ERR_IS	STALL_IS
7	6	5	4	3	2	1	0
NAK_IS	DATA_RxED_IS	DATA_TxED_IS	PING_IS	IN_TK_IS	OUT_TK_IS	SETUP_PK_IS	SETUP_TK_IS

Bits	Descriptions	
[12]	**EMPTY_IS**	**Buffer Empty Interrupt.** (Read Only) This bit is set when the control-ednpt buffer is empty.
[11]	**FULL_IS**	**Buffer Full Interrupt.** (Write "1" Clear) This bit is set when the control-endpt buffer is full.
[10]	**STACOM_IS**	**Status Completion Interrupt.** (Write "1" Clear) This bit is set when the status stage of a USB transaction has completed successfully.
[9]	**ERR_IS**	**USB Error Interrupt.** (Write "1" Clear) This bit is set when an error had occurred during the transaction.
[8]	**STALL_IS**	**STALL Sent Interrupt.** (Write "1" Clear) This bit is set when a stall-token is sent in response to an in/out token
[7]	**NAK_IS**	**NAK Sent Interrupt.** (Write "1" Clear) This bit is set when a nak-token is sent in response to an in/out token
[6]	**DATA_RxED_IS**	**Data Packet Received Interrupt**. (Write "1" Clear) This bit is set when a data packet is successfully received from the host for an out-token and an ack is sent to the host.
[5]	**DATA_TxED_IS**	**Data Packet Transmitted Interrupt.** (Write "1" Clear) This bit is set when a data packet is successfully transmitted to the host in response to an in-token and an ack-token is received for the same.

[4]	**PING_IS**	**Ping Token Interrupt.** (Write "1" Clear) This bit is set when the control-endpt receives a ping token from the host.
[3]	**IN_TK_IS**	**In Token Interrupt.** (Write "1" Clear) This bit is set when the control-endpt receives an in token from the host.
[2]	**OUT_TK_IS**	**Out Token Interrupt.** (Write "1" Clear) This bit is set when the control-endpoint receives an out token from the host.
[1]	**SETUP_PK_IS**	**Setup Packet Interrupt.** (Write "1" Clear) This bit is set when a setup packet has been received from the host. This bit must be cleared (by writing a 1) before the next setup packet can be received. If the bit is not cleared, then the successive setup packets will be overwritten in the setup packet buffer.
[0]	**SETUP_TK_IS**	**Setup Token Interrupt.** (Write "1" Clear) This bit indicates when a setup token is received. Writing a 1 clears this status bit

In-transfer data count (IN_TRF_CNT)

Register	Address	R/W	Description	Default Value
IN_TRF_CNT	0xB000_6038	R/W	In-transfer data count	0x0000_0000

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

Bits	Descriptions	
[7:0]	**IN_TRF_CNT**	**In-transfer data count.** There is no mode selection for the control endpoint (but it operates like manual mode).The local-CPU has to fill the control-endpoint buffer with the data to be sent for an in-token and to write the count of bytes in this register. When zero is written into this field, a zero length packet is sent to the host. When the count written in the register is more than the MPS, the data sent will be of only MPS.

Out-transfer data count (OUT_TRF_CNT)

Register	Address	R/W	Description	Default Value
OUT_TRF_CNT	0xB000_603C	R	Out-transfer data count	0x0000_0000

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

Bits	Descriptions	
[15:0]	**OUT_TRF_CNT**	**Out-Transfer Data Count.** The DEVICE CONTROLLER maintains the count of the data received in case of an out transfer, during the control transfer.

Control- Endpoint data count (CEP_CNT)

Register	Address	R/W	Description	Default Value
CEP_CNT	0xB000_6040	R	Control-ep data count	0x0000_0000

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

Bits	Descriptions	
[15:0]	**CEP_CNT**	**Control-ep Data Count.** The DEVICE CONTROLLER maintains the count of the data of control-ep.

Setup1 & Setup0 bytes (SETUP1_0)

Register	Address	R/W	Description	Default Value
SETUP1_0	0xB000_6044	R	Setup1 & Setup0 bytes	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
SETUP1							
7	6	5	4	3	2	1	0
SETUP0							

Bits	Descriptions	
[15:8]	**SETUP1**	**Setup Byte 1[15:8].** This register provides byte 1 of the last setup packet received. For a Standard Device Request, the following bRequest Code information is returned.

Code	Descriptions
0x00	Get Status
0x01	Clear Feature
0x02	Reserved
0x03	Set Feature
0x04	Reserved
0x05	Set Address
0x06	Get Descriptor
0x07	Set Descriptor
0x08	Get Configuration
0x09	Set Configuration
0x0A	Get Interface
0x0B	Set Interface
0x0C	Synch Frame

<table>
<tr>
<td>[7:0]</td>
<td>SETUP0</td>
<td>Setup Byte 0[7:0].
This register provides byte 0 of the last setup packet received. For a Standard Device Request, the following bmRequestType information is returned.
<table>
<tr><th>Bits</th><th colspan="2">Descriptions</th></tr>
<tr><td>[7]</td><td>Direction</td><td>0 = host to device;
1 = device to host</td></tr>
<tr><td>[6:5]</td><td>Type</td><td>0 = Standard,
1 = Class,
2 = Vendor,
3 = Reserved</td></tr>
<tr><td>[4:0]</td><td>Recipient</td><td>0 = Device,
1 = Interface,
2 = Endpoint,
3 = Other,
4-31 Reserved</td></tr>
</table>
</td>
</tr>
</table>

Setup3 & Setup2 bytes (SETUP3_2)

Register	Address	R/W	Description	Default Value
SETUP3_2	0xB000_6048	R	Setup3 & Setup2 bytes	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
SETUP3							
7	6	5	4	3	2	1	0
SETUP2							

Bits	Descriptions	
[15:8]	**SETUP3**	**Setup Byte 3 [15:8].** This register provides byte 3 of the last setup packet received. For a Standard Device Request, the most significant byte of the wValue field is returned.
[7:0]	**SETUP2**	**Setup Byte 2 [7:0].** This register provides byte 2 of the last setup packet received. For a Standard Device Request, the least significant byte of the wValue field is returned.

Setup5 & Setup4 bytes (SETUP5_4)

Register	Address	R/W	Description	Default Value
SETUP5_4	0xB000_604C	R	Setup5 & Setup4 bytes	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
SETUP5							
7	6	5	4	3	2	1	0
SETUP4							

Bits	Descriptions	
[15:8]	**SETUP5**	**Setup Byte 5[15:8].** This register provides byte 5 of the last setup packet received. For a Standard Device Request, the most significant byte of the wIndex field is returned.
[7:0]	**SETUP4**	**Setup Byte 4[7:0].** This register provides byte 4 of the last setup packet received. For a Standard Device Request, the least significant byte of the wIndex is returned.

Setup7 & Setup6 bytes (SETUP7_6)

Register	Address	R/W	Description	Default Value
SETUP7_6	0xB000_6050	R	Setup7 & Setup6 bytes	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
SETUP7							
7	6	5	4	3	2	1	0
SETUP6							

Bits	Descriptions	
[15:8]	**SETUP7**	**Setup Byte 7[15:8].** This register provides byte 7 of the last setup packet received. For a Standard Device Request, the most significant byte of the wLength field is returned.
[7:0]	**SETUP6**	**Setup Byte 6[7:0].** This register provides byte 6 of the last setup packet received. For a Standard Device Request, the least significant byte of the wLength field is returned.

Control Endpoint RAM Start Address Register (CEP_START_ADDR)

Register	Address	R/W	Description	Default Value
CEP_START_ADDR	0xB000_6054	R/W	Control EP RAM Start Address Register	0x0000_0000

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

Bits	Descriptions	
[10:0]	**CEP_START_ADDR**	This is the start-address of the RAM space allocated for the control-endpoint

Control Endpoint RAM End Address Register (CEP_END_ADDR)

Register	Address	R/W	Description	Default Value
CEP_END_ADDR	0xB000_6058	R/W	Control EP RAM End Address Register	0x0000_0000

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

Bits	Descriptions	
[10:0]	**CEP_END_ADDR**	This is the end-address of the RAM space allocated for the control-endpoint

DMA Control Status Register (DMA_CTRL_STS)

Register	Address	R/W	Description	Default Value
DMA_CTRL_STS	0xB000_605C	R/W	DMA Control Status Register	0x0000_0000

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

Bits	Descriptions	
[7]	**RST_DMA**	**Reset DMA state machine.**
[6]	**SCAT_GA_EN**	**Scatter gather function enable**
[5]	**DMA_EN**	**DMA Enable Bit**
[4]	**DMA_RD**	**DMA Operation Bit.** If '1', the operation is a DMA read and if '0' the operation is a DMA write.
[3:0]	**DMA_ADDR**	**DMA ep_addr Bits**

When enable scatter gather DMA function, SCAT_GA_EN needs to be set high and DMA_CNT set to 8 bytes. Then DMA will enable to fetch the descriptor which describes the real memory address and length. The descriptor will be a
8-byte format, like the following:

[31]	[30]	[29:0]	
MEM_ADDR[31:0]			
EOT	RD	reserved	count[19:0]

MEM_ADDR: It specifies the memory address (AHB address).
EOT: end of transfer. When this bit sets to high, it means this is the last descriptor.
RD: "1" means read from memory into buffer. "0" means read from buffer into memory.

DMA Count Register (DMA_CNT)

Register	Address	R/W	Description	Default Value
DMA_CNT	0xB000_6060	R/W	DMA Count Register	0x0000_0000

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

Bits	Descriptions	
[19:0]	**DMA_CNT**	The transfer count of the DMA operation to be performed is written to this register.

Endpoint A~F Data Register (EPA_DATA_BUF~ EPF_DATA_BUF)

Register	Address	R/W	Description	Default Value
EPA_DATA_BUF	0xB000_6064	R/W	Endpoint A Data Register	0x0000_0000
EPB_DATA_BUF	0xB000_608C	R/W	Endpoint B Data Register	0x0000_0000
EPC_DATA_BUF	0xB000_60B4	R/W	Endpoint C Data Register	0x0000_0000
EPD_DATA_BUF	0xB000_60DC	R/W	Endpoint D Data Register	0x0000_0000
EPE_DATA_BUF	0xB000_6104	R/W	Endpoint E Data Register	0x0000_0000
EPF_DATA_BUF	0xB000_612C	R/W	Endpoint F Data Register	0x0000_0000

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

Bits	Descriptions	
[15:0]	**EP_DATA_BUF**	**Endpoint A~F Data Register.** Bits [15:8] of this register provide the high order byte and bits [7:0] of this register provide the lower order byte for the buffer transaction (read or write).

Endpoint A~F Interrupt Status Register (EPA_IRQ_STAT~ EPF_IRQ_STAT)

Register	Address	R/W	Description	Default Value
EPA_IRQ_STAT	0xB000_6068	R/W	Endpoint A Interrupt Status Register	0x0000_0002
EPB_IRQ_STAT	0xB000_6090	R/W	Endpoint B Interrupt Status Register	0x0000_0002
EPC_IRQ_STAT	0xB000_60B8	R/W	Endpoint C Interrupt Status Register	0x0000_0002
EPD_IRQ_STAT	0xB000_60E0	R/W	Endpoint D Interrupt Status Register	0x0000_0002
EPE_IRQ_STAT	0xB000_6104	R/W	Endpoint E Interrupt Status Register	0x0000_0002
EPF_IRQ_STAT	0xB000_6130	R/W	Endpoint F Interrupt Status Register	0x0000_0002

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved			O_SHORT_PKT_IS	ERR_IS	NYET_IS	STALL_IS	NAK_IS
7	6	5	4	3	2	1	0
PING_IS	IN_TK_IS	OUT_TK_IS	DATA_RxED_IS	DATA_TxED_IS	SHORT_PKT_IS	EMPTY_IS	FULL_IS

Bits	Descriptions	
[12]	O_SHORT_PKT_IS	**Bulk Out Short Packet Received** (Writing a '1' clears this bit.) Received bulk out short packet (including zero length packet)
[11]	ERR_IS	**ERR Sent.** (Writing a '1' clears this bit.) This bit is set when there occurs any error in the transaction.
[10]	NYET_IS	**NYET Sent.** (Writing a '1' clears this bit.) This bit is set when the space available in the RAM is not sufficient to accommodate the next on coming data packet.
[9]	STALL_IS	**USB STALL Sent.** (Writing a '1' clears this bit.) The last USB packet could not be accepted or provided because the endpoint was stalled, and was acknowledged with a STALL.
[8]	NAK_IS	**USB NAK Sent.** (Writing a '1' clears this bit.) The last USB IN packet could not be provided, and was acknowledged with a NAK.
[7]	PING_IS	**PING Token Interrupt.** (Writing a '1' clears this bit.) This bit is set when a Data IN token has been received from the host.

[6]	**IN_TK_IS**	**Data IN Token Interrupt.** (Writing a '1' clears this bit.) This bit is set when a Data IN token has been received from the host.
[5]	**OUT_TK_IS**	**Data OUT Token Interrupt.** (Writing a '1' clears this bit.) This bit is set when a Data OUT token has been received from the host. This bit also set by PING tokens (in high-speed only).
[4]	**DATA_RxED_IS**	**Data Packet Received Interrupt.** (Writing a '1' clears this bit.) This bit is set when a data packet is received from the host by the endpoint.
[3]	**DATA_TxED_IS**	**Data Packet Transmitted Interrupt.** (Writing a '1' clears this bit.) This bit is set when a data packet is transmitted from the endpoint to the host.
[2]	**SHORT_PKT_IS**	**Short Packet Transferred Interrupt.** (Writing a '1' clears this bit.) This bit is set when the length of the last packet was less than the Maximum Packet Size (EP_MPS).
[1]	**EMPTY_IS**	**Buffer Empty.** (READ ONLY) For an IN endpoint, a buffer is available to the local side for writing up to FIFO full of bytes. This bit is set when the endpoint buffer is empty. For an OUT endpoint, the currently selected buffer has a count of 0, or no buffer is available on the local side (nothing to read).
[0]	**FULL_IS**	**Buffer Full.** (READ ONLY) This bit is set when the endpoint packet buffer is full. For an IN endpoint, the currently selected buffer is full, or no buffer is available to the local side for writing (no space to write). For an OUT endpoint, there is a buffer available on the local side, and there are FIFO full of bytes available to be read (entire packet is available for reading).

Endpoint A~F Interrupt Enable Register (EPA_IRQ_ENB~ EPF_IRQ_ENB)

Register	Address	R/W	Description	Default Value
EPA_IRQ_ENB	0xB000_606C	R/W	Endpoint A Interrupt Enable Register	0x0000_0000
EPB_IRQ_ENB	0xB000_6094	R/W	Endpoint B Interrupt Enable Register	0x0000_0000
EPC_IRQ_ENB	0xB000_60BC	R/W	Endpoint C Interrupt Enable Register	0x0000_0000
EPD_IRQ_ENB	0xB000_60E4	R/W	Endpoint D Interrupt Enable Register	0x0000_0000
EPE_IRQ_ENB	0xB000_610C	R/W	Endpoint E Interrupt Enable Register	0x0000_0000
EPF_IRQ_ENB	0xB000_6134	R/W	Endpoint F Interrupt Enable Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved			O_SHORT_PKT_IE	ERR_IE	NYET_IE	STALL_IE	NAK_IE
7	6	5	4	3	2	1	0
PING_IE	IN_TK_IE	OUT_TK_IE	DATA_RxED_IE	DATA_TxED_IE	SHORT_PKT_IE	EMPTY_IE	FULL_IE

Bits	Descriptions	
[12]	**O_SHORT_PKT_IE**	**Bulk Out Short Packet Interrupt Enable** When set, this bit enables a local interrupt to be set whenever bulk-out short packet occurs on the bus for this endpoint.
[11]	**ERR_IE**	**ERR interrupt Enable.** When set, this bit enables a local interrupt to be set whenever ERR condition occurs on the bus for this endpoint.
[10]	**NYET_IE**	**NYET Interrupt Enable.** When set, this bit enables a local interrupt to be set whenever NYET condition occurs on the bus for this endpoint.
[9]	**STALL_IE**	**USB STALL Sent Interrupt Enable.** When set, this bit enables a local interrupt to be set when a stall token is sent to the host.
[8]	**NAK_IE**	**USB NAK Sent Interrupt Enable.** When set, this bit enables a local interrupt to be set when a nak token is sent to the host.

[7]	**PING_IE**	**PING Token Interrupt Enable.** When set, this bit enables a local interrupt to be set when a ping token has been received from the host.
[6]	**IN_TK_IE**	**Data IN Token Interrupt Enable.** When set, this bit enables a local interrupt to be set when a Data IN token has been received from the host.
[5]	**OUT_TK_IE**	**Data OUT Token Interrupt Enable.** When set, this bit enables a local interrupt to be set when a Data OUT token has been received from the host.
[4]	**DATA_RxED_IE**	**Data Packet Received Interrupt Enable**. When set, this bit enables a local interrupt to be set when a data packet has been transmitted to the host.
[3]	**DATA_TxED_IE**	**Data Packet Transmitted Interrupt Enable.** When set, this bit enables a local interrupt to be set when a data packet has been received from the host.
[2]	**SHORT_PKT_IE**	**Short Packet Transferred Interrupt Enable.** When set, this bit enables a local interrupt to be set when a short data packet has been transferred to/from the host.
[1]	**EMPTY_IE**	**Buffer Empty Interrupt.** When set, this bit enables a local interrupt to be set when a buffer empty condition is detected on the bus.
[0]	**FULL_IE**	**Buffer Full Interrupt.** When set, this bit enables a local interrupt to be set when a buffer full condition is detected on the bus.

Endpoint A~F Data Available count register (EPA_DATA_CNT~ EPF_DATA_CNT)

Register	Address	R/W	Description	Default Value
EPA_DATA_CNT	0xB000_6070	R	Endpoint A Data Available count register	0x0000_0000
EPB_DATA_CNT	0xB000_6098	R	Endpoint B Data Available count register	0x0000_0000
EPC_DATA_CNT	0xB000_60C0	R	Endpoint C Data Available count register	0x0000_0000
EPD_DATA_CNT	0xB000_60E8	R	Endpoint D Data Available count register	0x0000_0000
EPE_DATA_CNT	0xB000_6110	R	Endpoint E Data Available count register	0x0000_0000
EPF_DATA_CNT	0xB000_6138	R	Endpoint F Data Available count register	0x0000_0000

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

Bits	Descriptions	
[30:16]	**DMA_LOOP**	This register is the remaining DMA loop to complete. Each loop means 32-byte transfer.
[15:0]	**DATA_CNT**	For an OUT / IN endpoint, this register returns the number of valid bytes in the endpoint packet buffer.

Endpoint A~F Response Set/Clear Register (EPA_RSP_SC~ EPF_RSP_SC)

Register	Address	R/W	Description	Default Value
EPA_RSP_SC	0xB000_6074	R/W	Endpoint A Response Set/Clear Register	0x0000_0000
EPB_RSP_SC	0xB000_609C	R/W	Endpoint B Response Set/Clear Register	0x0000_0000
EPC_RSP_SC	0xB000_60C4	R/W	Endpoint C Response Set/Clear Register	0x0000_0000
EPD_RSP_SC	0xB000_60EC	R/W	Endpoint D Response Set/Clear Register	0x0000_0000
EPE_RSP_SC	0xB000_6114	R/W	Endpoint E Response Set/Clear Register	0x0000_0000
EPF_RSP_SC	0xB000_613C	R/W	Endpoint F Response Set/Clear Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
DIS_BUF	PK_END	ZEROLEN	HALT	TOGGLE	MODE		BUF_FLUSH

Bits	Descriptions	
[7]	**DIS_BUF**	**Disable Buffer** This bit is used to disable buffer (set buffer size to 1) when received a bulk-out short packet.
[6]	**PK_END**	**Packet End.** This bit is applicable only in case of Auto-Validate Method. This bit is set to validate any remaining data in the buffer which is not equal to the MPS of the endpoint, and happens to be the last transfer.
[5]	**ZEROLEN**	**Zerolen In.** This bit is used to send a zero-length packet n response to an in-token. When this bit is set, a zero packet is sent to the host on reception of an in-token.
[4]	**HALT**	**Endpoint Halt.** This bit is used to send a stall handshake as response to the token from the host. When an Endpoint Set Feature (ep_halt) is detected by the local CPU, it must write a '1' to this bit.

<table>
<tr><td>[3]</td><td>TOGGLE</td><td>Endpoint Toggle.
This bit is used to clear the endpoint data toggle bit. Reading this bit returns the current state of the endpoint data toggle bit.
The local CPU may use this bit, to initialize the end-point’s toggle incase of reception of a Set Interface request or a Clear Feature (ep_halt) request from the host. Only when toggle bit is “1”, this bit can be written into the inversed write data bit[3].</td></tr>
<tr><td>[2:1]</td><td>MODE</td><td>Mode.
These two bits decide the mode of operation of the in-endpoint.
<table>
<tr><td>MODE[2:1]</td><td>Mode Description</td></tr>
<tr><td>2’b00</td><td>Auto-Validate Mode</td></tr>
<tr><td>2’b01</td><td>Manual-Validate Mode</td></tr>
<tr><td>2’b10</td><td>Fly Mode</td></tr>
<tr><td>2’b11</td><td>Reserved.</td></tr>
</table>
These bits are not valid for an out-endpoint. The auto validate mode will be activated when the reserved mode is selected.
(These modes are explained detailed in later sections)</td></tr>
<tr><td>[0]</td><td>BUF_FLUSH</td><td>Buffer Flush.
Writing a 1 to this bit causes the packet buffer to be flushed and the corresponding EP_AVAIL register to be cleared. This bit is self-clearing. This bit should always be written after a configuration event.</td></tr>
</table>

Endpoint A~F Maximum Packet Size Register (EPA_MPS~ EPF_MPS)

Register	Address	R/W	Description	Default Value
EPA_MPS	0xB000_6078	R/W	Endpoint A Maximum Packet Size Register	0x0000_0000
EPB_MPS	0xB000_60A0	R/W	Endpoint B Maximum Packet Size Register	0x0000_0000
EPC_MPS	0xB000_60C8	R/W	Endpoint C Maximum Packet Size Register	0x0000_0000
EPD_MPS	0xB000_60F0	R/W	Endpoint D Maximum Packet Size Register	0x0000_0000
EPE_MPS	0xB000_6118	R/W	Endpoint E Maximum Packet Size Register	0x0000_0000
EPF_MPS	0xB000_6140	R/W	Endpoint F Maximum Packet Size Register	0x0000_0000

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

Bits	Descriptions	
[10:0]	**EP_MPS**	**Endpoint Maximum Packet Size.** This field determines the Endpoint Maximum Packet Size.

Endpoint A~F Transfer Count Register (EPA_TRF_CNT~ EPF_TRF_CNT)

Register	Address	R/W	Description	Default Value
EPA_TRF_CNT	0xB000_607C	R/W	Endpoint A Transfer Count Register	0x0000_0000
EPB_TRF_CNT	0xB000_60A4	R/W	Endpoint B Transfer Count Register	0x0000_0000
EPC_TRF_CNT	0xB000_60CC	R/W	Endpoint C Transfer Count Register	0x0000_0000
EPD_TRF_CNT	0xB000_60F4	R/W	Endpoint D Transfer Count Register	0x0000_0000
EPE_TRF_CNT	0xB000_611C	R/W	Endpoint E Transfer Count Register	0x0000_0000
EPF_TRF_CNT	0xB000_6144	R/W	Endpoint F Transfer Count Register	0x0000_0000

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

Bits	Descriptions	
[10:0]	**EP_TRF_CNT**	For IN endpoints, this field determines the total number of bytes to be sent to the host in case of manual validation method. For OUT endpoints, this field has no effect

Endpoint A~F Configuration Register (EPA_CFG~ EPF_CFG)

Register	Address	R/W	Description	Default Value
EPA_CFG	0xB000_6080	R/W	Endpoint A Configuration Register	0x0000_0012
EPB_CFG	0xB000_60A8	R/W	Endpoint B Configuration Register	0x0000_0022
EPC_CFG	0xB000_60D0	R/W	Endpoint C Configuration Register	0x0000_0032
EPD_CFG	0xB000_60F8	R/W	Endpoint D Configuration Register	0x0000_0042
EPE_CFG	0xB000_6120	R/W	Endpoint E Configuration Register	0x0000_0052
EPF_CFG	0xB000_6148	R/W	Endpoint F Configuration Register	0x0000_0062

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

Bits	Descriptions	
[9:8]	**EP_MULT**	**MULT Field**. This field indicates number of transactions to be carried out in one single micro frame. [9:8] / Description: 0x00 One transaction; 0x01 Reserved; 0x10 Reserved; 0x11 Invalid
[7:4]	**EP_NUM**	**Endpoint Number**. This field selects the number of the endpoint. Valid numbers 1 to 15.
[3]	**EP_DIR**	**Endpoint Direction**. EP_DIR = 0 - OUT EP (Host OUT to Device) EP_DIR = 1- IN EP (Host IN to Device) Note that a maximum of one OUT and IN endpoint is allowed for each endpoint number.

[2:1]	**EP_TYPE**	**Endpoint Type**. This field selects the type of this endpoint. Endpoint 0 is forced to a Control type. **[2:1] / Description** 0x00 Reserved 0x01 Bulk 0x10 Interrupt 0x11 Isochronous
[0]	**EP_VALID**	**Endpoint Valid**. When set, this bit enables this endpoint. This bit has no effect on Endpoint 0, which is always enabled.

Endpoint A~F RAM Start Address Register (EPA_START_ADDR~ EPF_START_ADDR)

Register	Address	R/W	Description	Default Value
EPA_START_ADDR	0xB000_6084	R/W	Endpoint A RAM Start Address Register	0x0000_0000
EPB_START_ADDR	0xB000_60AC	R/W	Endpoint B RAM Start Address Register	0x0000_0000
EPC_START_ADDR	0xB000_60D4	R/W	Endpoint C RAM Start Address Register	0x0000_0000
EPD_START_ADDR	0xB000_60FC	R/W	Endpoint D RAM Start Address Register	0x0000_0000
EPE_START_ADDR	0xB000_6124	R/W	Endpoint E RAM Start Address Register	0x0000_0000
EPF_START_ADDR	0xB000_614C	R/W	Endpoint F RAM Start Address Register	0x0000_0000

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

Bits	Descriptions	
[10:0]	**EP_START_ADDR**	This is the start-address of the RAM space allocated for the endpoint A~F.

Endpoint A~F RAM End Address Register (EPA_END_ADDR~ EPF_END_ADDR)

Register	Address	R/W	Description	Default Value
EPA_END_ADDR	0xB000_6088	R/W	Endpoint A RAM End Address Register	0x0000_0000
EPB_END_ADDR	0xB000_60B0	R/W	Endpoint B RAM End Address Register	0x0000_0000
EPC_END_ADDR	0xB000_60D8	R/W	Endpoint C RAM End Address Register	0x0000_0000
EPD_END_ADDR	0xB000_6100	R/W	Endpoint D RAM End Address Register	0x0000_0000
EPE_END_ADDR	0xB000_6128	R/W	Endpoint E RAM End Address Register	0x0000_0000
EPF_END_ADDR	0xB000_6150	R/W	Endpoint F RAM End Address Register	0x0000_0000

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

Bits	Descriptions	
[10:0]	**EP_END_ADDR**	This is the end-address of the RAM space allocated for the endpoint A~F.

USB Address Register (USB_DMA_ADDR)

Register	Address	R/W	Description	Default Value
USB_DMA_ADDR	0xB000_6700	R/W	USB DMA address register	0x0000_0000

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

USB_DMA_ADDR

Bits	Descriptions	
[31:0]	**USB_DMA_ADDR**	It specifies the address from which the DMA has to read / write. The address must WORD (32-bit) aligned.

USB PHY Control (USB_PHY_CTL)

Register	Address	R/W	Description	Default Value
USB_PHY_CTL	0xB000_6704	R/W	USB PHY control register	0x0000_0260

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

Bits	Descriptions	
[9]	Phy_suspend	Set this bit low will cause USB PHY suspended.

7.9 DMA Controller (DMAC)

The DMA Controller provides a DMA (Direct Memory Access) function for FMI to exchange data between system memory (ex. SDRAM) and shared buffer (one 2048 bytes). Software just simply fills in the starting address and enables DMAC, and then you can let DMAC to handle the data transfer automatically.
There is one 2048 bytes shared buffer inside DMAC, separate into four 512 bytes ping-pong FIFO. It can provide multi-block transfers using ping-pong mechanism for FMI. Software can access these shared buffers directly when FMI are not in busy.

Features:

- Support single DMA channel
- Support hardware Scatter-Getter function
- One 2048 bytes shared buffer is embedded
- Automatic arbitration of DMA request for FMI

7.9.1 DMA Controller Registers Map

R: read only, **W**: write only, **R/W**: both read and write

Register	Offset	R/W	Description	Reset Value
Shared Buffer				
FB_0 …… FB_511	0XB000_C000 …… 0xB000_C7FC	R/W	Shared Buffer (FIFO)	N/A
DMAC Registers				
DMACCSR	0xB000_C800	R/W	DMAC Control and Status Register	0x0000_0000
DMACSAR2	0xB000_C808	R/W	DMAC Transfer Starting Address Register 2	0x0000_0000
DMACBCR	0xB000_C80C	R	DMAC Transfer Byte Count Register	0x0000_0000
DMACIER	0xB000_C810	R/W	DMAC Interrupt Enable Register	0x0000_0001
DMACISR	0xB000_C814	R/W	DMAC Interrupt Status Register	0x0000_0000

7.9.2 DMAC Registers

DMAC Control and Status Register (DMACCSR)

Register	Offset	R/W	Description	Reset Value
DMACCSR	0xB000_C800	R/W	DMAC Control and Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved						FMI_BUSY	Reserved
7	6	5	4	3	2	1	0
Reserved				SG_EN2	SG_EN1	SW_RST	DMACEN

Bits	Descriptions	
[9]	**FMI_BUSY**	**FMI DMA Transfer is in progress** This bit indicates if FMI is granted and doing DMA transfer or not. 0 = FMI DMA transfer is not in progress. 1 = FMI DMA transfer is in progress.
[3]	**SG_EN2**	**Enable Scatter-Getter Function for FMI** Enable DMA scatter-getter function or not. • 0 = Normal operation. DMAC will treat the starting address in **DMACSAR2** as starting pointer of a single block memory. • 1 = Enable scatter-getter operation. DMAC will treat the starting address in **DMACSAR2** as a starting address of Physical Address Descriptor (PAD) table. The format of these PADs will be described later.

[2]	**SG_EN1**	**Enable Scatter-Getter Function for ATAPI** Enable DMA scatter-getter function or not. • 0 = Normal operation. DMAC will treat the starting address in **DMACSAR1** as starting pointer of a single block memory. • 1 = Enable scatter-getter operation. DMAC will treat the starting address in **DMACSAR1** as a starting address of Physical Address Descriptor (PAD) table. The format of these PADs will be described later.
[1]	**SW_RST**	**Software Engine Reset** 0 = Writing 0 to this bit has no effect. 1 = Writing 1 to this bit will reset the internal state machine and pointers. The contents of control register will not be cleared. This bit will auto clear after few clock cycles.
[0]	**DMACEN**	**DMAC Engine Enable** Setting this bit to 1 enables DMAC's operation. If this bit is cleared, DMAC will ignore all DMA request from FMI and force Bus Master into IDLE state. 0 = Disable DMAC. 1 = Enable DMAC. **NOTE:** If target abort is occurred, DMACEN will be cleared.

DMAC Transfer Starting Address Register 2 (DMACSAR2)

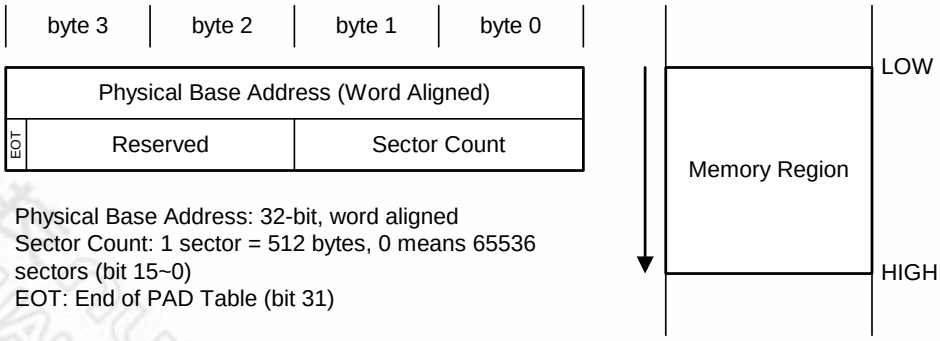
Register	Offset	R/W	Description	Reset Value
DMACSAR2	0xB000_C808	R/W	DMAC Transfer Starting Address Register 2	0x0000_0000

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

Bits	Descriptions	
[31:0]	DMACSA	**DMA Transfer Starting Address for FMI** This field indicates a 32-bit starting address of system memory (SRAM/SDRAM) for DMAC to retrieve or fill in data (for FMI engine). If DMAC is not in normal mode, this field will be interpreted as a starting address of Physical Address Descriptor (PAD) table.

NOTE: Starting address should be word alignment, for example, 0x0000_0000, 0x0000_0004…

The format of PAD table must like below. Note that the total sector count of all PADs must be equal to or greater than the sector count filled in FMI engine. EOT should be set to 1 in the last descriptor.



DMAC Transfer Byte Count Register (DMACBCR)

Register	Offset	R/W	Description	Reset Value
DMACBCR	0xB000_C80C	R	DMAC Transfer Byte Count Register	0x0000_0000

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

Bits	Descriptions	
[25:0]	BCNT	**DMA Transfer Byte Count (Read Only)** This field indicates the remained byte count of DMAC transfer. The value of this field is valid only when FMI is busy; otherwise, it is zero.

DMAC Interrupt Enable Register (DMACIER)

Register	Offset	R/W	Description	Reset Value
DMACIER	0xB000_C810	R/W	DMAC Interrupt Enable Register	0x0000_0001

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

Bits	Descriptions	
[1]	WEOT_IE	**Wrong EOT Encountered Interrupt Enable** • 0 = Disable interrupt generation when wrong EOT is encountered. • 1 = Enable interrupt generation when wrong EOT is encountered.
[0]	TABORT_IE	**DMA Read/Write Target Abort Interrupt Enable** 0 = Disable target abort interrupt generation during DMA transfer. 1 = Enable target abort interrupt generation during DMA transfer.

DMAC Interrupt Status Register (DMACISR)

Register	Offset	R/W	Description	Reset Value
DMACISR	0xB000_C814	R/W	DMAC Interrupt Status Register	0x0000_0000

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

Bits	Descriptions	
[1]	**WEOT_IF**	**Wrong EOT Encountered Interrupt Flag** When DMA Scatter-Getter function is enabled, and EOT of the descriptor is encountered before DMA transfer finished (that means the total sector count of all PAD is less than the sector count of FMI), this bit will be set. 0 = No EOT encountered before DMA transfer finished. 1 = EOT encountered before DMA transfer finished. **NOTE:** This bit is read only, but can be cleared by writing '1' to it.
[0]	**TABORT_IF**	**DMA Read/Write Target Abort Interrupt Flag** 0 = No bus ERROR response received. 1 = Bus ERROR response received. **NOTE:** This bit is read only, but can be cleared by writing '1' to it.

NOTE: When DMAC's bus master received ERROR response, it means that target abort is happened. DMAC will stop transfer and respond this event to software and FMI; then go to IDLE state. When target abort occurred or WEOT_IF is set, suggest software reset DMAC and IP, and then transfer those data again.

7.10 Flash Memory Interface Controller (FMI)

The Flash Memory Interface (FMI) supports Secure Digital (SD, SDIO & MMC) and Memory Stick (Memory stick PRO). FMI is cooperated with DMAC to provide a fast data transfer between system memory and cards. There is one single 2048-byte buffer embedded in DMAC for temporary data storage. Due to DMAC only has single channel, that means only one interface can be active at the same time.

Feature:

- Interface with DMAC for register read/write and data transfer
- 3 interfaces are provided: Secure Digital(2.0)/MMC(4.2) and Memory Stick/Memory Stick PRO
- Using single 2048-byte shared buffer for data exchange between system memory and cards

7.10.1 FMI Controller Registers Map

R: read only, **W**: write only, **R/W**: both read and write

Register	Address	R/W	Description	Reset Value
FMI Global Registers (FMI_BA = 0xB000_D000)				
FMICSR	0xB000_D000	R/W	Global Control and Status Register	0x0000_0000
FMIIER	0xB000_D004	R/W	Global Interrupt Control Register	0x0000_0001
FMIISR	0xB000_D008	R/W	Global Interrupt Status Register	0x0000_0000
Secure Digital Registers				
SDCSR	0xB000_D020	R/W	SD Control and Status Register	0x0101_0000
SDARG	0xB000_D024	R/W	SD Command Argument Register	0x0000_0000
SDIER	0xB000_D028	R/W	SD Interrupt Control Register	0x0000_0000
SDISR	0xB000_D02C	R/W	SD Interrupt Status Register	0x000X_008C
SDRSP0	0xB000_D030	R	SD Receiving Response Token Register 0	0x0000_0000
SDRSP1	0xB000_D034	R	SD Receiving Response Token Register 1	0x0000_0000
SDBLEN	0xB000_D038	R/W	SD Block Length Register	0x0000_01FF
Memory Stick Registers				
MSCSR	0xB000_D060	R/W	Memory Stick Control and Status Register	0x0000_0008
MSIER	0xB000_D064	R/W	Memory Stick Interrupt Control Register	0x0000_0000
MSISR	0xB000_D068	R/W	Memory Stick Interrupt Status Register	0x0000_0000
MSBUF1	0xB000_D06C	R/W	Memory Stick Register Buffer 1	0x0000_0000
MSBUF2	0xB000_D070	R/W	Memory Stick Register Buffer 2	0x0000_0000

7.10.2 Register Details

Global Control and Status Register (FMICSR)

Register	Address	R/W	Description	Reset Value
FMICSR	0xB000_D000	R/W	Global Control and Status Register	0x0000_0000

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

Bits	Descriptions	
[2]	**MS_EN**	**Memory Stick Functionality Enable** 0 = Disable MS functionality of FMI. 1 = Enable MS functionality of FMI.
[1]	**SD_EN**	**Secure Digital Functionality Enable** 0 = Disable SD functionality of FMI. 1 = Enable SD functionality of FMI.
[0]	**SW_RST**	**Software Engine Reset** 0 = Writing 0 to this bit has no effect. 1 = Writing 1 to this bit will reset all FMI engines. The contents of control register will not be cleared. This bit will auto clear after few clock cycles.

NOTE: Software can enable only one engine at one time, or FMI will work abnormal.

Global Interrupt Control Register (FMIIER)

Register	Address	R/W	Description	Reset Value
FMIIER	0xB000_D004	R/W	Global Interrupt Control Register	0x0000_0001

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

Bits	Descriptions	
[0]	DTA_IE	**DMAC READ/WRITE Target Abort Interrupt Enable** 0 = Disable DMAC READ/WRITE target abort interrupt generation. 1 = Enable DMAC READ/WRITE target abort interrupt generation.

Global Interrupt Status Register (FMIISR)

Register	Address	R/W	Description	Reset Value
FMIISR	0xB000_D008	R/W	Global Interrupt Status Register	0x0000_0000

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

Bits	Descriptions	
[0]	DTA_IF	**DMAC READ/WRITE Target Abort Interrupt Flag (Read Only)** This bit indicates DMAC received an ERROR response from internal AHB bus during DMA read/write operation. When Target Abort is occurred, please reset all engine. 0 = No bus ERROR response received. 1 = Bus ERROR response received. **NOTE:** This bit is read only, but can be cleared by writing '1' to it.

NOTE: No matter interrupt enable is turn on or not, the interrupt flag will be set when target condition is occurred.

SD Control and Status Register (SDCSR)

Register	Address	R/W	Description	Reset Value
SDCSR	0xB000_D020	R/W	SD Control and Status Register	0x0101_0000

31	30	29	28	27	26	25	24
CLK_KEEP1	SDPORT		Reserved	SDNWR			
23	22	21	20	19	18	17	16
BLK_CNT							
15	14	13	12	11	10	9	8
DBW	SW_RST	CMD_CODE					
7	6	5	4	3	2	1	0
CLK_KEEP0	CLK8_OE	CLK74_OE	R2_EN	DO_EN	DI_EN	RI_EN	CO_EN

Bits	Descriptions	
[31]	**CLK_KEEP1**	**SD Clock Enable for Port 1** 0 = Disable SD clock generation. 1 = SD clock always keeps free running.
[30:29]	**SDPORT**	**SD Port Selection** 00 = Port 0 is selected. 10 = Port 1 is selected. X1 = Reserved
[27:24]	**SDNWR**	**N_{WR} Parameter for Block Write Operation** This value indicates the N_{WR} parameter for data block write operation in clock counts. The actual clock cycle will be SDNWR+1.
[23:16]	**BLK_CNT**	**Block Counts to Be Transferred or Received** This field contains the block counts for data-in and data-out transfer. For **READ_MULTIPLE_BLOCK** and **WRITE_MULTIPLE_BLOCK** command, software can use this function to accelerate data transfer and improve performance. Note that only when SDBLEN=0x1FF, this field is valid. Otherwise, blob counts will be set to 1 inside SD host engine. **NOTE:** Value 0x0 in this field means 256.
[15]	**DBW**	**SD Data Bus Width** 0 = Data bus width is 1-bit. 1 = Data bus width is 4-bit.

[14]	**SW_RST**	**Software Engine Reset** 0 = Writing 0 to this bit has no effect. 1 = Writing 1 to this bit will reset the internal state machine and counters. The contents of control register will not be cleared (but RI_EN, DI_EN, DO_EN and R2_EN will be cleared). This bit will be auto cleared after few clock cycles.
[13:8]	**CMD_CODE**	**SD Command Code** This register contains the SD command code (0x00 – 0x3F).
[7]	**CLK_KEEP0**	**SD Clock Enable for Port 0** 0 = Disable SD clock generation. 1 = SD clock always keeps free running.
[6]	**CLK8_OE**	**Generating 8 Clock Cycles Output Enable** 0 = No effect. 1 = Enable, SD host will output 8 clock cycles. **NOTE:** When this operation is finished, this bit will be cleared automatically.
[5]	**CLK74_OE**	**Initial 74 Clock Cycles Output Enable** 0 = No effect. 1 = Enable, SD host will output 74 clock cycles to SD card. **NOTE:** When this operation is finished, this bit will be cleared automatically.
[4]	**R2_EN**	**Response R2 Input Enable** 0 = No effect. (Please use SDCSR[SW_RST] to clear this bit.) 1 = Enable, SD host will wait to receive a response R2 from SD card and store the response data into DMAC's flash buffer (exclude CRC-7). **NOTE:** When the R2 response operation is finished, this bit will be cleared automatically.
[3]	**DO_EN**	**Data Output Enable** 0 = No effect. (Please use SDCSR[SW_RST] to clear this bit.) 1 = Enable, SD host will transfer block data and the CRC-16 value to SD card. **NOTE:** When the data output operation is finished, this bit will be cleared automatically.

[2]	**DI_EN**	**Data Input Enable** 0 = No effect. (Please use SDCSR[SW_RST] to clear this bit.) 1 = Enable, SD host will wait to receive block data and the CRC-16 value from SD card. **NOTE:** When the data input operation is finished, this bit will be cleared automatically.
[1]	**RI_EN**	**Response Input Enable** 0 = No effect. (Please use SDCSR[SW_RST] to clear this bit.) 1 = Enable, SD host will wait to receive a response from SD card. **NOTE:** When the response input operation is finished, this bit will be cleared automatically.
[0]	**CO_EN**	**Command Output Enable** 0 = No effect. 1 = Enable, SD host will output a command to SD card. **NOTE:** When the command output operation is finished, this bit will be cleared automatically.

SD Command Argument Register (SDARG)

Register	Address	R/W	Description	Reset Value
SDARG	0xB000_D024	R/W	SD Command Argument Register	0x0000_0000

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

Bits	Descriptions	
[31:0]	SD_CMD_ARG	**SD Command Argument** This register contains a 32-bit value specifies the argument of SD command from host controller to SD card.

SD Interrupt Control Register (SDIER)

Register	Address	R/W	Description	Reset Value
SDIER	0xB000_D028	R/W	SD Interrupt Control Register	0x0000_0000

31	30	29	28	27	26	25	24
CD1SRC	CD0SRC	Reserved					
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved	WKUP_EN	DITO_IE	RITO_IE	SDIO1_IE	SDIO0_IE	CD1_IE	CD0_IE
7	6	5	4	3	2	1	0
Reserved						CRC_IE	BLKD_IE

Bits	Descriptions	
[31]	**CD1SRC**	**SD1 Card Detect Source Selection** • 0 = From SD1 card's DAT3 pin. • 1 = From GPIO pin.
[30]	**CD0SRC**	**SD0 Card Detect Source Selection** • 0 = From SD0 card's DAT3 pin. • 1 = From GPIO pin.
[14]	**WKUP_EN**	**Wake-Up Signal Generating Enable** Enable/Disable wake-up signal generating of SD host when SDIO card (current using) issues an interrupt (wake-up) via DAT[1] to host. 0 = Disable. 1 = Enable.
[13]	**DITO_IE**	**Data Input Time-out Interrupt Enable** Enable/Disable interrupt generation of SD controller when data input time-out. Time-out value is specified at **SDTMOUT**. 0 = Disable. 1 = Enable.
[12]	**RITO_IE**	**Response Time-out Interrupt Enable** Enable/Disable interrupt generation of SD controller when receiving response or R2 time-out. Time-out value is specified at **SDTMOUT**. 0 = Disable. 1 = Enable.

[11]	**SDIO1_IE**	**SDIO Interrupt Enable for Port 1** Enable/Disable interrupt generation of SD host when SDIO card 1 issues an interrupt via DAT[1] to host. 0 = Disable. 1 = Enable.
[10]	**SDIO0_IE**	**SDIO Interrupt Enable for Port 0** Enable/Disable interrupt generation of SD host when SDIO card 0 issues an interrupt via DAT[1] to host. 0 = Disable. 1 = Enable.
[9]	**CD1_IE**	**SD1 Card Detection Interrupt Enable** Enable/Disable interrupt generation of SD controller when card 1 is inserted or removed. 0 = Disable. 1 = Enable.
[8]	**CD0_IE**	**SD0 Card Detection Interrupt Enable** Enable/Disable interrupt generation of SD controller when card 0 is inserted or removed. 0 = Disable. 1 = Enable.
[1]	**CRC_IE**	**CRC-7, CRC-16 and CRC Status Error Interrupt Enable** 0 = SD host will not generate interrupt when CRC-7, CRC-16 and CRC status is error. 1 = SD host will generate interrupt when CRC-7, CRC-16 and CRC status is error.
[0]	**BLKD_IE**	**Block Transfer Done Interrupt Enable** 0 = SD host will not generate interrupt when data-in (out) transfer done. 1 = SD host will generate interrupt when data-in (out) transfer done.

SD Interrupt Status Register (SDISR)

Register	Address	R/W	Description	Reset Value
SDISR	0xB000_D02C	R/W	SD Interrupt Status Register	0x000X_008C

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved				SD1DAT1	SD0DAT1	CDPS1	CDPS0
15	14	13	12	11	10	9	8
Reserved		DITO_IF	RITO_IF	SDIO1_IF	SDIO0_IF	CD1_IF	CD0_IF
7	6	5	4	3	2	1	0
SDDAT0	CRCSTAT			CRC-16	CRC-7	CRC_IF	BLKD_IF

Bits	Descriptions	
[19]	SD1DAT1	**DAT1 Pin Status of SD1 (Read Only)** This bit is the DAT1 pin status of SD1.
[18]	SD0DAT1	**DAT1 Pin Status of SD0 (Read Only)** This bit is the DAT1 pin status of SD0.
[17]	CDPS1	**Card Detect Pin Status of SD1 (Read Only)** This bit is the DAT3 pin status of SD1, and it is using for card detection. When there is a card inserted in or removed from SD1, software should check this bit to confirm if there is really a card insertion or remove.
[16]	CDPS0	**Card Detect Pin Status of SD0 (Read Only)** This bit is the DAT3 pin status of SD0, and it is using for card detection. When there is a card inserted in or removed from SD0, software should check this bit to confirm if there is really a card insertion or remove.
[13]	DITO_IF	**Data Input Time-out Interrupt Flag (Read Only)** This bit indicates that SD host counts to time-out value when receiving data (waiting start bit). 0 = Not time-out. 1 = Data input time-out. **NOTE:** This bit is read only, but can be cleared by writing '1' to it.

[12]	**RITO_IF**	**Response Time-out Interrupt Flag (Read Only)** This bit indicates that SD host counts to time-out value when receiving response or R2 (waiting start bit). 0 = Not time-out. 1 = Response time-out. **NOTE:** This bit is read only, but can be cleared by writing '1' to it.
[11]	**SDIO1_IF**	**SDIO 1 Interrupt Flag (Read Only)** This bit indicates that SDIO card 1 issues an interrupt to host. 0 = No interrupt is issued by SDIO card 1. 1 = An interrupt is issued by SDIO card 1. **NOTE:** This bit is read only, but can be cleared by writing '1' to it.
[10]	**SDIO0_IF**	**SDIO 0 Interrupt Flag (Read Only)** This bit indicates that SDIO card 0 issues an interrupt to host. 0 = No interrupt is issued by SDIO card 0. 1 = An interrupt is issued by SDIO card 0. **NOTE:** This bit is read only, but can be cleared by writing '1' to it.
[9]	**CD1_IF**	**SD1 Card Detection Interrupt Flag (Read Only)** This bit indicates that SD card 1 is inserted or removed. Only if SDIER[CD1_IE] is set to 1, this bit is active. 0 = No card is inserted or removed. 1 = There is a card inserted in or removed from SD1. **NOTE:** This bit is read only, but can be cleared by writing '1' to it.
[8]	**CD0_IF**	**SD0 Card Detection Interrupt Flag (Read Only)** This bit indicates that SD card 0 is inserted or removed. Only if SDIER[CD0_IE] is set to 1, this bit is active. 0 = No card is inserted or removed. 1 = There is a card inserted in or removed from SD0. **NOTE:** This bit is read only, but can be cleared by writing '1' to it.
[7]	**SDDAT0**	**DAT0 Pin Status of Current Selected SD (Read Only)** This bit is the DAT0 pin status of current selected SD port.

[6:4]	**CRCSTAT**	**CRC Status Value of Data-out Transfer (Read Only)** SD host will record CRC status of data-out transfer. Software could use this value to identify what type of error is during data-out transfer. 010 = Positive CRC status. 101 = Negative CRC status 111 = SD card programming error occurs.
[3]	**CRC-16**	**CRC-16 Check Status of Data-in Transfer (Read Only)** SD host will check CRC-16 correctness after data-in transfer. 0 = Fault. 1 = OK.
[2]	**CRC-7**	**CRC-7 Check Status (Read Only)** SD host will check CRC-7 correctness during each response in. If that response does not contain CRC-7 information (R3), then software should turn off SDIER[CRC_IE] and ignore this bit. 0 = Fault. 1 = OK.
[1]	**CRC_IF**	**CRC-7, CRC-16 and CRC Status Error Interrupt Flag (Read Only)** This bit indicates that SD host has occurred CRC error during response in, data-in or data-out (CRC status error) transfer. When CRC error is occurred, software should reset SD engine. Some response (ex. R3) doesn't have CRC-7 information with it; SD host will still calculate CRC-7, get CRC error and set this flag. In this condition, software should ignore CRC error and clears this bit manually. 0 = No CRC error is occurred. 1 = CRC error is occurred. **NOTE:** This bit is read only, but can be cleared by writing '1' to it.
[0]	**BLKD_IF**	**Block Transfer Done Interrupt Flag (Read Only)** This bit indicates that SD host has finished data-in or data-out block transfer. If there is a CRC-16 error or incorrect CRC status during multiple block data transfer, the transfer will be broken and this bit will be set. 0 = Not finished yet. 1 = Done. **NOTE:** This bit is read only, but can be cleared by writing '1' to it.

SD Receiving Response Token Register 0 (SDRSP0)

Register	Address	R/W	Description	Reset Value
SDRSP0	0xB000_D030	R	SD Receiving Response Token Register 0	0x0000_0000

31	30	29	28	27	26	25	24
SD_RSP_TK0							
23	22	21	20	19	18	17	16
SD_RSP_TK0							
15	14	13	12	11	10	9	8
SD_RSP_TK0							
7	6	5	4	3	2	1	0
SD_RSP_TK0							

Bits	Descriptions	
[31:0]	SD_RSP_TK0	**SD Receiving Response Token 0** SD host controller will receive a response token for getting a reply from SD card when SDCSR[RI_EN] is set. This field contains response bit 47-16 of the response token.

SD Receiving Response Token Register 1 (SDRSP1)

Register	Address	R/W	Description	Reset Value
SDRSP1	0xB000_D034	R	SD Receiving Response Token Register 1	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
SD_RSP_TK1							

Bits	Descriptions	
[7:0]	SD_RSP_TK1	**SD Receiving Response Token 1** SD host controller will receive a response token for getting a reply from SD card when SDCSR[RI_EN] is set. This register contains the bit 15-8 of the response token.

SD Block Length Register (SDBLEN)

Register	Address	R/W	Description	Reset Value
SDBLEN	0xB000_D038	R/W	SD Block Length Register	0x0000_01FF

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

Bits	Descriptions	
[8:0]	SDBLEN	**SD BLOCK LENGTH in Byte Unit** A 9-bit value specifies the SD transfer byte count. The actual byte count is equal to SDBLEN+1.

SD Response/Data-in Time-out Register (SDTMOUT)

Register	Offset	R/W	Description	Reset Value
SDTMOUT	0xB000_D03C	R/W	SD Response/Data-in Time-out Register	0x0000_0000

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

Bits	Descriptions	
[23:0]	SDTMOUT	**SD Response/Data-in Time-out Value** A 24-bit value specifies the time-out counts of response and data input. SD host controller will wait start bit of response or data-in until this value reached. The time period is depended on SD engine clock frequency. Do not write a small number into this field, or you may never get response or data due to time-out. **NOTE:** Fill 0x0 into this field will disable hardware time-out function.

Memory Stick Control and Status Register (MSCSR)

Register	Address	R/W	Description	Reset Value
MSCSR	0xB000_D060	R/W	Memory Stick Control and Status Register	0x0000_0008

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved		MSPORT	DSIZE		DCNT		
15	14	13	12	11	10	9	8
Reserved				TPC			
7	6	5	4	3	2	1	0
Reserved				SERIAL	MSPRO	MS_GO	SW_RST

Bits	Descriptions	
[21]	MSPORT	**Memory Stick Port Selection** 0 = Port 0 is selected. 1 = Port 1 is selected.
[20:19]	DSIZE	**Data Size for Transfer (for Memory Stick PRO Only)** This field defines how many bytes should be transferred of following TPC codes. Data will be obtained from (stored in) DMAC's FIFO. READ_SHORT_DATA and WRITE_SHORT_DATA. 00 = 32 Bytes. 01 = 64 Bytes. 10 = 128 Bytes. 11 = 256 Bytes. **NOTE:** This field is invalid when other TPC codes are executed.
[18:16]	DCNT	**Data Count Number (in Byte Unit)** This field defines how many bytes should be transferred of following TPC codes. Data will be obtained from (stored in) MSBUF1 and MSBUF2. READ_REG, GET_INT, WRITE_REG, SET_R/W_REG_ADRS, SET_CMD and EX_SET_CMD. For example, when software wants to use SET_R/W_REG_ADRS, you should write 0x4 into this field; when you want to use SET_CMD, you should write 0x1 into this field, etc. **NOTE:** Value 0x0 means 8 bytes should be transferred, and it is the largest length this core can provide.

[11:8]	**TPC**	**TPC Code of the Packet** This field defines the TPC code of the packet which software wants to transfer. This core supports all TPC code of Memory Stick and Memory Stick PRO specification. The lower 4 bits of TPC (TPC Check Code) will be generated by hardware automatically.
[3]	**SERIAL**	**Serial or Parallel Mode** 0 = MS host is working at parallel mode. 1 = MS host is working at serial mode (Default).
[2]	**MSPRO**	**Memory Stick or Memory Stick PRO** 0 = Type of the card is Memory Stick. 1 = Type of the card is Memory Stick PRO.
[1]	**MS_GO**	**Trigger Memory Stick Core to Transfer Packet** 0 = Writing 0 to this bit has no effect. 1 = Trigger Memory Stick core to transfer packet. When TPC code is READ_REG, GET_INT, WRITE_REG, SET_R/W_REG_ADRS, SET_CMD or EX_SET_CMD, data will be obtained from (stored in) MSBUF1 and MSBUF2. When TPC code is READ_LONG_DATA (READ_PAGE_DATA), READ_SHORT_DATA, WRITE_LONG_DATA (WRITE_PAGE_DATA) or WRITE_SHORT_DATA, data will be obtained from (stored in) DMAC's FIFO.
[0]	**SW_RST**	**Software Engine Reset** 0 = Writing 0 to this bit has no effect. 1 = Writing 1 to this bit will reset the internal state machine and counters. The contents of control register will not be cleared. This bit will be auto cleared after few clock cycles.

Memory Stick Interrupt Control Register (MSIER)

Register	Address	R/W	Description	Reset Value
MSIER	0xB000_D064	R/W	Memory Stick Interrupt Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved						CD1_IE	CD0_IE
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved			CRC_IE	BSYTO_IE	INTTO_IE	MSINT_IE	PKT_IE

Bits	Descriptions	
[17]	CD1_IE	**MS Card Detection 1 Interrupt Enable** Enable/Disable Interrupt generation of MS controller when card 1 is inserted or removed. • 0 = Disable. • 1 = Enable.
[16]	CD0_IE	**MS Card Detection 0 Interrupt Enable** Enable/Disable Interrupt generation of MS controller when card 0 is inserted or removed. • 0 = Disable. • 1 = Enable.
[4]	CRC_IE	**CRC-16 Error Interrupt Enable** 0 = the core will not generate interrupt when CRC-16 is error. 1 = the core will generate interrupt when CRC-16 is error.
[3]	BSYTO_IE	**Busy to Ready Check Timeout Interrupt Enable** 0 = Disable Busy to Ready check timeout interrupt. 1 = Enable Busy to Ready check timeout interrupt.

[2]	**INTTO_IE**	**INT Response Timeout Interrupt Enable** 0 = Disable INT response timeout interrupt generation. 1 = Enable INT response timeout interrupt generation.
[1]	**MSINT_IE**	**Memory Stick Card's Interrupt Enable** 0 = the core will not generate interrupt when MS card generates INT. 1 = the core will generate interrupt when MS card generates INT. **NOTE:** Software should set MSIER[INTTO_IE] to '1' to enable INT detection function of the core, and set this bit to '1' if you want to get INT from MS card.
[0]	**PKT_IE**	**Packet Transfer Done Interrupt Enable** 0 = the core will not generate interrupt when packet transfer is done. 1 = the core will generate interrupt when packet transfer is done.

Memory Stick Interrupt Status Register (MSISR)

Register	Address	R/W	Description	Reset Value
MSISR	0xB000_D068	R/W	Memory Stick Interrupt Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved						CD1_	CD0_
23	22	21	20	19	18	17	16
Reserved						CD1_IF	CD0_IF
15	14	13	12	11	10	9	8
Reserved				CMDNK	BREQ	ERR	CED
7	6	5	4	3	2	1	0
Reserved			CRC_IF	BSYTO_IF	INTTO_IF	MSINT_IF	PKT_IF

Bits	Descriptions	
[25]	**CD1_**	**Pin Status of MS Card Detection 1 (Read Only)** This is the pin status of MS card detection 1. When there is a card insertion or removal, software should check this bit to confirm if it is really a card insertion or removal. NOTE: Software should perform de-bounce for card detection function.
[24]	**CD0_**	**Pin Status of MS Card Detection 0 (Read Only)** This is the pin status of MS card detection 0. When there is a card insertion or removal, software should check this bit to confirm if it is really a card insertion or removal. NOTE: Software should perform de-bounce for card detection function.
[17]	**CD1_IF**	**MS Card Detection 1 Interrupt Flag (Read Only)** This bit indicates that MS card 1 is inserted or removed. Only if MSIER[CD1_IE] is set, this bit is active; otherwise, this bit is invalid. 0 = No card is inserted or removed. 1 = There is a card inserted in or removed from MS1. **NOTE:** This bit is read only, but can be cleared by writing '1' to it.
[16]	**CD0_IF**	**MS Card Detection 0 Interrupt Flag (Read Only)** This bit indicates that MS card 0 is inserted or removed. Only if MSIER[CD0_IE] is set, this bit is active; otherwise, this bit is invalid. 0 = No card is inserted or removed. 1 = There is a card inserted in or removed from MS0. **NOTE:** This bit is read only, but can be cleared by writing '1' to it.

[11:8]	**CMDNK BREQ ERR CED**	**INT Status of Memory Stick PRO (Read Only)** These 4 bits indicates the INT status of Memory Stick PRO card (only for parallel mode). When MSIER[INTTO_IE] is set, the core will wait for INT signal from card. If the card is working at parallel mode; after INT is occurred (MSISR[MSINT_IF] is set), the contents of INT register can be informed by these bits. **NOTE:** These bits are valid in parallel mode only.
[4]	**CRC_IF**	**CRC-16 Error Interrupt Flag (Read Only)** When the packet transfer is done, the core will compare the value of CRC-16 which it calculated and received. If CRC-16 value is not the same, this flag will be set. The comparison executes only for READ packet. 0 = CRC-16 ok. 1 = CRC-16 failed. **NOTE:** This bit is read only, but can be cleared by writing '1' to it.
[3]	**BSYTO_IF**	**Busy to Ready Check Timeout Interrupt Flag (Read Only)** This bit indicates that the core cannot detect RDY signal on DATA[0] pin during Handshake State. It means some errors are occurred during packet transfer. The maximum timeout duration for RDY signal is 16 SCLKs. 0 = No RDY timeout occurred. 1 = RDY timeout occurred. **NOTE:** This bit is read only, but can be cleared by writing '1' to it.
[2]	**INTTO_IF**	**INT Response Timeout Interrupt Flag (Read Only)** This bit indicates that the core cannot detect INT signal of MS card after a period of time. In Memory Stick, the maximum period is 100ms. In Memory Stick PRO, the maximum period is 3500ms. If INT timeout is occurred, it means the card maybe malfunction. 0 = INT detection is not timeout. 1 = INT detection is timeout, no INT signal occurred. **NOTE:** This bit is read only, but can be cleared by writing '1' to it.

[1]	**MSINT_IF**	**Memory Stick Card's Interrupt Flag (Read Only)** Memory Stick will generate INT signal after some TPC codes are executed, ex. SET_CMD. This bit indicates that Memory Stick has generated INT signal after TPC code execution. This core will check INT for software only when MSIER[INTTO_IE] is set to '1', or this bit is invalid. 0 = No INT signal is detected. 1 = INT signal is detected. **NOTE:** This bit is read only, but can be cleared by writing '1' to it.
[0]	**PKT_IF**	**Packet Transfer Done Interrupt Flag (Read Only)** This bit indicates that the whole packet transfer is done. The four states of Memory Stick are BS1, BS2, BS3 and BS0. 0 = Packet transfer is not done yet. 1 = Packet transfer is done. **NOTE:** This bit is read only, but can be cleared by writing '1' to it.

NOTE: No matter interrupt enable is turn on or not, the interrupt flag will be set when target condition is occurred.

Memory Stick Register Buffer 1 (MSBUF1)

Memory Stick Register Buffer 2 (MSBUF2)

Register	Address	R/W	Description	Reset Value
MSBUF1 **MSBUF2**	0xB000_D06C 0xB000_D070	R/W	Memory Stick Register Buffer 1 Memory Stick Register Buffer 2	0x0000_0x0000

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

Bits	Descriptions	
[31:0]	**DATA**	**Data Content of Packet Transfer** This field contains the data of READ/WRITE TPC codes. When software uses following TPC codes, data will be obtained from (stored in) this field. READ_REG, GET_INT, WRITE_REG, SET_R/W_REG_ADRS, SET_CMD and EX_SET_CMD. This core will always send (store) data from MSB of MSBUF2. For example, if software wants to WRITE a packet with 1 byte data, you should put the data at MSBUF2[31:24] and write 0x1 into MSCSR[DCNT] then trigger the core. The order of transfer will be MSBUF2[31], MSBUF2[30] …, MSBUF2[24]. If you want to WRITE a packet with 6 bytes data, you should put the data at MSBUF2[31:0] and MSBUF1[31:16] and write 0x6 into MSCSR[DCNT] then trigger the core. The order of transfer will be MSBUF2[31:24], … MSBUF2[7:0], MSBUF1[31:24], MSBUF1[23:16]. The same order will be applied to READ packet.

MSBUF1	MSBUF2
BYTE 5	BYTE 1
BYTE 6	BYTE 2
BYTE 7	BYTE 3
BYTE 8	BYTE 4

7.11 UART Controller

The Universal Asynchronous Receiver/Transmitter (UART) performs a serial-to-parallel conversion on data characters received from the peripheral, and a parallel-to-serial conversion on data characters received from the CPU. There are five UART blocks and accessory logic in this chip.

7.11.1 UART Feature Description

7.11.1.1 UART0

UART0 is a general UART block.

UART0	
Clock Source	External Crystal
UART Type	General UART
FIFO Number	16-byte receiving FIFO and 16 byte transmitting FIFO
Modem Function	None
Accessory Function	None
I/O pin	TXD0, RXD0

7.11.1.2 UART1

UART1 is a high speed UART. The FIFO has 64-byte for receiving and 64-byte for transmitting. The clock source is programmable in chip clock generator.

UART1	
Clock Source	External Crystal or internal PLL (Programmable)
UART Type	High speed UART
FIFO Number	64-byte receiving FIFO and 64 byte transmitting FIFO
Modem Function	None
Accessory Function	None
I/O pin	TXD1, RXD1

7.11.2 UART Control Registers Map

R: read only, **W**: write only, **R/W**: both read and write, **C**: Only value 0 can be written

Register	Offset	R/W	Description	Condition	Reset Value
UART0 :					
RBR	0xB800_0000	R	Receive Buffer Register	DLAB = 0	Undefined
THR	0xB800_0000	W	Transmit Holding Register	DLAB = 0	Undefined
IER	0xB800_0004	R/W	Interrupt Enable Register	DLAB = 0	0x0000_0000
DLL	0xB800_0000	R/W	Divisor Latch Register (LS)	DLAB = 1	0x0000_0000
DLM	0xB800_0004	R/W	Divisor Latch Register (MS)	DLAB = 1	0x0000_0000
IIR	0xB800_0008	R	Interrupt Identification Register		0x8181_8181
FCR	0xB800_0008	W	FIFO Control Register		Undefined
LCR	0xB800_000C	R/W	Line Control Register		0x0000_0000
LSR	0xB800_0014	R	Line Status Register		0x6060_6060
TOR	0xB800_001C	R/W	Time Out Register		0x0000_0000
UART1 :					
RBR	0xB800_0100	R	Receive Buffer Register	DLAB = 0	Undefined
THR	0xB800_0100	W	Transmit Holding Register	DLAB = 0	Undefined
IER	0xB800_0104	R/W	Interrupt Enable Register	DLAB = 0	0x0000_0000
DLL	0xB800_0100	R/W	Divisor Latch Register (LS)	DLAB = 1	0x0000_0000
DLM	0xB800_0104	R/W	Divisor Latch Register (MS)	DLAB = 1	0x0000_0000
IIR	0xB800_0108	R	Interrupt Identification Register		0x8181_8181
FCR	0xB800_0108	W	FIFO Control Register		Undefined
LCR	0xB800_010C	R/W	Line Control Register		0x0000_0000
LSR	0xB800_0114	R	Line Status Register		0x6060_6060
TOR	0xB800_011C	R/W	Time Out Register		0x0000_0000

Receive Buffer Register (RBR)

Register	Offset	R/W	Description	Reset Value
RBR	0XB800_0x00	R	Receive Buffer Register (DLAB = 0)	Undefined

7	6	5	4	3	2	1	0
8-bit Received Data							

Bits	Descriptions	
[7:0]	8-bit Received Data	By reading this register, the UART will return an 8-bit data received from SIN pin (LSB first).

Transmit Holding Register (THR)

Register	offset	R/W	Description	Reset Value
THR	0XB800_0x00	W	Transmit Holding Register (DLAB = 0)	Undefined

7	6	5	4	3	2	1	0
8-bit Transmitted Data							

Bits	Descriptions	
[7:0]	8-bit Transmitted Data	By writing to this register, the UART will send out an 8-bit data through the SOUT pin (LSB first).

Interrupt Enable Register (IER)

Register	offset	R/W	Description	Reset Value
IER	0XB800_0x04	R/W	Interrupt Enable Register (DLAB = 0)	0x0000.0000

7	6	5	4	3	2	1	0
RESERVED				Reserved	RLSIE	THREIE	RDAIE

Bits	Descriptions	
		•
[2]	RLSIE	**Receive Line Status Interrupt (Irpt_RLS) Enable** • 0 = Mask off Irpt_RLS • 1 = Enable Irpt_RLS
[1]	THREIE	**Transmit Holding Register Empty Interrupt (Irpt_THRE) Enable** • 0 = Mask off Irpt_THRE • 1 = Enable Irpt_THRE
[0]	RDAIE	**R**eceive **D**ata **A**vailable **I**nterrupt **(Irpt_RDA) E**nable and Time-out Interrupt **(Irpt_TOUT)** Enable • 0 = Mask off Irpt_RDA and Irpt_TOUT • 1 = Enable Irpt_RDA and Irpt_TOUT

Divider Latch (Low Byte) Register (DLL)

Register	Offset	R/W	Description	Reset Value
DLL	0XB800_0x00	R/W	Divisor Latch Register (LS) (DLAB = 1)	0x0000_0000

7	6	5	4	3	2	1	0
Baud Rate Divider (Low Byte)							

Bits	Descriptions	
[7:0]	**Baud Rate Divisor (Low Byte)**	The low byte of the baud rate divider

Divisor Latch (High Byte) Register (DLM)

Register	Offset	R/W	Description	Reset Value
DLM	0XB800_0x04	R/W	Divisor Latch Register (MS) (DLAB = 1)	0x0000_0000

7	6	5	4	3	2	1	0
Baud Rate Divider (High Byte)							

Bits	Descriptions	
[7:0]	**Baud Rate Divisor (High Byte)**	The high byte of the baud rate divider

This 16-bit divider {DLM, DLL} is used to determine the baud rate as follows

Baud Rate = Crystal Clock / {16 * [Divisor + 2]}

Note: This definition is different from 16550

Interrupt Identification Register (IIR)

Register	Offset	R/W	Description	Reset Value
IIR	0XB800_0x08	R	Interrupt Identification Register	0x8181_8181

7	6	5	4	3	2	1	0
FMES	RFTLS		DMS	IID			NIP

Bits	Descriptions	
[7]	FMES	**FIFO Mode Enable Status** This bit indicates whether the FIFO mode is enabled or not. Since the FIFO mode is always enabled, this bit always shows the logical 1 when CPU is reading this register.
[6:5]	RFTLS	**Rx FIFO Threshold Level Status** These bits show the current setting of receiver FIFO threshold level (RTHO). The meaning of RTHO is defined in the following FCR description.
[4]	DMS	**DMA Mode Select** The DMA function is not implemented in this version. When reading IIR, the DMS is always returned 0.
[3:1]	IID	**Interrupt Identification** The IID together with NIP indicates the current interrupt request from UART.
[0]	NIP	**No Interrupt Pending** There is no pending interrupt.

Interrupt Control Functions

IIR [3:0]	Priority	Interrupt Type	Interrupt Source	Interrupt Reset control
---1	--	None	None	--
0110	Highest	Receiver Line Status (Irpt_RLS)	Overrun error, parity error, framing error, or break interrupt	Reading the LSR
0100	Second	Received Data Available (Irpt_RDA)	Receiver FIFO threshold level is reached	Receiver FIFO drops below the threshold level
1100	Second	Receiver FIFO Time-out (Irpt_TOUT)	Receiver FIFO is non-empty and no activities are occurred in the receiver FIFO during the TOR defined time duration	Reading the RBR
0010	Third	Transmitter Holing Register Empty (Irpt_THRE)	Transmitter holding register empty	Reading the IIR (if source of interrupt is Irpt_THRE) or writing into the THR
0000	Fourth	Reserved	Reseved	Reserved

Note: These definitions of bit 7, bit 6, bit 5, and bit 4 are different from the 16550.

FIFO Control Register (FCR)

Register	Offset	R/W	Description	Reset Value
FCR	0XB800_0x08	W	FIFO Control Register	Undefined

7	6	5	4	3	2	1	0
RFITL				DMS	TFR	RFR	FME

Bits	Descriptions	
[7:4]	RFITL	**Rx FIFO Interrupt (Irpt_RDA) Trigger Level** UART0: RFITL [7:4] / Trigger Level: 00xx → 01 bytes; 01xx → 04 bytes; 10xx → 08 bytes; 11xx → 14 bytes UART1: RFITL[7:4] / Trigger Level: 0000 → 01 bytes; 0001 → 04 bytes; 0010 → 08 bytes; 0011 → 14 bytes; 0100 → 30 bytes; 0101 → 46 bytes; others → 62 bytes
[3]	DMS	**DMA Mode Select** The DMA function is not implemented in this version.
[2]	TFR	**Tx FIFO Reset** Setting this bit will generate an OSC cycle reset pulse to reset Tx FIFO. The Tx FIFO becomes empty (Tx pointer is reset to 0) after such reset. This bit is returned to 0 automatically after the reset pulse is generated.
[1]	RFR	**Rx FIFO Reset** Setting this bit will generate an OSC cycle reset pulse to reset Rx FIFO. The Rx FIFO becomes empty (Rx pointer is reset to 0) after such reset. This bit is returned to 0 automatically after the reset pulse is generated.
[0]	FME	**FIFO Mode Enable** Because UART is always operating in the FIFO mode, writing this bit has no effect while reading always gets logical one. This bit must be 1 when other FCR bits are written to; otherwise, they will not be programmed.

Line Control Register (LCR)

Register	offset	R/W	Description	Reset Value
LCR	0XB800_0x0C	R/W	Line Control Register	0x0000_0000

7	6	5	4	3	2	1	0
DLAB	BCB	SPE	EPE	PBE	NSB	WLS	

Bits	Descriptions	
[7]	**DLAB**	**Divider Latch Access Bit** 0 = It is used to access RBR, THR or IER. 1 = It is used to access Divisor Latch Registers {DLL, DLM}.
[6]	**BCB**	**Break Control Bit** When this bit is set to logic 1, the serial data output (SOUT) is forced to the Spacing State (logic 0). This bit acts only on SOUT and has no effect on the transmitter logic.
[5]	**SPE**	**Stick Parity Enable** 0 = Disable stick parity 1 = Parity bit is transmitted and checked as a logic 1 if bit 4 is 0 (odd parity), or as a logic 0 if bit 4 is 1 (even parity). This bit has effect only when bit 3 (parity bit enable) is set.
[4]	**EPE**	**Even Parity Enable** 0 = Odd number of logic 1's are transmitted or checked in the data word and parity bits. 1 = Even number of logic 1's are transmitted or checked in the data word and parity bits. This bit has effect only when bit 3 (parity bit enable) is set.

[3]	**PBE**	**Parity Bit Enable** 0 = Parity bit is not generated (transmit data) or checked (receive data) during transfer. 1 = Parity bit is generated or checked between the "last data word bit" and "stop bit" of the serial data.
[2]	**NSB**	**Number of "STOP bit"** 0= One " STOP bit" is generated in the transmitted data 1= One and a half " STOP bit" is generated in the transmitted data when 5-bit word length is selected; Two " STOP bit" is generated when 6-, 7- and 8-bit word length is selected.
[1:0]	**WLS**	**Word Length Select** **WLS[1:0]** / **Character length** 00 / 5 bits 01 / 6 bits 10 / 7 bits 11 / 8 bits

Line Status Control Register (LSR)

Register	Offset	R/W	Description	Reset Value
LSR	0XB800_0x14	R	Line Status Register	0x6060_6060

7	6	5	4	3	2	1	0
ERR_Rx	TE	THRE	BII	FEI	PEI	OEI	RFDR

Bits	Descriptions	
[7]	**ERR_Rx**	**Rx FIFO Error** 0 = Rx FIFO works normally 1 = There is at least one parity error (PE), framing error (FE), or break indication (BI) in the FIFO. ERR_Rx is cleared when CPU reads the LSR and if there are no subsequent errors in the Rx FIFO.
[6]	**TE**	**Transmitter Empty** 0 = Either Transmitter Holding Register (**THR** - Tx FIFO) or Transmitter Shift Register (**TSR**) are not empty. 1 = Both THR and TSR are empty.
[5]	**THRE**	**Transmitter Holding Register Empty** 0 = THR is not empty. 1 = THR is empty. THRE is set when the last data word of Tx FIFO is transferred to Transmitter Shift Register (TSR). The CPU resets this bit when the THR (or Tx FIFO) is loaded. This bit also causes the UART to issue an interrupt (Irpt_THRE) to the CPU when IER [1]=1.
[4]	**BII**	**Break Interrupt Indicator** This bit is set to a logic 1 whenever the received data input is held in the "spacing state" (logic 0) for longer than a full word transmission time (that is, the total time of "start bit" + data bits + parity + stop bits) and is reset whenever the CPU reads the contents of the LSR.

[3]	**FEI**	**Framing Error Indicator** This bit is set to logic 1 whenever the received character does not have a valid "stop bit" (that is, the stop bit following the last data bit or parity bit is detected as a logic 0), and is reset whenever the CPU reads the contents of the LSR.
[2]	**PEI**	**Parity Error Indicator** This bit is set to logic 1 whenever the received character does not have a valid "parity bit", and is reset whenever the CPU reads the contents of the LSR.
[1]	**OEI**	**Overrun Error Indicator** An overrun error will occur only after the Rx FIFO is full and the next character has been completely received in the shift register. The character in the shift register is overwritten, but it is not transferred to the Rx FIFO. OE is indicated to the CPU as soon as it happens and is reset whenever the CPU reads the contents of the LSR.
[0]	**RFDR**	**Rx FIFO Data Ready** 0 = Rx FIFO is empty 1 = Rx FIFO contains at least 1 received data word.

LSR [4:2] (BII, FEI, PEI) are revealed to the CPU when its associated character is at the top of the Rx FIFO. These three error indicators are reset whenever the CPU reads the contents of the LSR.

LSR [4:1] (BII, FEI, PEI, OEI) are the error conditions that produce a "receiver line status interrupt" (Irpt_RLS) when IER [2]=1. Reading LSR clears Irpt_RLS. Writing LSR is a null operation (not suggested).

Time-Out Register (TOR)

Register	offset	R/W	Description	Reset Value
TOR	0XB800_0x1C	R/W	Time Out Register	0x0000_0000

7	6	5	4	3	2	1	0
TOIE	TOIC						

Bits	Descriptions	
[7]	TOIE	**Time Out Interrupt Enable** The feature of receiver time out interrupt is enabled only when TOR [7] = IER[0] = 1.
[6:0]	TOIC	**Time Out Interrupt Comparator** The time out counter resets and starts counting (the counting clock = baud rate) whenever the Rx FIFO receives a new data word. Once the content of time out counter (TOUT_CNT) is equal to that of time out interrupt comparator (TOIC), a receiver time out interrupt (Irpt_TOUT) is generated if TOR [7] = IER [0] = 1. A new incoming data word or Rx FIFO empty clears Irpt_TOUT.

7.12 TIMER Controller

7.12.1 General Timer Controller

The timer module includes five channels, TIMER0~TIMER4, they can easily be implemented as counting scheme. The timer can perform functions like frequency measurement, event counting, interval measurement, pulse generation, delay timing, and so on. The timer possesses features such as adjustable resolution, programmable counting period, and detailed information. The timer can generate an interrupt signal upon timeout, or provide the current value of count during operation.

The general TIMER Controller includes the following features

- Five channels with a 24-bit down counter and an interrupt request each
- Independent clock source for each channel
- Maximum uninterrupted time = (1 / 15 MHz) * (255) * (2^24 - 1), if TCLK = 15 MHz

7.12.2 Timer Control Registers Map

R: read only, **W**: write only, **R/W**: both read and write, **C**: Only value 0 can be written

Register	Address	R/W/C	Description	Reset Value
TMR_BA = 0xB800_1000				
TCSR0	0xB800_1000	R/W	Timer Control and Status Register 0	0x0000_0005
TCSR1	0xB800_1004	R/W	Timer Control and Status Register 1	0x0000_0005
TICR0	0xB800_1008	R/W	Timer Initial Control Register 0	0x0000_0000
TICR1	0xB800_100C	R/W	Timer Initial Control Register 1	0x0000_0000
TDR0	0xB800_1010	R	Timer Data Register 0	0x0000_0000
TDR1	0xB800_1014	R	Timer Data Register 1	0x0000_0000
TISR	0xB800_1018	R/C	Timer Interrupt Status Register	0x0000_0000
WTCR	0xB800_101C	R/W	Watchdog Timer Control Register	0x0000_0400
TCSR2	0xB800_1020	R/W	Timer Control and Status Register 2	0x0000_0005
TCSR3	0xB800_1024	R/W	Timer Control and Status Register 3	0x0000_0005
TICR2	0xB800_1028	R/W	Timer Initial Control Register 2	0x0000_0000
TICR3	0xB800_102C	R/W	Timer Initial Control Register 3	0x0000_0000
TDR2	0xB800_1030	R	Timer Data Register 2	0x0000_0000
TDR3	0xB800_1034	R	Timer Data Register 3	0x0000_0000
TCSR4	0xB800_1040	R/W	Timer Control and Status Register 4	0x0000_0005
TICR4	0xB800_1048	R/W	Timer Initial Control Register 4	0x0000_0000
TDR4	0xB800_1050	R	Timer Data Register 4	0x0000_0000

Timer Control and Status Register 0~4 (TCR0~TCR4)

Register	Address	R/W/C	Description	Reset Value
TCSR0	0xB800_1000	R/W	Timer Control and Status Register 0	0x0000_0005
TCSR1	0xB800_1004	R/W	Timer Control and Status Register 1	0x0000_0005
TCSR2	0xB800_1020	R/W	Timer Control and Status Register 2	0x0000_0005
TCSR3	0xB800_1024	R/W	Timer Control and Status Register 3	0x0000_0005
TCSR4	0xB800_1040	R/W	Timer Control and Status Register 4	0x0000_0005

31	30	29	28	27	26	25	24
RESERVED	CE	IE	MODE		CRST	CACT	RESERVED
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
PRESCALE							

Bits	Descriptions	
[30]	**CE**	**Counter Enable** 0 = Stops counting 1 = Starts counting
[29]	**IE**	**Interrupt Enable** 0 = Disables timer interrupt 1 = Enables timer interrupt. If timer interrupt is enabled, the timer asserts its interrupt signal when the associated counter decrements to zero.

<table>
<tr><td>[28:27]</td><td>MODE</td><td>Timer Operating Mode
<table>
<tr><th>MODE [28:27]</th><th>Timer Operating Mode</th></tr>
<tr><td>00</td><td>The timer is operating in the one-shot mode. The associated interrupt signal is generated once (if IE is enabled) and CE is automatically cleared then.</td></tr>
<tr><td>01</td><td>The timer is operating in the periodic mode. The associated interrupt signal is generated periodically (if IE is enabled).</td></tr>
<tr><td>10</td><td>The timer is operating in the toggle mode. The associated interrupt signal is changing back and forth (if IE is enabled) with 50% duty cycle.</td></tr>
<tr><td>11</td><td>Reserved for further use</td></tr>
</table></td></tr>
<tr><td>[26]</td><td>CRST</td><td>Counter Reset
Set this bit will reset the TIMER counter, and also force CEN to 0.
0 = No effect.
1 = Reset Timer’s pre-scale counter, internal 24-bit counter and CEN.</td></tr>
<tr><td>[25]</td><td>CACT</td><td>Timer is in Active
This bit indicates the counter status of timer.
0 = Timer is not active.
1 = Timer is in active.</td></tr>
<tr><td>[7:0]</td><td>PRESCALE</td><td>Clock Pre-scale Divide Count
Clock input is divided by PRESCALE + 1 before it is fed to the counter (here PRESCALE is considered as a decimal number). If PRESCALE = 0, then there is no scaling.</td></tr>
</table>

Timer Initial Count Register 0~4 (TICR0~TICR4)

Register	Address	R/W/C	Description	Reset Value
TICR0	0xB800_1008	R/W	Timer Initial Control Register 0	0x0000_00FF
TICR1	0xB800_100C	R/W	Timer Initial Control Register 1	0x0000_00FF
TICR2	0xB800_1028	R/W	Timer Initial Control Register 2	0x0000_00FF
TICR3	0xB800_102C	R/W	Timer Initial Control Register 3	0x0000_00FF
TICR4	0xB800_1048	R/W	Timer Initial Control Register 4	0x0000_00FF

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

Bits	Descriptions	
[23:0]	TIC	**Timer Initial Count** This is a 24-bit value representing the initial count. Timer will reload this value whenever the counter is decremented to zero. **NOTE:** (1) Never write 0x0 in TIC, or the core will run into unknown state. (2) No matter CEN is 0 or 1, whenever software write a new value into this register, Timer will restart counting using this new value and abort previous count.

Timer Data Register 0~4 (TDR0~TDR4)

Register	Address	R/W/C	Description	Reset Value
TDR0	0xB800_1010	R	Timer Data Register 0	0x0000_00FF
TDR1	0xB800_1014	R	Timer Data Register 1	0x0000_00FF
TDR2	0xB800_1030	R	Timer Data Register 2	0x0000_00FF
TDR3	0xB800_1034	R	Timer Data Register 3	0x0000_00FF
TDR4	0xB800_1050	R	Timer Data Register 4	0x0000_00FF

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

Bits	Descriptions	
[23:0]	**TDR**	**Timer Data Register** The current count is registered in this 24-bit value. **NOTE:** Software can read a correct current value on this register only when **CEN = 0**, or the value represents here could not be a correct one.

Timer Interrupt Status Register (TISR)

Register	Address	R/W/C	Description	Reset Value
TISR	0xB800_1018	R/C	Timer Interrupt Status Register	0x0000_0000

31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
RESERVED			TIF4	TIF3	TIF2	TIF1	TIF0

Bits	Descriptions	
[4]	TIF4	**Timer Interrupt Flag 4** 0 = It indicates that the timer 4 does not count down to zero yet. Software can reset this bit after the timer interrupt 4 had occurred. 1 = It indicates that the counter of timer 4 is decremented to zero; **NOTE**: This bit is read only, but can be cleared by writing 1 to this bit.
[3]	TIF3	**Timer Interrupt Flag 3** 0 = It indicates that the timer 3 does not count down to zero yet. Software can reset this bit after the timer interrupt 3 had occurred. 1 = It indicates that the counter of timer 3 is decremented to zero; **NOTE**: This bit is read only, but can be cleared by writing 1 to this bit.
[2]	TIF2	**Timer Interrupt Flag 2** 0 = It indicates that the timer 2 does not count down to zero yet. Software can reset this bit after the timer interrupt 2 had occurred. 1 = It indicates that the counter of timer 2 is decremented to zero; **NOTE**: This bit is read only, but can be cleared by writing 1 to this bit.

[1]	**TIF1**	**Timer Interrupt Flag 1** 0 = It indicates that the timer 1 does not count down to zero yet. Software can reset this bit after the timer interrupt 1 had occurred. 1 = It indicates that the counter of timer 1 is decremented to zero; **NOTE**: This bit is read only, but can be cleared by writing 1 to this bit.
[0]	**TIF0**	**Timer Interrupt Flag 0** 0 = It indicates that the timer 0 does not count down to zero yet. Software can reset this bit after the timer interrupt 0 had occurred. 1 = It indicates that the counter of timer 0 is decremented to zero; **NOTE**: This bit is read only, but can be cleared by writing 1 to this bit.

7.13 Advanced Interrupt Controller

An *interrupt* temporarily changes the sequence of program execution to react to a particular event such as power failure, watchdog timer timeout, transmit/receive request from Ethernet MAC Controller, and so on. The CPU processor provides two modes of interrupt, the **Fast Interrupt (FIQ)** mode for critical session and the *Interrupt* **(IRQ)** mode for general purpose. The IRQ request is occurred when the nIRQ input is asserted. Similarly, the FIQ request is occurred when the nFIQ input is asserted. The FIQ has privilege over the IRQ and can preempt an ongoing IRQ. It is possible to ignore the FIQ and the IRQ by setting the F and I bits in the **current program status register (CPSR)**.

The **Advanced Interrupt Controller (AIC)** is capable of processing the interrupt requests up to 32 different sources. Currently, 30 interrupt sources are defined. Each interrupt source is uniquely assigned to an *interrupt channel*. For example, the watchdog timer interrupt is assigned to channel 1. The AIC implements a proprietary eight-level priority scheme that categories the available 30 interrupt sources into eight priority levels. Interrupt sources within the priority level 0 is the highest priority and the priority level 7 is the lowest. In order to make this scheme work properly, a certain priority level must be specified to each interrupt source during power-on initialization; otherwise, the system shall behave unexpectedly. Within each priority level, interrupt source that is positioned in a lower channel has a higher priority. Interrupt source that is active, enabled, and positioned in the lowest channel with priority level 0 is promoted to the FIQ. Interrupt sources within the priority levels other than 0 are routed to the IRQ. The IRQ can be preempted by the occurrence of the FIQ. Interrupt nesting is performed automatically by the AIC.

Though interrupt sources originated from the chip itself are intrinsically high-level sensitive, the AIC can be configured as either low-level sensitive, high-level sensitive, negative-edge triggered, or positive-edge triggered to each interrupt source.

The advanced interrupt controller includes the following features:

- External interrupts can be programmed as either edge-triggered or level-sensitive
- External interrupts can be programmed as either low-active or high-active
- Flags to reflect the status of each interrupt source
- Individual mask for each interrupt source
- Proprietary 8-level interrupt scheme to employ the priority scheme.
- Priority methodology is adopted to allow for interrupt daisy-chaining
- Automatically masking out the lower priority interrupt during interrupt nesting
- Automatically clearing the interrupt flag when the external interrupt source is programmed to be edge-triggered

7.13.1 Interrupt Sources

Priority	Name	Mode	Source
1 (Highest)	WDT_INT	Positive Level	Watch Dog Timer Interrupt
2	nIRQ_Group0	Positive Level	External Interrupt Group 0
3	nIRQ_Group1	Positive Level	External Interrupt Group 1
4	Reserved	Reserved	Reserved
5	Reserved	Reserved	Reserved
6	RTC_INT	Positive Level	RTC Interrupt
7	UART_INT0	Positive Level	UART Interrupt0
8	UART_INT1	Positive Level	UART Interrupt1
9	Reserved	Reserved	Reserved
10	Reserved	Reserved	Reserved
11	Reserved	Reserved	Reserved
12	T_INT0	Positive Level	Timer Interrupt 0
13	T_INT1	Positive Level	Timer Interrupt 1
14	T_INT_Group	Positive Level	Timer Interrupt Group
15	USBH_INT_Group	Positive Level	USB Host Interrupt Group
16	EMCTx_INT	Positive Level	EMC Tx Interrupt
17	EMCRx_INT	Positive Level	EMC Rx Interrupt
18	Reserved	Reserved	Reserved
19	DMAC_INT	Positive Level	DMAC Interrupt
20	FMI_INT	Positive Level	FMI Interrupt
21	USBD_INT	Positive Level	USB Device Interrupt
22	Reserved	Reserved	Reserved
23	Reserved	Reserved	Reserved
24	Reserved	Reserved	Reserved
25	SC_INT_Group	Positive Level	Smart Card Interrupt Group
26	I2C_INT_Group	Positive Level	I2C Interrupt Group
27	USI_INT	Positive Level	USI Interrupt
28	Reserved	Reserved	Reserved
29	Reserved	Reserved	Reserved
30	Reserved	Reserved	Reserved
31	Reserved	Reserved	Reserved

Interrupt Group	Interrupt Sources
External Interrupt Group 0	External Pins : nIRQ[0]
Timer Interrupt Group	TIMER2, TIMER3, and TIMER4
USB Host Interrupt Group	OHCI and EHCI USB Host Controller
I2C interrupt Group	I2C Line 0 and I2C Line 1

7.13.2 AIC Registers Map

Register	Address	R/W	Description	Reset Value
AIC_BA = 0xB800_2000				
AIC_SCR1	0xB800_2004	R/W	Source Control Register 1	0x0000_0047
AIC_SCR2	0xB800_2008	R/W	Source Control Register 2	0x0000_0047
AIC_SCR3	0xB800_200C	R/W	Source Control Register 3	0x0000_0047
AIC_SCR4	0xB800_2010	R/W	Source Control Register 4	0x0000_0047
AIC_SCR5	0xB800_2014	R/W	Source Control Register 5	0x0000_0047
AIC_SCR6	0xB800_2018	R/W	Source Control Register 6	0x0000_0047
AIC_SCR7	0xB800_201C	R/W	Source Control Register 7	0x0000_0047
AIC_SCR8	0xB800_2020	R/W	Source Control Register 8	0x0000_0047
AIC_SCR9	0xB800_2024	R/W	Source Control Register 9	0x0000_0047
AIC_SCR10	0xB800_2028	R/W	Source Control Register 10	0x0000_0047
AIC_SCR11	0xB800_202C	R/W	Source Control Register 11	0x0000_0047
AIC_SCR12	0xB800_2030	R/W	Source Control Register 12	0x0000_0047
AIC_SCR13	0xB800_2034	R/W	Source Control Register 13	0x0000_0047
AIC_SCR14	0xB800_2038	R/W	Source Control Register 14	0x0000_0047
AIC_SCR15	0xB800_203C	R/W	Source Control Register 15	0x0000_0047
AIC_SCR16	0xB800_2040	R/W	Source Control Register 16	0x0000_0047
AIC_SCR17	0xB800_2044	R/W	Source Control Register 17	0x0000_0047
AIC_SCR18	0xB800_2048	R/W	Source Control Register 18	0x0000_0047
AIC_SCR19	0xB800_204C	R/W	Source Control Register 19	0x0000_0047
AIC_SCR20	0xB800_2050	R/W	Source Control Register 20	0x0000_0047
AIC_SCR21	0xB800_2054	R/W	Source Control Register 21	0x0000_0047
AIC_SCR22	0xB800_2058	R/W	Source Control Register 22	0x0000_0047
AIC_SCR23	0xB800_205C	R/W	Source Control Register 23	0x0000_0047
AIC_SCR24	0xB800_2060	R/W	Reserved	0x0000_0047

AIC_SCR25	0xB800_2064	R/W	Source Control Register 25	0x0000_0047
AIC_SCR26	0xB800_2068	R/W	Source Control Register 26	0x0000_0047
AIC_SCR27	0xB800_206C	R/W	Source Control Register 27	0x0000_0047
AIC_SCR28	0xB800_2070	R/W	Source Control Register 28	0x0000_0047
AIC_SCR29	0xB800_2074	R/W	Source Control Register 29	0x0000_0047
AIC_SCR30	0xB800_2078	R/W	Source Control Register 30	0x0000_0047
AIC_SCR31	0xB800_207C	R/W	Source Control Register 31	0x0000_0047
AIC_IRQSC	0xB800_2080	R/W	External Interrupt Control Register	0x0000_0000
AIC_GEN	0xB800_2084	R/W	Interrupt Group Enable Control Register	0x0000_0000
AIC_GASR	0xB800_2088	R	Interrupt Group Active Status Register	0x0000_0000
AIC_GSCR	0xB800_208C	W/R	Interrupt Group Status Clear Register	0x0000_0000
AIC_IRSR	0xB800_2100	R	Interrupt Raw Status Register	0x0000_0000
AIC_IASR	0xB800_2104	R	Interrupt Active Status Register	0x0000_0000
AIC_ISR	0xB800_2108	R	Interrupt Status Register	0x0000_0000
AIC_IPER	0xB800_210C	R	Interrupt Priority Encoding Register	0x0000_0000
AIC_ISNR	0xB800_2110	R	Interrupt Source Number Register	0x0000_0000
AIC_IMR	0xB800_2114	R	Interrupt Mask Register	0x0000_0000
AIC_OISR	0xB800_2118	R	Output Interrupt Status Register	0x0000_0000
AIC_MECR	0xB800_2120	W	Mask Enable Command Register	Undefined
AIC_MDCR	0xB800_2124	W	Mask Disable Command Register	Undefined
AIC_EOSCR	0xB800_2130	W	End of Service Command Register	Undefined

AIC Source Control Registers (AIC_SCR1 ~ AIC_SCR31)

Register	Address	R/W	Description	Reset Value
AIC_SCR1	0xB800_2004	R/W	Source Control Register 1	0x0000_0047
AIC_SCR2	0xB800_2008	R/W	Source Control Register 2	0x0000_0047
...	...	...	...	...
AIC_SCR28	0xB800_2070	R/W	Source Control Register 28	0x0000_0047
AIC_SCR29	0xB800_2074	R/W	Source Control Register 29	0x0000_0047
AIC_SCR30	0xB800_2078	R/W	Source Control Register 30	0x0000_0047
AIC_SCR31	0xB800_207C	R/W	Source Control Register 31	0x0000_0047

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

Bits	Descriptions	
[7:6]	SRCTYPE	**Interrupt Source Type** Whether an interrupt source is considered active or not by the AIC is subject to the settings of this field. Interrupt sources should be configured as level sensitive during normal operation unless in the testing situation.
[2:0]	PRIORITY	**Priority Level** Every interrupt source must be assigned a priority level during initiation.

SRCTYPE [7:6]		Interrupt Source Type
0	0	Low-level Sensitive
0	1	High-level Sensitive
1	0	Negative-edge Triggered
1	1	Positive-edge Triggered

		Among them, priority level 0 has the highest priority and priority level 7 the lowest. Interrupt sources with priority level 0 are promoted to FIQ. Interrupt sources with priority level other than 0 belong to IRQ. For interrupt sources of the same priority level, which located in the lower channel number has higher priority.

External Interrupt Control Register (AIC_IRQSC)

Register	Address	R/W	Description	Reset Value
AIC_IRQSC	0xB800_2080	R/W	External Interrupt Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
nIRQ7		nIRQ6		nIRQ5		nIRQ4	
7	6	5	4	3	2	1	0
nIRQ3		nIRQ2		nIRQ1		nIRQ0	

Bits	Descriptions	
[15:0]	nIRQ*x*	**External Interrupt Source Type** nIRQx: 0 0 — Low-level Sensitive 0 1 — High-level Sensitive 1 0 — Negative-edge Triggered 1 1 — Positive-edge Triggered

Interrupt Group Enable Control Register (AIC_GEN)

Register	Address	R/W	Description	Reset Value
AIC_GEN	0xB800_2084	R/W	Interrupt Group Enable Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved				I2C		Reserved	
23	22	21	20	19	18	17	16
Reserved				Reserved	TIMER		
15	14	13	12	11	10	9	8
Reserved						USBH	
7	6	5	4	3	2	1	0
Reserved							nIRQ[3:0]

Bits	Descriptions	
[27:26]	I2C	**I2C Controller Interrupt Group** Bit[27] is for I2C Line 1, Bit[26] is for Line 0 1: Interrupt Enable for each bit 0: Interrupt Disable for each bit
[18:16]	TIMER	**TIMER Controller Interrupt Group** Bit[18] is for TIMER4, Bit[17] is for TIMER3, and Bit[16] is for TIME2 1: Interrupt Enable for each bit 0: Interrupt Disable for each bit
[9:8]	USBH	**USB Host Controller Interrupt Group** Bit[9] is for OHCI Host Controller, Bit[9] is for EHCI Host Controller 1: Interrupt Enable for each bit 0: Interrupt Disable for each bit
[0]	nIRQ[0]	**External Interrupt Group 0** 1: Interrupt Enable for each bit 0: Interrupt Disable for each bit

Interrupt Group Active Status Register (AIC_GASR)

Register	Address	R/W	Description	Reset Value
AIC_GASR	0xB800_2088	R	Interrupt Group Active Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved				I2C		Reserved	
23	22	21	20	19	18	17	16
Reserved					TIMER		
15	14	13	12	11	10	9	8
Reserved						USBH	
7	6	5	4	3	2	1	0
Reserved							nIRQ[0]

Bits	Descriptions	
[27:26]	I2C	**I2C Controller Interrupt Group** Bit[27] is for I2C Line 1, Bit[26] is for Line 0
[18:16]	TIMER	**TIMER Controller Interrupt Group** Bit[18] is for TIMER4, Bit[17] is for TIMER3, and Bit[16] is for TIME2
[9:8]	USBH	**USB Host Controller Interrupt Group** Bit[9] is for OHCI Host Controller, Bit[9] is for EHCI Host Controller
[0]	nIRQ[0]	**External Interrupt Group 0**

Interrupt Group Status Clear Register (AIC_GSCR)

Register	Address	R/W	Description	Reset Value
AIC_GSCR	0xB800_208C	R/W	Interrupt Group Status Clear Register	0x0000_0000

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

Bits	Descriptions	
[0]	nIRQ[0]	**External Interrupt Group 0** Write 1: Clear the relative status bit, and this bit is auto clear to 0 Write 0: No action

AIC Interrupt Raw Status Register (AIC_IRSR)

Register	Address	R/W	Description	Reset Value
AIC_IRSR	0xB800_2100	R	Interrupt Raw Status Register	0x0000_0000

31	30	29	28	27	26	25	24
IRS31	IRS30	IRS29	IRS28	IRS27	IRS26	IRS25	IRS24
23	22	21	20	19	18	17	16
IRS23	IRS22	IRS21	IRS20	IRS19	IRS18	IRS17	IRS16
15	14	13	12	11	10	9	8
IRS15	IRS14	IRS13	IRS12	IRS11	IRS10	IRS9	IRS8
7	6	5	4	3	2	1	0
IRS7	IRS6	IRS5	IRS4	IRS3	IRS2	IRS1	RESERVED

Bits	Descriptions	
[31:1]	**IRS*x***	**Interrupt Status** Indicate the intrinsic status of the corresponding interrupt source 0 = Interrupt channel is in the voltage level 0 1 = Interrupt channel is in the voltage level 1

This register records the intrinsic state within each interrupt channel.

:

AIC Interrupt Active Status Register (AIC_IASR)

This register indicates the status of each interrupt channel in consideration of the interrupt source type as defined in the corresponding Source Control Register, but regardless of its mask setting.

Register	Address	R/W	Description	Reset Value
AIC_IASR	0xB800_2104	R	Interrupt Active Status Register	0x0000_0000

31	30	29	28	27	26	25	24
IAS31	IAS30	IAS29	IAS28	IAS27	IAS26	IAS25	IAS24
23	22	21	20	19	18	17	16
IAS23	IAS22	IAS21	IAS20	IAS19	IAS18	IAS17	IAS16
15	14	13	12	11	10	9	8
IAS15	IAS14	IAS13	IAS12	IAS11	IAS10	IAS9	IAS8
7	6	5	4	3	2	1	0
IAS7	IAS6	IAS5	IAS4	IAS3	IAS2	IAS1	RESERVED

Bits	Descriptions	
[31:1]	IAS*x*	**Interrupt Active Status** Indicate the status of the corresponding interrupt source 0 = Corresponding interrupt channel is inactive 1 = Corresponding interrupt channel is active

:

AIC Interrupt Status Register (AIC_ISR)

This register identifies those interrupt channels whose are both active and enabled.

Register	Address	R/W	Description	Reset Value
AIC_ISR	0xB800_2108	R	Interrupt Status Register	0x0000_0000

31	30	29	28	27	26	25	24
IS31	IS30	IS29	IS28	IS27	IS26	IS25	IS24
23	22	21	20	19	18	17	16
IS23	IS22	IS21	IS20	IS19	IS18	IS17	IS16
15	14	13	12	11	10	9	8
IS15	IS14	IS13	IS12	IS11	IS10	IS9	IS8
7	6	5	4	3	2	1	0
IS7	IS6	IS5	IS4	IS3	IS2	IS1	RESERVED

Bits	Descriptions	
[31:1]	**IS*x***	**Interrupt Status** Indicates the status of corresponding interrupt channel 0 = Two possibilities: (1) The corresponding interrupt channel is inactive no matter whether it is enabled or disabled; (2) It is active but not enabled 1 = Corresponding interrupt channel is both active and enabled (can assert an interrupt)

AIC IRQ Priority Encoding Register (AIC_IPER)

When the AIC generates the interrupt, **VECTOR** represents the interrupt channel number that is active, enabled, and has the highest priority. If the representing interrupt channel possesses a priority level 0, then the interrupt asserted is FIQ; otherwise, it is IRQ. The value of **VECTOR** is copied to the register AIC_ISNR thereafter by the AIC. This register was restored a value 0 after it was read by the interrupt handler. This register can help indexing into a branch table to quickly jump to the corresponding interrupt service routine.

Register	Address	R/W	Description	Reset Value
AIC_IPER	0xB800_210C	R	Interrupt Priority Encoding Register	0x0000_0000

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

Bits	Descriptions	
[6:2]	**VECTOR**	**Interrupt Vector** 0 = no interrupt occurs 1 ~ 31 = representing the interrupt channel that is active, enabled, and having the highest priority

AIC Interrupt Source Number Register (AIC_ISNR)

The purpose of this register is to record the interrupt channel number that is active, enabled, and has the highest priority.

Register	Address	R/W	Description	Reset Value
AIC_ISNR	0xB800_2110	R	Interrupt Source Number Register	0x0000_0000

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

Bits	Descriptions	
[4:0]	IRQID	**IRQ Identification** Stands for the interrupt channel number

AIC Interrupt Mask Register (AIC_IMR)

Register	Address	R/W	Description	Reset Value
AIC_IMR	0xB800_2114	R	Interrupt Mask Register	0x0000_0000

31	30	29	28	27	26	25	24
IM31	IM30	IM29	IM28	IM27	IM26	IM25	IM24
23	22	21	20	19	18	17	16
IM23	IM22	IM21	IM20	IM19	IM18	IM17	IM16
15	14	13	12	11	10	9	8
IM15	IM14	IM13	IM12	IM11	IM10	IM9	IM8
7	6	5	4	3	2	1	0
IM7	IM6	IM5	IM4	IM3	IM2	IM1	RESERVED

Bits	Descriptions	
[31:1]	**IM*x***	**Interrupt Mask** This bit determines whether the corresponding interrupt channel is enabled or disabled. Every interrupt channel can be active no matter whether it is enabled or disabled. If an interrupt channel is enabled, it does not definitely mean it is active. Every interrupt channel can be authorized by the AIC only when it is both active and enabled. 0 = Corresponding interrupt channel is disabled 1 = Corresponding interrupt channel is enabled

:

AIC Output Interrupt Status Register (AIC_OISR)

The AIC classifies the interrupt into FIQ and IRQ. This register indicates whether the asserted interrupt is FIQ or IRQ. If both IRQ and FIQ are equal to 0, it means there is no interrupt occurred.

Register	Address	R/W	Description	Reset Value
AIC_OISR	0xB800_2118	R	Output Interrupt Status Register	0x0000_0000

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

Bits	Descriptions	
[1]	IRQ	**Interrupt Request** 0 = nIRQ line is inactive. 1 = nIRQ line is active.
[0]	FIQ	**Fast Interrupt Request** 0 = nFIQ line is inactive. 1 = nFIQ line is active

AIC Mask Enable Command Register (AIC_MECR)

Register	Address	R/W	Description	Reset Value
AIC_MECR	0xB800_2120	W	Mask Enable Command Register	Undefined

31	30	29	28	27	26	25	24
MEC31	MEC30	MEC29	MEC28	MEC27	MEC26	MEC25	MEC24
23	22	21	20	19	18	17	16
MEC23	MEC22	MEC21	MEC20	MEC19	MEC18	MEC17	MEC16
15	14	13	12	11	10	9	8
MEC15	MEC14	MEC13	MEC12	MEC11	MEC10	MEC9	MEC8
7	6	5	4	3	2	1	0
MEC7	MEC6	MEC5	MEC4	MEC3	MEC2	MEC1	RESERVED

Bits	Descriptions	
[31:1]	**MEC*x***	**Mask Enable Command** 0 = No effect 1 = Enables the corresponding interrupt channel MEC24 has to set 0 for the reserved interrupt source.

:

AIC Mask Disable Command Register (AIC_MDCR)

Register	Address	R/W	Description	Reset Value
AIC_MDCR	0xB800_2124	W	Mask Disable Command Register	Undefined

31	30	29	28	27	26	25	24
MDC31	MDC30	MDC29	MDC28	MDC27	MDC26	MDC25	MDC24
23	22	21	20	19	18	17	16
MDC23	MDC22	MDC21	MDC20	MDC19	MDC18	MDC17	MDC16
15	14	13	12	11	10	9	8
MDC15	MDC14	MDC13	MDC12	MDC11	MDC10	MDC9	MDC8
7	6	5	4	3	2	1	0
MDC7	MDC6	MDC5	MDC4	MDC3	MDC2	MDC1	RESERVED

Bits	Descriptions	
[31:1]	**MDC*x***	**Mask Disable Command** 0 = No effect 1 = Disables the corresponding interrupt channel

AIC End of Service Command Register (AIC_EOSCR)

This register is used by the interrupt service routine to indicate that it is completely served. Thus, the interrupt handler can write any value to this register to indicate the end of its interrupt service.

Register	Address	R/W	Description	Reset Value
AIC_EOSCR	0xB800_2130	W	End of Service Command Register	Undefined

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

7.14 General-Purpose Input/Output (GPIO)

7.14.1 Overview

The General-Purpose Input/Output (**GPIO**) module possesses 38 pins, and serves as multiple function purposes. Each port can be easily configured by software to meet various system configurations and design requirements. Software must define which function of each pin is used before starting the main program. If a pin is not used for multiplexed functions, the pin can be configured as I/O ports.

These 38 IO pins are divided into 6 groups according to its peripheral interface definition.

- PortC: 11-pin input/output port
- PortD: 8-pin input/output port
- PortE: 4-pin input/output port
- PortF: 10-pin input/output port
- PortG: 4-pin input/output port
- PortH: 1-pin input/output port

7.14.2 GPIO Multiplexed Functions Table

GPIO Groups	Shared Interface
	NUC946ADN
GPIOC (11 pins)	GPIO
GPIOC[2]	
GPIOC[4]	
GPIOC[5]	
GPIOC[6]	
GPIOC[7]	
GPIOC[8]	
GPIOC[9]	
GPIOC[10]	
GPIOC[11]	
GPIOC[13]	
GPIOC[14]	
GPIOD (8 pins)	SD(SDIO) Interface
GPIOD[0]	SD_CMD
GPIOD[1]	SD_CLK
GPIOD[2]	SD_DAT0
GPIOD[3]	SD_DAT1
GPIOD[4]	SD_DAT2
GPIOD[5]	SD_DAT3
GPIOD[6]	SD_CD
GPIOD[8]	SD_nPWR
GPIOE (4 pins)	UART Interface
GPIOE[0]	TXD0
GPIOE[1]	RXD0
GPIOE[2]	TXD1(B)
GPIOE[3]	RXD1(B)
GPIOF (10 pins)	RMII Interface
GPIOF[0]	PHY_MDC
GPIOF [1]	PHY_MDIO
GPIOF [3:2]	PHY_TXD [1:0]
GPIOF [4]	PHY_TXEN
GPIOF [5]	PHY_REFCLK
GPIOF [7:6]	PHY_RXD [1:0]
GPIOF [8]	PHY_CRSDV
GPIOF [9]	PHY_RXERR
GPIOG (4 pins)	I2C/USI
GPIOG[0]	SCL0 / SFRM
GPIOG[1]	SDA0 / SSPTXD
GPIOG[2]	SCL1 / SCLK
GPIOG[3]	SDA1 / SSPRXD
GPIOH (1 pin)	nIRQ Interface
GPIOH[0]	nIRQ[0]

7.14.3 GPIO Control Registers Map

Register	Address	R/W	Description	Reset Value
GPIO_BA = 0xB800_3000				
GPIOC_DIR	0xB800_3004	R/W	GPIO portC direction control register	0x0000_0000
GPIOC_DATAOUT	0xB800_3008	R/W	GPIO portC data output register	0x0000_0000
GPIOC_DATAIN	0xB800_300C	R	GPIO portC data input register	Undefined
GPIOD_DIR	0xB800_3014	R/W	GPIO portD direction control register	0x0000_0000
GPIOD_DATAOUT	0xB800_3018	R/W	GPIO portD data output register	0x0000_0000
GPIOD_DATAIN	0xB800_301C	R	GPIO portD data input register	Undefined
GPIOE_DIR	0xB800_3024	R/W	GPIO portE direction control register	0x0000_0000
GPIOE_DATAOUT	0xB800_3028	R/W	GPIO portE data output register	0x0000_0000
GPIOE_DATAIN	0xB800_302C	R	GPIO portE data input register	0x0000_0000
GPIOF_DIR	0xB800_3034	R/W	GPIO portF direction control register	0x0000_0000
GPIOF_DATAOUT	0xB800_3038	R/W	GPIO portF data output register	0x0000_0000
GPIOF_DATAIN	0xB800_303C	R	GPIO portF data input register	Undefined
GPIOG_DIR	0xB800_3044	R/W	GPIO portG direction control register	0x0000_0000
GPIOG_DATAOUT	0xB800_3048	R/W	GPIO portG data output register	0x0000_0000
GPIOG_DATAIN	0xB800_304C	R	GPIO portG data input register	Undefined
GPIOH_DBNCE	0xB800_3050	R/W	GPIO portH input de-bounce control reg.	0x0000_0000
GPIOH_DIR	0xB800_3054	R/W	GPIO portH direction control register	0x0000_0000
GPIOH_DATAOUT	0xB800_3058	R/W	GPIO portH data output register	0x0000_0000
GPIOH_DATAIN	0xB800_305C	R	GPIO portH data input register	Undefined

GPIO PortC Direction Control Register (GPIOC_DIR)

Register	Address	R/W	Description	Reset Value
GPIOC_DIR	0xB800_3004	R/W	GPIO portC in/out direction control register	0x0000_0000

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

Bits	Descriptions	
[14..13] [11..4] [2]	**OUTEN**	**GPIO PortC Output Enable Control** Each GPIO pin can be enabled individually by setting the corresponding control bit. 0 = Input Mode 1 = Output Mode

GPIO PortC Data Output Register (GPIOC_DATAOUT)

Register	Address	R/W	Description	Reset Value
GPIOC_DATAOUT	0xB800_3008	R/W	GPIO portC data output register	0x0000_0000

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

Bits	Descriptions	
[14..13] [11..4] [2]	DATAOUT	**GPIO PortC Data Output Value** Writing data to this register will reflect the data value on the corresponding pin when it is configured as general output pin. And writing data to reserved bits is not effective.

GPIO PortC Data Input Register (GPIOC_DATAIN)

Register	Address	R/W	Description	Reset Value
GPIOC_DATAIN	0xB800_300C	R	GPIO portC data input register	0xxxxx_xxxx

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

Bits	Descriptions	
[14..13] [11..4] [2]	DATAIN	**GPIO PortC Data Input Value** The DATAIN indicates the status of each GPIO portC pin regardless of its operation mode. The reserved bits will be read as "0".

GPIO PortD Direction Control Register (GPIOD_DIR)

Register	Address	R/W	Description	Reset Value
GPIOD_DIR	0xB800_3014	R/W	GPIO portD in/out direction control register	0x0000_0000

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

Bits	Descriptions	
[8] [6:0]	OUTEN	**GPIO PortD Output Enable Control** Each GPIO pin can be enabled individually by setting the corresponding control bit. 0 = Input Mode 1 = Output Mode

GPIO PortD Data Output Register (GPIOD_DATAOUT)

Register	Address	R/W	Description	Reset Value
GPIOD_DATAOUT	0xB800_3018	R/W	GPIO portD data output register	0x0000_0000

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

Bits	Descriptions	
[8] [6:0]	DATAOUT	**GPIO PortD Data Output Value** Writing data to this register will reflect the data value on the corresponding pin when it is configured as general output pin. And writing data to reserved bits is not effective.

GPIO PortD Data Input Register (GPIOD_DATAIN)

Register	Address	R/W	Description	Reset Value
GPIOD_DATAIN	0xB800_301C	R	GPIO portD data input register	0xxxxx_xxxx

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

Bits	Descriptions	
[8] [6:0]	DATAIN	**GPIO PortD Data Input Value** The DATAIN indicates the status of each GPIO portD pin regardless of its operation mode. The reserved bits will be read as "0".

GPIO PortE Direction Control Register (GPIOE_DIR)

Register	Address	R/W	Description	Reset Value
GPIOE_DIR	0xB800_3024	R/W	GPIO portE in/out direction control register	0x0000_0000

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

Bits	Descriptions	
[3:0]	**OUTEN**	**GPIO PortE Output Enable Control** Each GPIO pin can be enabled individually by setting the corresponding control bit. 0 = Input Mode 1 = Output Mode

GPIO PortE Data Output Register (GPIOE_DATAOUT)

Register	Address	R/W	Description	Reset Value
GPIOE_DATAOUT	0xB800_3028	R/W	GPIO portE data output register	0x0000_0000

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

Bits	Descriptions	
[3:0]	**DATAOUT**	**GPIO PortE Data Output Value** Writing data to this register will reflect the data value on the corresponding pin when it is configured as general output pin. And writing data to reserved bits is not effective.

GPIO PortE Data Input Register (GPIOE_DATAIN)

Register	Address	R/W	Description	Reset Value
GPIOE_DATAIN	0xB800_302C	R	GPIO portE data input register	0xxxxx_xxxx

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

Bits	Descriptions	
[3:0]	**DATAIN**	**GPIO PortE Data Input Value** The DATAIN indicates the status of each GPIO portE pin regardless of its operation mode. The reserved bits will be read as "0".

GPIO PortF Direction Control Register (GPIOF_DIR)

Register	Address	R/W	Description	Reset Value
GPIOF_DIR	0xB800_3034	R/W	GPIO portF in/out direction control register	0x0000_0000

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

Bits	Descriptions	
[9:0]	**OUTEN**	**GPIO PortF Output Enable Control** Each GPIO pin can be enabled individually by setting the corresponding control bit. 0 = Input Mode 1 = Output Mode

GPIO PortF Data Output Register (GPIOF_DATAOUT)

Register	Address	R/W	Description	Reset Value
GPIOF_DATAOUT	0xB800_3038	R/W	GPIO portF data output register	0x0000_0000

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

Bits	Descriptions	
[9:0]	DATAOUT	**GPIO PortF Data Output Value** Writing data to this register will reflect the data value on the corresponding pin when it is configured as general output pin. And writing data to reserved bits is not effective.

GPIO PortF Data Input Register (GPIOF_DATAIN)

Register	Address	R/W	Description	Reset Value
GPIOF_DATAIN	0xB800_303C	R	GPIO portF data input register	0xxxxx_xxxx

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

Bits	Descriptions	
[9:0]	DATAIN	**GPIO PortF Data Input Value** The DATAIN indicates the status of each GPIO portF pin regardless of its operation mode. The reserved bits will be read as "0".

GPIO PortG Direction Control Register (GPIOG_DIR)

Register	Address	R/W	Description	Reset Value
GPIOG_DIR	0xB800_3044	R/W	GPIO portG in/out direction control register	0x0000_0000

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

Bits	Descriptions	
[3:0]	**OUTEN**	**GPIO PortG Output Enable Control** Each GPIO pin can be enabled individually by setting the corresponding control bit. 0 = Input Mode 1 = Output Mode

GPIO PortG Data Output Register (GPIOG_DATAOUT)

Register	Address	R/W	Description	Reset Value
GPIOG_DATAOUT	0xB800_3048	R/W	GPIO portG data output register	0x0000_0000

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

Bits	Descriptions	
[3:0]	**DATAOUT**	**GPIO PortG Data Output Value** Writing data to this register will reflect the data value on the corresponding pin when it is configured as general output pin. And writing data to reserved bits is not effective.

GPIO PortG Data Input Register (GPIOG_DATAIN)

Register	Address	R/W	Description	Reset Value
GPIOG_DATAIN	0xB800_304C	R	GPIO portG data input register	0xxxxx_xxxx

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

Bits	Descriptions	
[3:0]	**DATAIN**	**GPIO PortG Data Input Value** The DATAIN indicates the status of each GPIO portG pin regardless of its operation mode. The reserved bits will be read as "0".

GPIO PortH De-bounce Enable Control Register (GPIOH_DBNCE)

Register	Address	R/W	Description	Reset Value
GPIOH_DBNCE	0xB800_3050	R/W	GPIO PortH de-bounce control register	0xxxxx_xxxx

31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED					DBCLKSEL		
7	6	5	4	3	2	1	0
Reserved							DBEN0

Bits	Descriptions	
[10:8]	**DBCLKSEL**	**De-bounce Clock Selection** These 3 bits are used to select the clock rate for de-bouncer circuit. The relationship between the system clock HCLK and the de-bounce clock TCLK_BUN is as follows: TCLK_BUN = HCLK / $2^{DBCLKSEL}$
[0]	**DBEN0**	**De-bounce Circuit Enable for GPIOH0 (nIRQ0) Input** 1 = Enable De-bounce 0 = Disable De-bounce

GPIO PortH Direction Control Register (GPIOH_DIR)

Register	Address	R/W	Description	Reset Value
GPIOH_DIR	0xB800_3054	R/W	GPIO portH in/out direction control register	0x0000_0000

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

Bits	Descriptions	
[0]	**OUTEN**	**GPIO PortH Output Enable Control** Each GPIO pin can be enabled individually by setting the corresponding control bit. 0 = Input Mode 1 = Output Mode

GPIO PortH Data Output Register (GPIOH_DATAOUT)

Register	Address	R/W	Description	Reset Value
GPIOH_DATAOUT	0xB800_3058	R/W	GPIO portH data output register	0x0000_0000

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

Bits	Descriptions	
[0]	**DATAOUT**	**GPIO PortH Data Output Value** Writing data to this register will reflect the data value on the corresponding pin when it is configured as general output pin. And writing data to reserved bits is not effective.

GPIO PortH Data Input Register (GPIOH_DATAIN)

Register	Address	R/W	Description	Reset Value
GPIOH_DATAIN	0xB800_305C	R	GPIO portH data input register	0xxxxx_xxxx

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

Bits	Descriptions	
[0]	**DATAIN**	**GPIO PortH Data Input Value** The DATAIN indicates the status of each GPIO portH pin regardless of its operation mode. The reserved bits will be read as "0".

7.15 I²C Synchronous Serial Interface Controller

I²C is a two-wire, bi-directional serial bus that provides a simple and efficient method of data exchange between devices. The I²C standard is a true multi-master bus including collision detection and arbitration that prevents data corruption if two or more masters attempt to control the bus simultaneously.
Serial, 8-bit oriented bi-directional data transfers can be up to 100 Kb/s in Standard-mode, 400 Kb/s in the Fast-mode, or 3.4 Mb/s in the High-speed mode. Only 100kbps and 400kbps modes are supported directly in this chip.
Data transfer is synchronized to SCL signal between a Master and a Slave with byte-by-byte basis. Each data byte is 8 bits long. There is one SCL clock pulse for each data bit with the MSB being transmitted first. An acknowledge bit follows each transferred byte. Each bit is sampled during the high period of SCL; therefore, the SDA line may be changed only during the low period of SCL and must be held stable during the high period of SCL. A transition on the SDA line while SCL is high is interpreted as a command (START or STOP).

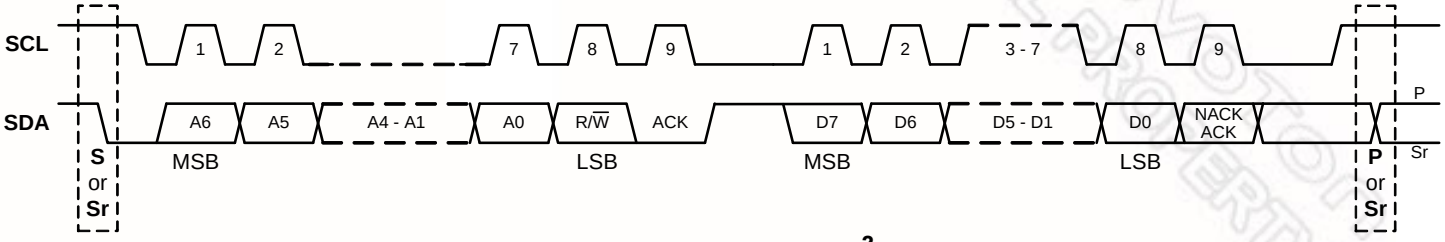
The I²C Master Core includes the following features:

- Compatible with I²C standard, support master mode
- Multi Master Operation.
- Clock stretching and wait state generation.
- Provide multi-byte transmit operation, up to 4 bytes can be transmitted in a single transfer
- Software programmable acknowledge bit.
- Arbitration lost interrupt, with automatic transfer cancellation.
- Start/Stop/Repeated Start/Acknowledge generation.
- Start/Stop/Repeated Start detection.
- Bus busy detection.
- Supports 7 bit addressing mode.
- Fully static synchronous design with one clock domain.
- Software mode I²C.

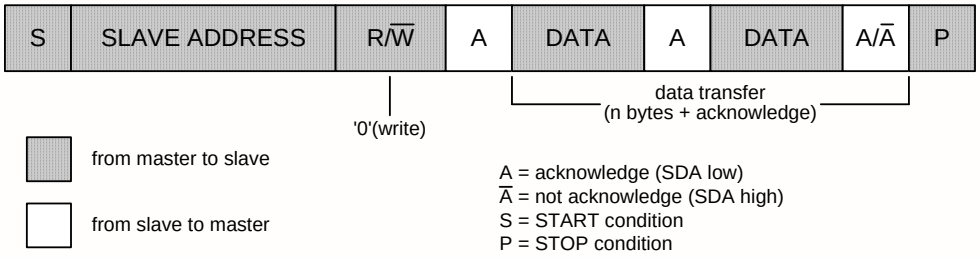
7.15.1 I²C Protocol

Normally, a standard communication consists of four parts:

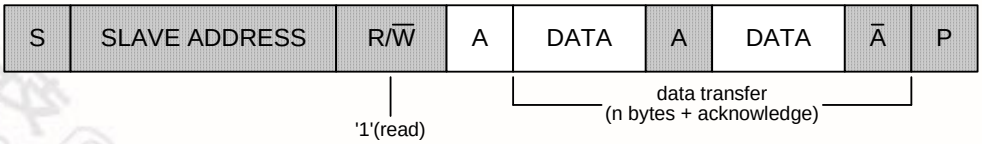
1) START or Repeated START signal generation
2) Slave address transfer
3) Data transfer
4) STOP signal generation



Data transfer on the I²C-bus



A master-transmitter addressing a slave receiver with a 7-bit address
The transfer direction is not changed



A master reads a slave immediately after the first byte (address)

START or Repeated START signal

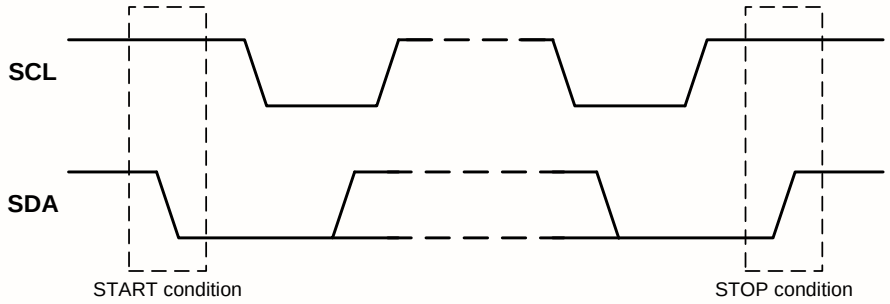
When the bus is free/idle, meaning no master device is engaging the bus (both SCL and SDA lines are high), a master can initiate a transfer by sending a START signal. A START signal, usually referred to as the **S-bit**, is defined as a **HIGH to LOW** transition on the SDA line while SCL is **HIGH**. The START signal denotes the beginning of a new data transfer.

A Repeated START (Sr) is a START signal without first generating a STOP signal. The master uses this method to communicate with another slave or the same slave in a different transfer direction (e.g. from writing to a device to reading from a device) without releasing the bus.

The I^2C core generates a START signal when the START bit in the Command Register (CMDR) is set and the READ or WRITE bits are also set. Depending on the current status of the SCL line, a START or Repeated START is generated.

STOP signal

The master can terminate the communication by generating a STOP signal. A STOP signal, usually referred to as the **P-bit**, is defined as a **LOW to HIGH** transition on the SDA line while SCL is **HIGH**.

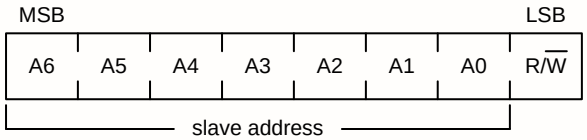


START and STOP conditions

Slave Address Transfer

The first byte of data transferred by the master immediately after the START signal is the slave address. This is a 7-bits calling address followed by a RW bit. The RW bit signals the slave the data transfer direction. No two slaves in the system can have the same address. Only the slave with an address that matches the one transmitted by the master will respond by returning an acknowledge bit by pulling the SDA low at the 9th SCL clock cycle.

The core treats a Slave Address Transfer as any other write action. Store the slave device's address in the Transmit Register (TxR) and set the WRITE bit. The core will then transfer the slave address on the bus.



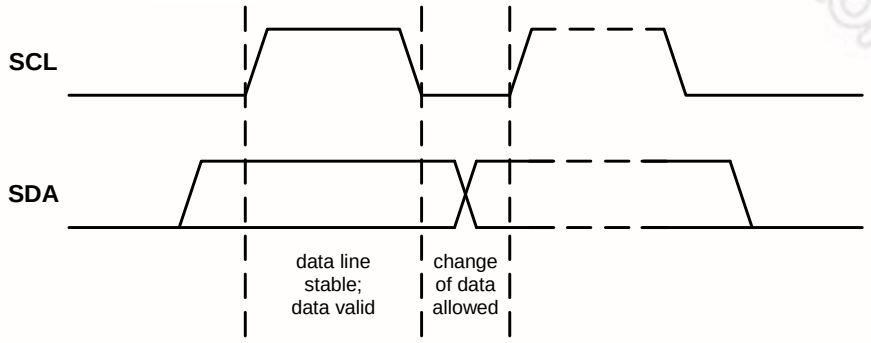
The first byte after the START procedure

Data Transfer

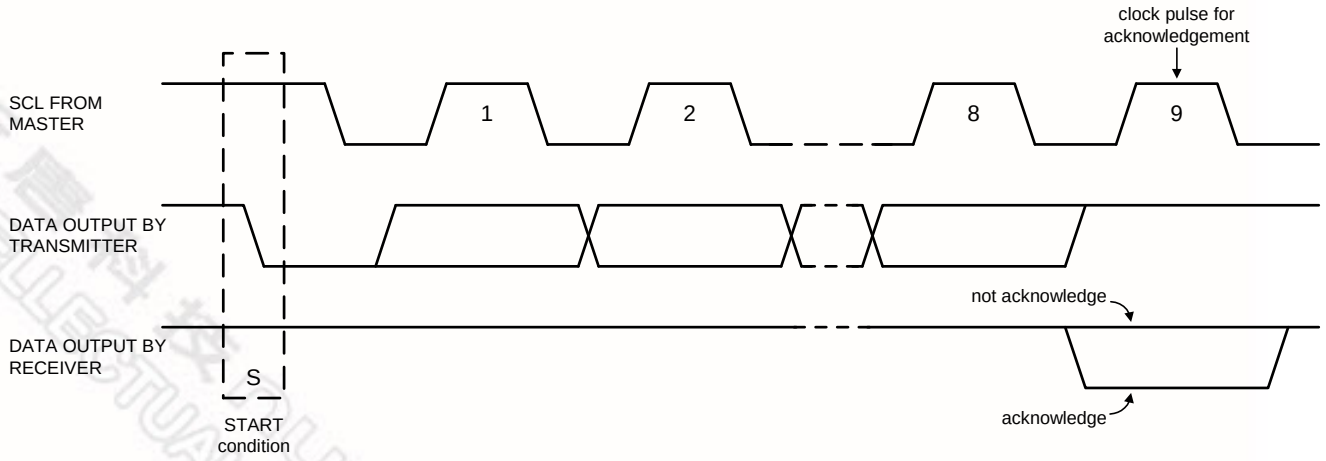
Once successful slave addressing has been achieved, the data transfer can proceed on a byte-by-byte basis in the direction specified by the RW bit sent by the master. Each transferred byte is followed by an acknowledge bit on the 9th SCL clock cycle. If the slave signals a **Not Acknowledge (NACK)**, the master can generate a STOP signal to abort the data transfer or generate a Repeated START signal and start a new transfer cycle.

If the master, as the receiving device, does **Not Acknowledge (NACK)** the slave, the slave releases the SDA line for the master to generate a STOP or Repeated START signal.

To write data to a slave, store the data to be transmitted in the Transmit Register (TxR) and set the WRITE bit. To read data from a slave, set the READ bit. During a transfer the core set the I2C_TIP flag, indicating that a **Transfer is in Progress**. When the transfer is done the I2C_TIP flag is cleared, and the IF flag set. And if IE is enabled, then an interrupt generated. The Receive Register (RxR) contains valid data after the IF flag has been set. The software may issue a new write or read command when the I2C_TIP flag is cleared.



Bit transfer on the I²C-bus



Acknowledge on the I²C-bus

7.15.2 I2C Serial Interface Control Registers Map

R: read only, **W**: write only, **R/W**: both read and write

Register	Offset	R/W/C	Description	Reset Value
I2C Port0 : I2C_BA = 0xB800_6000 **I2C Port1 : I2C_BA = 0xB800_6100**				
CSR	0xB800_6x00	R/W	Control and Status Register	0x0000_0000
DIVIDER	0XB800_6x04	R/W	Clock Pre-scale Register	0x0000_0000
CMDR	0XB800_6x08	R/W	Command Register	0x0000_0000
SWR	0XB800_6x0C	R/W	Software Mode Control Register	0x0000_003F
RxR	0XB800_6x10	R	Data Receive Register	0x0000_0000
TxR	0XB800_6x14	R/W	Data Transmit Register	0x0000_0000

NOTE: The reset value of SWR is 0x3F only when SCR, SDR and SER are connected to pull high resistor.

Control and Status Register (CSR)

Register	Offset	R/W/C	Description	Reset Value
CSR	0XB800_6x00	R/W	Control and Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved				I2C_RxACK	I2C_BUSY	I2C_AL	I2C_TIP
7	6	5	4	3	2	1	0
Reserved		Tx_NUM		Reserved	IF	IE	I2C_EN

Bits	Descriptions	
[11]	I2C_RxACK	**Received Acknowledge From Slave (Read only)** This flag represents acknowledge from the addressed slave. • 0 = Acknowledge received (ACK). • 1 = Not acknowledge received (NACK).
[10]	I2C_BUSY	**I²C Bus Busy (Read only)** • 0 = After STOP signal detected. • 1 = After START signal detected.
[9]	I2C_AL	**Arbitration Lost (Read only)** This bit is set when the I²C core lost arbitration. Arbitration is lost when: • A STOP signal is detected, but no requested. • The master drives SDA high, but SDA is low.
[8]	I2C_TIP	**Transfer In Progress (Read only)** • 0 = Transfer complete. • 1 = Transferring data. **NOTE:** When a transfer is in progress, you will not allow writing to any register of the I²C master core except SWR.

[5:4]	**Tx_NUM**	**Transmit Byte Counts** These two bits represent how many bytes are remained to transmit. When a byte has been transmitted, the Tx_NUM will decrease 1 until all bytes are transmitted (Tx_NUM = 0x0) or NACK received from slave. Then the interrupt signal will assert if IE was set. 0x0 = Only one byte is left for transmission. 0x1 = Two bytes are left to for transmission. 0x2 = Three bytes are left for transmission. 0x3 = Four bytes are left for transmission.
[2]	**IF**	**Interrupt Flag** The Interrupt Flag is set when: • Transfer has been completed. • Transfer has not been completed, but slave responded NACK (in multi-byte transmit mode). • Arbitration is lost. **NOTE:** This bit is read only, but can be cleared by writing 1 to this bit.
[1]	**IE**	**Interrupt Enable** 0 = Disable I^2C Interrupt. 1 = Enable I^2C Interrupt.
[0]	**I2C_EN**	**I^2C Core Enable** 0 = Disable I^2C core, serial bus outputs are controlled by SDW/SCW. 1 = Enable I^2C core, serial bus outputs are controlled by I^2C core.

Pre-scale Register (DIVIDER)

Register	Offset	R/W/C	Description	Reset Value
DIVIDER	0XB800_6x04	R/W	Clock Pre-scale Register	0x0000_0000

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

Bits	Descriptions	
[15:0]	DIVIDER	**Clock Pre-scale Register** It is used to pre-scale the SCL clock line. Due to the structure of the I^2C interface, the core uses a 5*SCL clock internally. The pre-scale register must be programmed to this 5*SCL frequency (minus 1). Change the value of the pre-scale register only when the "I2C_EN" bit is cleared. Example: pclk = 32MHz, desired SCL = 100KHz $prescale = \frac{32\ MHz}{5 * 100\ KHz} - 1 = 63\,(dec) = 3F\,(hex)$

Command Register (CMDR)

Register	Offset	R/W/C	Description	Reset Value
CMDR	0XB800_6x08	R/W	Command Register	0x0000_0000

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

NOTE: Software can write this register only when I2C_EN = 1.

Bits	Descriptions	
[4]	**START**	**Generate Start Condition** Generate (repeated) start condition on I^2C bus.
[3]	**STOP**	**Generate Stop Condition** Generate stop condition on I^2C bus.
[2]	**READ**	**Read Data From Slave** Retrieve data from slave.
[1]	**WRITE**	**Write Data To Slave** Transmit data to slave.
[0]	**ACK**	**Send Acknowledge To Slave** When I^2C behaves as a receiver, sent ACK (ACK = '0') or NACK (ACK = '1') to slave.

NOTE: The START, STOP, READ and WRITE bits are cleared automatically while transfer finished. READ and WRITE cannot be set concurrently.

Software Mode Register (SWR)

Register	Offset	R/W/C	Description	Reset Value
SWR	0XB800_6x0C	R/W	Software Mode Control Register	0x0000_003F

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved		SER	SDR	SCR	SEW	SDW	SCW

NOTE: This register is used as software mode of I^2C. Software can read/write this register no matter I2C_EN is 0 or 1. But SCL and SDA are controlled by software only when I2C_EN = 0.

Bits	Descriptions	
[5]	**SER**	**Serial Interface SDO Status (Read only)** 0 = SDO is Low. 1 = SDO is High.
[4]	**SDR**	**Serial Interface SDA Status (Read only)** 0 = SDA is Low. 1 = SDA is High.
[3]	**SCR**	**Serial Interface SCK Status (Read only)** 0 = SCL is Low. 1 = SCL is High.
[2]	**SEW**	**Serial Interface SDO Output Control** 0 = SDO pin is driven Low. 1 = SDO pin is tri-state.
[1]	**SDW**	**Serial Interface SDA Output Control** 0 = SDA pin is driven Low. 1 = SDA pin is tri-state.
[0]	**SCW**	**Serial Interface SCK Output Control** 0 = SCL pin is driven Low. 1 = SCL pin is tri-state.

Data Receive Register (RxR)

Register	Offset	R/W/C	Description	Reset Value
RxR	0XB800_6x10	R	Data Receive Register	0x0000_0000

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

Bits	Descriptions	
[7:0]	**Rx**	**Data Receive Register** The last byte received via I^2C bus will put on this register. The I^2C core only used 8-bit receive buffer.

Data Transmit Register (TxR)

Register	Offset	R/W/C	Description	Reset Value
TxR	0XB800_6x14	R/W	Data Transmit Register	0x0000_0000

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

Bits	Descriptions	
[31:0]	**Tx**	**Data Transmit Register** The I^2C core used 32-bit transmit buffer and provide multi-byte transmit function. Set CSR[Tx_NUM] to a value that you want to transmit. I^2C core will always issue a transfer from the highest byte first. For example, if CSR[Tx_NUM] = 0x3, Tx[31:24] will be transmitted first, then Tx[23:16], and so on. In case of a data transfer, all bits will be treated as data. In case of a slave address transfer, the first 7 bits will be treated as 7-bit address and the LSB represent the R/W bit. In this case, LSB = 1, reading from slave LSB = 0, writing to slave

7.16 Universal Serial Interface Controller (USI)

The USI is a synchronous serial interface performs a serial-to-parallel conversion on data characters received from the peripheral, and a parallel-to-serial conversion on data characters received from CPU. This interface can drive up to 2 external peripherals and is seen as the master. It can generate an interrupt signal when data transfer is finished and can be cleared by writing 1 to the interrupt flag. The active level of device/slave select signal can be chosen to low active or high active, which depends on the peripheral it's connected. Writing a divisor into DIVIDER register can program the frequency of serial clock output. This master core contains four 32-bit transmit/receive buffers, and can provide burst mode operation. The maximum bits can be transmitted/received is 32 bits, and can transmit/receive data up to four times successive.

The USI (MICROWIRE/SPI) Master Core includes the following features:

Support MICROWIRE/SPI master mode

Full duplex synchronous serial data transfer

Variable length of transfer word up to 32 bits

Provide burst mode operation, transmit/receive can be executed up to four times in one transfer

MSB or LSB first data transfer

Rx and Tx on both rising and falling edge of serial clock independently

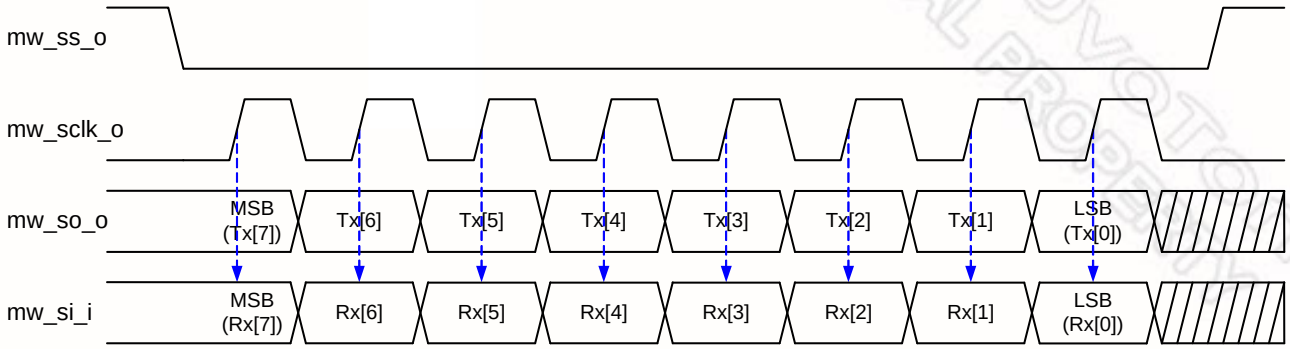
2 slave/device select lines

Fully static synchronous design with one clock domain

7.16.1 USI Timing Diagram

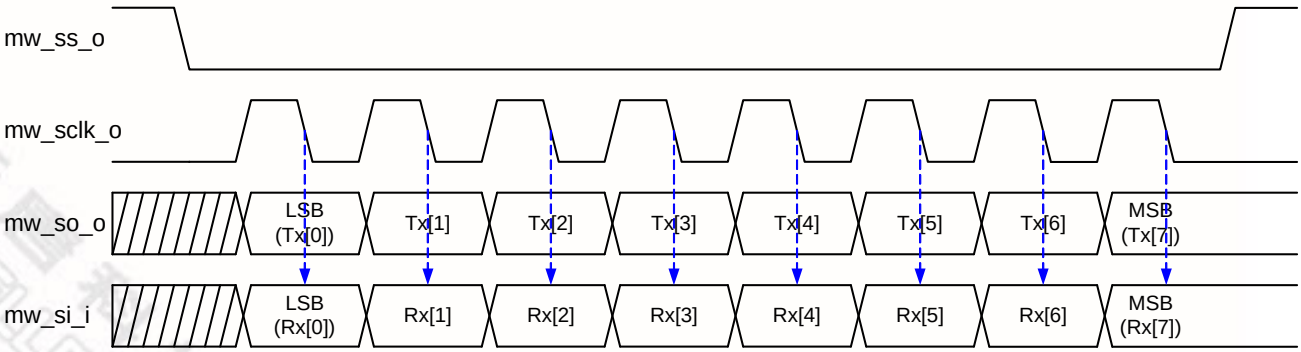
The timing diagram of USI is shown as following.

Pin descriptions:
SCLK (mw_sclk_o): USI serial clock output pin.
mw_int_o : USI interrupt signal output to AIC
SFAM(mw_ss_o) : USI slave/device select signal output.
SSPTXD(mw_so_o): USI serial data output pin (to slave device).
SSPRXD(mw_si_i) : USI serial data input pin (from slave device).



CNTRL[LSB]=0, CNTRL[Tx_NUM]=0x0, CNTRL[Tx_BIT_LEN]=0x08,
CNTRL[Tx_NEG]=1, CNTRL[Rx_NEG]=0, SSR[SS_LVL]=0

USI Timing



CNTRL[LSB]=1, CNTRL[Tx_NUM]=0x0, CNTRL[Tx_BIT_LEN]=0x08,
CNTRL[Tx_NEG]=0, CNTRL[Rx_NEG]=1, SSR[SS_LVL]=0

Alternate Phase SCLK Clock Timing

7.16.2 USI Control Registers Map

R: read only, **W**: write only, **R/W**: both read and write

Register	Offset	R/W	Description	Reset Value
USI_BA = 0xB800_6200				
CNTRL	0xB800_6200	R/W	Control and Status Register	0x0000_0004
DIVIDER	0xB800_6204	R/W	Clock Divider Register	0x0000_0000
SSR	0xB800_6208	R/W	Slave Select Register	0x0000_0000
Rx0	0xB800_6210	R	Data Receive Register 0	0x0000_0000
Rx1	0xB800_6214	R	Data Receive Register 1	0x0000_0000
Rx2	0xB800_6218	R	Data Receive Register 2	0x0000_0000
Rx3	0xB800_621C	R	Data Receive Register 3	0x0000_0000
Tx0	0xB800_6210	W	Data Transmit Register 0	0x0000_0000
Tx1	0xB800_6214	W	Data Transmit Register 1	0x0000_0000
Tx2	0xB800_6218	W	Data Transmit Register 2	0x0000_0000
Tx3	0xB800_621C	W	Data Transmit Register 3	0x0000_0000

NOTE 1: When software programs CNTRL, the GO_BUSY bit should be written last.

Control and Status Register (CNTRL)

Register	Offset	R/W	Description	Reset Value
CNTRL	0xB800_6200	R/W	Control and Status Register	0x0000_0004

31	30	29	28	27	26	25	24
CLK_POL	Reserved						
23	22	21	20	19	18	17	16
Reserved						IE	IF
15	14	13	12	11	10	9	8
SLEEP				Reserved	LSB	Tx_NUM	
7	6	5	4	3	2	1	0
Tx_BIT_LEN					Tx_NEG	Rx_NEG	GO_BUSY

Bits	Descriptions	
[31]	CLK_POL	**Clock Polarity** 0 = Normal polarity. 1 = Reverse polarity.
[17]	IE	**Interrupt Enable** 0 = Disable USI Interrupt. 1 = Enable USI Interrupt.
[16]	IF	**Interrupt Flag** 0 = It indicates that the transfer dose not finish yet. 1 = It indicates that the transfer is done. The interrupt flag is set if it was enable. NOTE: This bit is read only, but can be cleared by writing 1 to this bit.
[15:12]	SLEEP	**Suspend Interval** These four bits provide the configuration of suspend interval between two successive transmit/receive in a transfer. The default value is 0x0. When CNTRL[Tx_NUM] = 00, setting this field has no effect on transfer. The desired interval is obtained according to the following equation (from the last falling edge of current sclk to the first rising edge of next sclk): (CNTRL[SLEEP] + 2)*period of SCLK SLEEP = 0x0 … 2 SCLK clock cycle …… SLEEP = 0xf … 17 SCLK clock cycle

[10]	**LSB**	**Send LSB First** 0 = The **MSB** is transmitted/received first (which bit in TxX/RxX register that is depends on the Tx_BIT_LEN field in the CNTRL register). 1 = The **LSB** is sent first on the line (bit TxX[0]), and the first bit received from the line will be put in the LSB position in the Rx register (bit RxX[0]).
[9:8]	**Tx_NUM**	**Transmit/Receive Numbers** This field specifies how many transmit/receive numbers should be executed in one transfer. 00 = Only one transmit/receive will be executed in one transfer. 01 = Two successive transmit/receive will be executed in one transfer. 10 = Three successive transmit/receive will be executed in one transfer. 11 = Four successive transmit/receive will be executed in one transfer.
[7:3]	**Tx_BIT_LEN**	**Transmit Bit Length** This field specifies how many bits are transmitted in one transmit/receive. Up to 32 bits can be transmitted. Tx_BIT_LEN = 0x01 … 1 bit Tx_BIT_LEN = 0x02 … 2 bits …… Tx_BIT_LEN = 0x1f … 31 bits Tx_BIT_LEN = 0x00 … 32 bits
[2]	**Tx_NEG**	**Transmit On Negative Edge** 0 = The mw_so_o signal is changed on the **rising** edge of mw_sclk_o. 1 = The mw_so_o signal is changed on the **falling** edge of mw_sclk_o.
[1]	**Rx_NEG**	**Receive On Negative Edge** 0 = The mw_si_i signal is latched on the **rising** edge of mw_sclk_o. 1 = The mw_si_i signal is latched on the **falling** edge of mw_sclk_o.
[0]	**GO_BUSY**	**Go and Busy Status** 0 = Writing 0 to this bit has no effect. 1 = Writing 1 to this bit starts the transfer. This bit remains set during the transfer and is automatically cleared after transfer finished. **NOTE:** All registers should be set before writing 1 to the GO_BUSY bit in the CNTRL register. When a transfer is in progress, writing to any register of the USI master core has no effect.

Divider Register (DIVIDER)

Register	Offset	R/W	Description	Reset Value
DIVIDER	0xB800_6204	R/W	Clock Divider Register	0x0000_0000

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

Bits	Descriptions	
[15:0]	DIVIDER	**Clock Divider Register** The value in this field is the frequency divider of the system clock pclk to generate the serial clock on the output mw_sclk_o. The desired frequency is obtained according to the following equation: $f_{sclk} = \frac{f_{pclk}}{(DIVIDER+1)*2}$ **NOTE:** Suggest DIVIDER should be at least 1.

Slave Select Register (SSR)

Register	Offset	R/W	Description	Reset Value
SSR	0xB800_6208	R/W	Slave Select Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved				ASS	SS_LVL	SSR[1:0]	

Bits	Descriptions	
[3]	**ASS**	**Automatic Slave Select** 0 = If this bit is cleared, slave select signals are asserted and de-asserted by setting and clearing related bits in SSR register. 1 = If this bit is set, mw_ss_o signals are generated automatically. It means that device/slave select signal, which is set in SSR register is asserted by the USI controller when transmit/receive is started by setting CNTRL[GO_BUSY], and is de-asserted after every transmit/receive is finished.
[2]	**SS_LVL**	**Slave Select Active Level** It defines the active level of device/slave select signal (mw_ss_o). 0 = The mw_ss_o slave select signal is active Low. 1 = The mw_ss_o slave select signal is active High.
[1:0]	**SSR**	**Slave Select Register** If SSR[ASS] bit is cleared, writing 1 to any bit location of this field sets the proper mw_ss_o line to an active state and writing 0 sets the line back to inactive state. If SSR[ASS] bit is set, writing 1 to any bit location of this field will select appropriate mw_ss_o line to be automatically driven to active state for the duration of the transmit/receive, and will be driven to inactive state for the rest of the time. (The active level of mw_ss_o is specified in SSR[SS_LVL]). NOTE: This interface can only drive one device/slave at a given time. Therefore, the SSR of the selected device must be set to its active level before starting any read or write transfer.

Data Receive Register 0 (Rx0)

Data Receive Register 1 (Rx1)

Data Receive Register 2 (Rx2)

Data Receive Register 3 (Rx3)

Register	Offset	R/W	Description	Reset Value
Rx0	0xB800_6210	R	Data Receive Register 0	0x0000_0000
Rx1	0xB800_6214	R	Data Receive Register 1	0x0000_0000
Rx2	0xB800_6218	R	Data Receive Register 2	0x0000_0000
Rx3	0xB800_621C	R	Data Receive Register 3	0x0000_0000

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

Bits	Descriptions	
[31:0]	**Rx**	**Data Receive Register** The Data Receive Registers hold the value of received data of the last executed transfer. Valid bits depend on the transmit bit length field in the CNTRL register. For example, if CNTRL[Tx_BIT_LEN] is set to 0x08 and CNTRL[Tx_NUM] is set to 0x0, bit Rx0[7:0] holds the received data. NOTE: The Data Receive Registers are read only registers. A Write to these registers will actually modify the Data Transmit Registers because those registers share the same FFs.

Data Transmit Register 0 (Tx0)

Data Transmit Register 1 (Tx1)

Data Transmit Register 2 (Tx2)

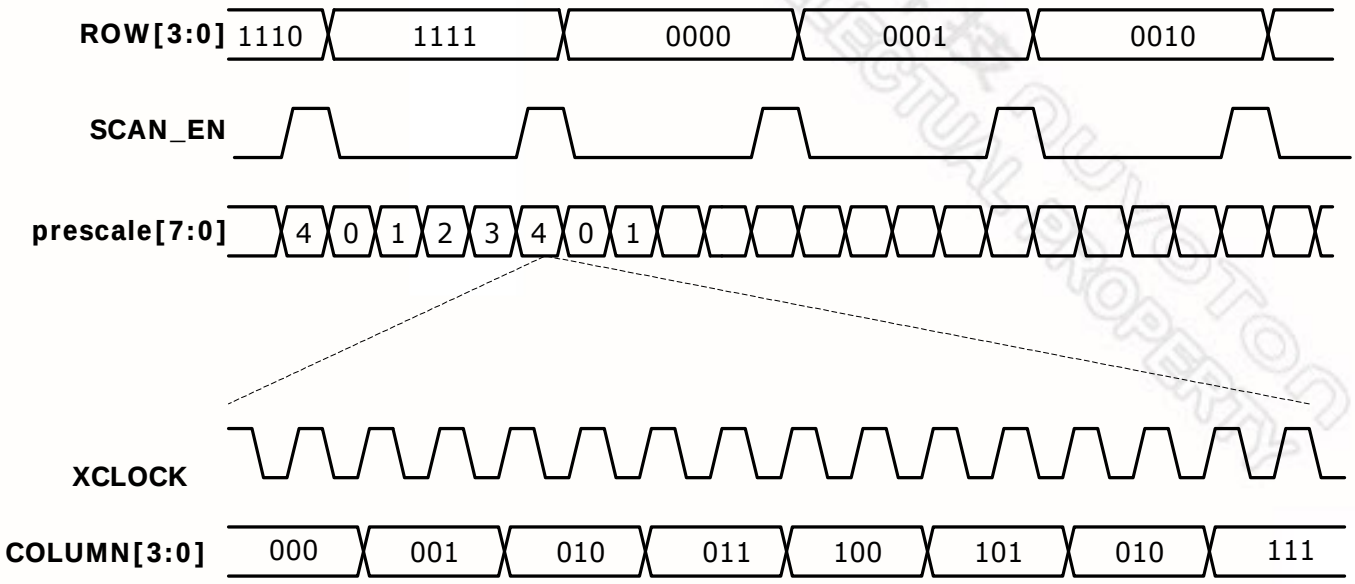
Data Transmit Register 3 (Tx3)

Register	Offset	R/W	Description	Reset Value
Tx0	0xB800_6210	W	Data Transmit Register 0	0x0000_0000
Tx1	0xB800_6214	W	Data Transmit Register 1	0x0000_0000
Tx2	0xB800_6218	W	Data Transmit Register 2	0x0000_0000
Tx3	0xB800_621C	W	Data Transmit Register 3	0x0000_0000

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

Bits	Descriptions	
[31:0]	**Tx**	**Data Transmit Register** The Data Transmit Registers hold the data to be transmitted in the next transfer. Valid bits depend on the transmit bit length field in the CNTRL register. For example, if CNTRL[Tx_BIT_LEN] is set to 0x08 and the CNTRL[Tx_NUM] is set to 0x0, the bit Tx0[7:0] will be transmitted in next transfer. If CNTRL[Tx_BIT_LEN] is set to 0x00 and CNTRL[Tx_NUM] is set to 0x3, the core will perform four 32-bit transmit/receive successive using the same setting (the order is Tx0[31:0], Tx1[31:0], Tx2[31:0], Tx3[31:0]). **NOTE:** The RxX and TxX registers share the same flip-flops, which mean that what is received from the input data line in one transfer will be transmitted on the output data line in the next transfer if no write access to the TxX register is executed between the transfers.

7.16.3 Timing Diagram



16x8 Keypad Scan Timing Diagram

8 Electrical Specifications

8.1 Absolute Maximum Ratings

Ambient temperature	TBD
Storage temperature	-50 °C ~ 125°C
Voltage on any pin	-0.5V ~ 6V
Power supply voltage (Core logic)	-0.5V ~ 2.5V
Power supply voltage (IO Buffer)	-0.5V ~ 4.6V
Injection current (latch-up testing)	100mA
Crystal Frequency	4MHz ~ 30MHz

8.2 DC Specifications

8.2.1 Digital DC Characteristics

(Normal test conditions: VDD33/AVDD33 = 3.3V+/-10%, VDD18/RTCVDD18/PLLVDD18 = 1.8V+/-10%, USBVDDC0/USBVDDC1/USBVDDT0/USBVDDT1= 3.3V+/-5%, TA = -40 °C ~ 85 °C unless otherwise specified)

SYMBOL	PARAMETER	CONDITION	MIN	TYP	MAX	UNIT
VDD33/ AVDD33	Power Supply		2.97	-	3.63	V
VDD18/ PLLVDD18	Power Supply		1.62	-	1.98	V
USBVDDC0/ USBVDDC1/ USBVDDT0/ USBVDDT1	Power Supply		3.13	-	3.46	V
V_{IL}	Input Low Voltage		-0.3	-	0.8	V
V_{IH}	Input High Voltage		2.0	-	5.5	V
VT+	Schmitt Trigger positive-going threshold		1.5	-	1.62	V
VT-	Schmitt trigger negative-going threshold		1.14	-	1.27	V
V_{OL}	Output Low Voltage	Depend on driving	-	-	0.4	V
V_{OH}	Output High Voltage	Depend on driving	2.4	-	-	V
I_{IH}	Input High Current	V_{IN} = 2.4 V	-1	-	1	uA
I_{IL}	Input Low Current	V_{IN} = 0.4 V	-1	-	1	uA
I_{OH}	Output High Current	EBI, GPIOC, GPIOD	-	35	-	mA
I_{OL}	Output Low Current	EBI, GPIOC, GPIOD	-	26	-	mA
I_{OH}	Output High Current	The other port	-	25	-	mA
I_{OL}	Output Low Current	The other port	-	17	-	mA
I_{OC}	Operation Current	Note 1	-	340	-	mA
I_{SC}	Standby Current	Note 2	-	50	-	uA

Note1:
This operation current is measured on VDD18 @ 1.8V, and all of IP clocks are enable with CPU clock/system clock @ 200MHz / 100MHz.
Note2:
The standby current is measured on VDD18 @1.8V, and all of IP clocks are disabling with power-down mode, all of GPIO pins are set to output and clock pins keep at 0V.

8.2.2 USB Low-/Full-Speed DC Electrical Specifications

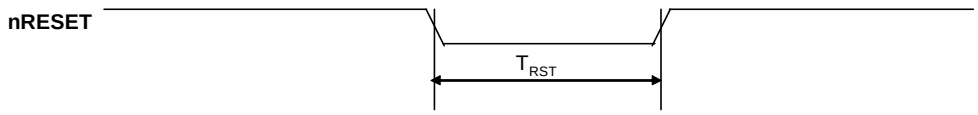
Symbol	Parameter	Conditions	MIN	TYP	MAX
V_{IH}	Pad input high voltage		2.0V		
V_{IL}	Pad input low voltage				0.8V
V_{DI}	Differential input sensitivity	|PADP-PADM|	0.2V		
V_{CM}	Common mode voltage range	include V_{DI} range	0.8V		2.5V
V_{SE}	Single-ended receiver threshold		0.8V		2.0V
V_{OL}	Pad output low voltage		0V		0.3V
V_{OH}	Pad output high voltage		2.8V		3.6V
V_{CRS}	Differential output signal cross-point voltage		1.3V		2.0V
R_{PU}	Internal pull-up resistor	Bus idle	900Ω		1575Ω
		Receiving	1425Ω		3090Ω
R_{PD}	Internal pull-down resistor		14.25KΩ		24.80KΩ
Z_{DRV}	Driver output resistance‡	Steady state drive		10Ω	
C_{IN}	Transceiver pad capacitance	Pad to ground			20pF

8.2.3 USB High-Speed DC Electrical Specifications

Symbol	Parameter	Conditions	MIN	TYP	MAX
V_{HSDI}	High-speed differential input signal level	|PADP-PADM|	150mV		
V_{HSSQ}	High-speed SQ detection threshold	|PADP-PADM|	100mV		150mV
V_{HSCM}	High-speed common mode voltage range		-50mV		500mV
V_{HSOH}	High-speed data signaling high		360mV		440mV
V_{HSOL}	High-speed data signaling low		-10mV		10mV
V_{CHIRPJ}	Chirp J level		700mV		1100mV
V_{CHIRPK}	Chirp K level		-900mV		-500mV
Z_{HSDRV}	High-speed driver output resistance	45Ω±10%	40.5Ω		49.5Ω

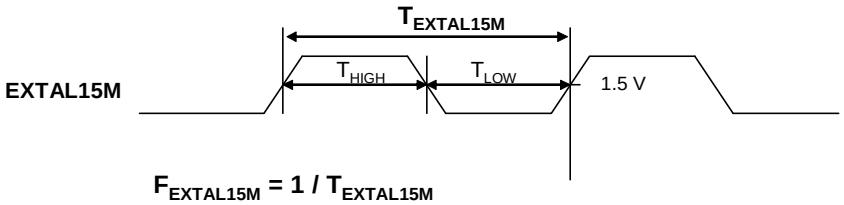
8.3 AC Specifications

8.3.1 RESET AC Characteristics



Symbol	Parameter	MIN	MAX	Unit
T_{RST}	Reset Pulse Width after Power stable	1.0	-	ms

8.3.2 Clock Input Characteristics

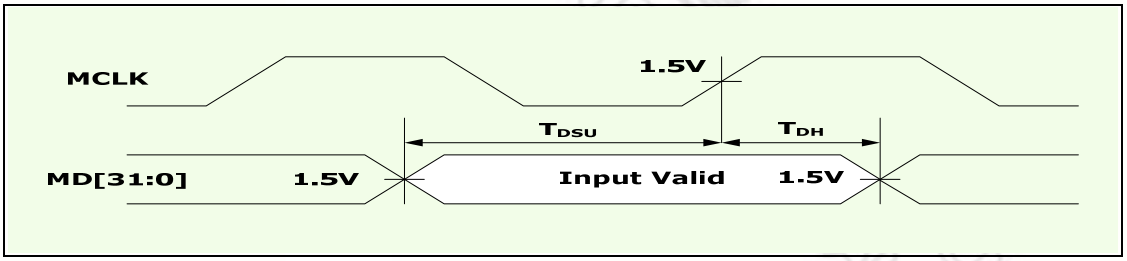


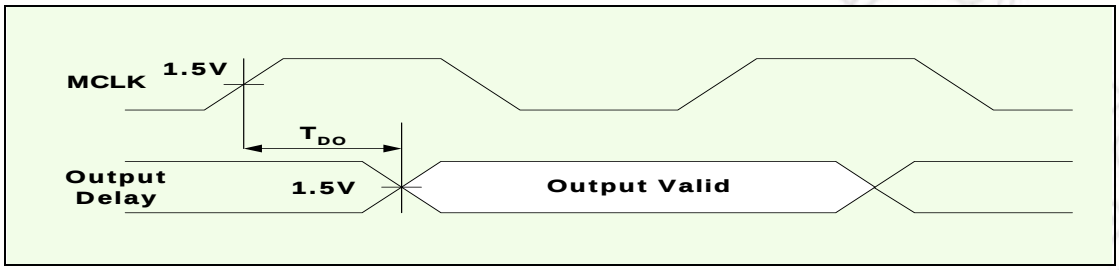
$F_{EXTAL15M} = 1 / T_{EXTAL15M}$

$EXTAL15M_{DUTY} = T_{HIGH} / (T_{HIGH} + T_{LOW})$

Symbol	Parameter	MIN	TYP	MAX	Unit
$F_{EXTAL15M}$	Clock Input Frequency	-	15.0	-	MHz
$EXTAL15M_{DUTY}$	Clock Input Duty Cycle	45	50	55	%
V_{IL} (EXTAL15M)	EXTAL15M Input Low Voltage	0	-	0.8	V
V_{IH} (EXTAL15M)	EXTAL15M Input High Voltage	2.0	-	VDD33 + 0.3	V

8.3.3 EBI/SDRAM Interface AC Characteristics

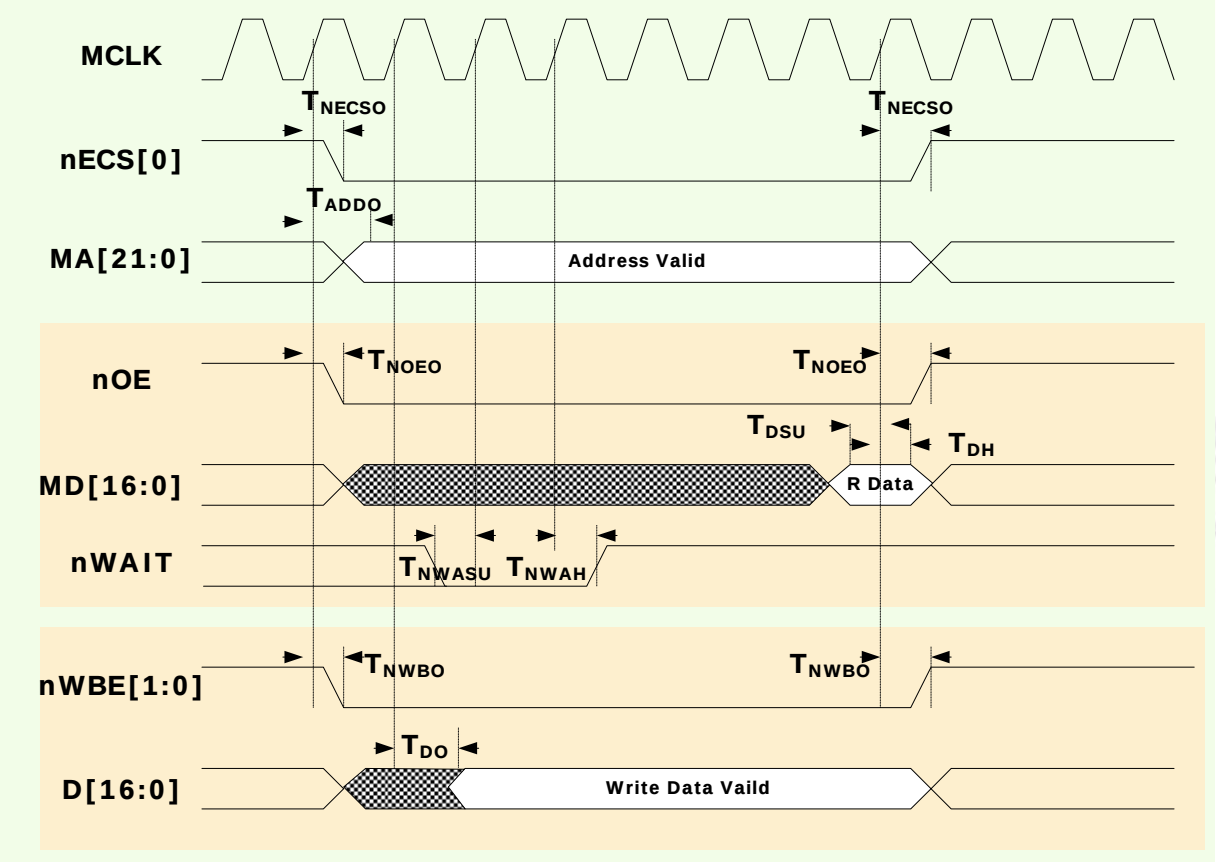




Symbol	Parameter	MIN	MAX	Unit
F_{MCLK}	SDRAM Clock Output Frequency	-	100	MHz
T_{DSU}	MD[31:0]] Input Setup Time	2	-	ns
T_{DH}	MD[31:0] Input Hold Time	2	-	ns
T_{OSU}	SDRAM Output Signal Valid Delay Time	2*	5*	ns

* The above T_{OSU} is based on the EBI CKSKEW register default setting on 0x48 and F_{MCLK} at 100MHz

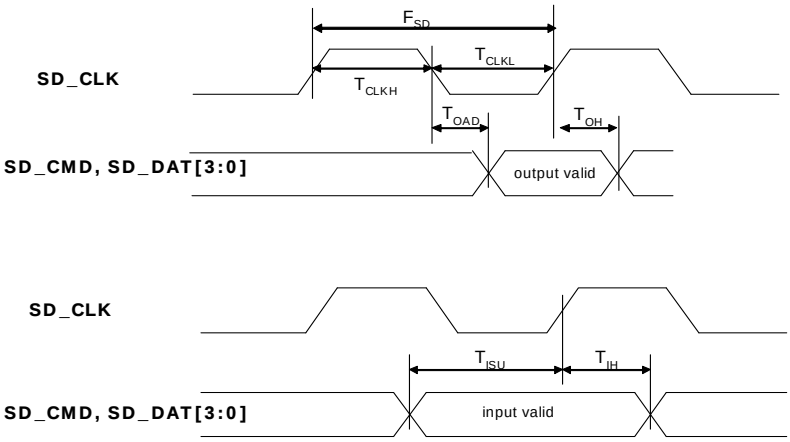
8.3.4 EBI (ROM/SRAM/External I/O) AC Characteristics



Symbol	Parameter	MIN	MAX	Unit
T_{ADDO}	Address Output Delay Time	2*	7*	ns
T_{NCSO}	ROM/SRAM/Flash or External I/O Chip Select Delay Time	2*	7*	ns
T_{NOEO}	ROM/SRAM or External I/O Bank Output Enable Delay	2*	7*	ns
T_{NWBO}	ROM/SRAM or External I/O Bank Write Byte Enable Delay	2*	7*	ns
T_{DH}	Read Data Hold Time	5		ns
T_{DSU}	Read Data Setup Time	1		ns
T_{DO}	Write Data Output Delay Time (SRAM or External I/O)	2*	7*	ns
T_{NWASU}	External Wait Setup Time	3		ns
T_{NWAH}	External Wait Hold Time	1		ns

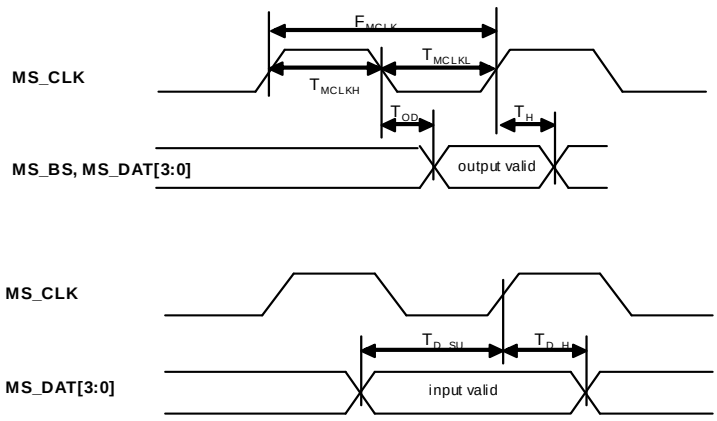
* The above data are based on the EBI CKSKEW register default setting on 0x48 and F_{MCLK} at 100MH

8.3.5 SD Host Interface AC Characteristics



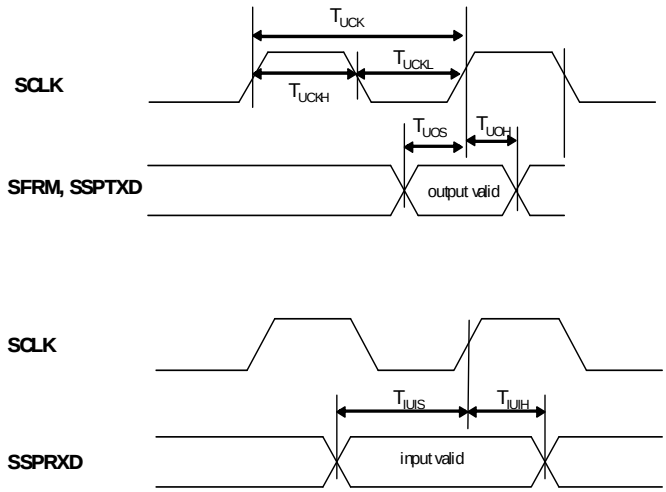
Symbol	Parameter	Conditions	MIN	MAX	Unit
F_{SD}	SD Clock Frequency	Identification Mode	**100**	**400**	KHz
F_{SD}	SD Clock Frequency	Data Transfer Mode	-	**50**	MHz
T_{CLKH}	SD Clock High Time	-	**10**	-	ns
T_{CLKL}	SD Clock Low Time	-	**10**	-	ns
T_{ISU}	SD CMD & Data Input Setup Time	-	**5**	-	ns
T_{IH}	SD CMD & Data Input Hold Time	-	**5**	-	ns
T_{OAD}	SD Output Active Delay (Falling Edge)	-	-	**14**	ns
T_{OH}	SD Output Hold Time	-	**0**	-	ns

8.3.6 Memory Stick Interface AC Characteristics



Symbol	Parameter	Conditions	MIN	MAX	Unit
F_{MCLK}	MS_CLK Clock Frequency	Serial Mode	**5**	**20**	MHz
F_{MCLK}	MS_CLK Clock Frequency	Parallel Mode	**10**	**40**	MHz
T_{MCLKH}	MS_CLK Clock High Time		**5**	-	ns
T_{MCLKL}	MS_CLK Clock Low Time		**5**	-	ns
T_{BS_OD}	MS_BS Output Delay (Falling Edge)		**5**	**15**	ns
T_{BS_H}	MS_BS Output Hold Time		**1**	-	ns
T_{D_SU}	Data Input Setup Time		**8**	-	ns
T_{D_H}	Data input Hold Time		**1**	-	ns
T_{D_OD}	Data Output Delay (Falling Edge)		**8**	**15**	ns
T_{D_OD}	Data Output Hold Time		**1**	-	ns

8.3.7 USI (SPI/MW) Interface AC Characteristics



Symbol	Parameter	MIN	MAX	Unit
T_{CLKH}	Clock Output High Time	**14.6**	-	ns
T_{CLKL}	Clock Output Low Time	**15.8**	-	ns
T_{CLK}	Clock Cycle Time	**30.4**	**-**	ns
T_{UOS}	SFRM, SSPTXD Output Setup Time	**15**	-	ns
T_{UOH}	SFRM, SSPTXD Output Hold Time	**13**	-	ns
T_{UIS}	SSPRXD Input Setup Time	**10**	-	ns
T_{UIH}	SSPRXD Input Hold Time	**10**	-	ns

8.3.8 USB Transceiver AC Characteristics

USB Transceiver: Low-Speed AC Electrical Specifications

Symbol	Parameter	Conditions	MIN	TYP	MAX
T_{LR}	Low-speed driver rise time	C_L=50pF	75ns		300ns
T_{LF}	Low-speed driver fall time	C_L=50pF	75ns		300ns
T_{LRFM}	Low-speed rise/fall time matching	$T_{LRFM} = T_{LR} / T_{LF}$	80%		125%

USB Transceiver: Full-Speed AC Electrical Specifications

Symbol	Parameter	Conditions	MIN	TYP	MAX
T_{FR}	Full-speed driver rise time	C_L=50pF	4ns		20ns
T_{FF}	Full-speed driver fall time	C_L=50pF	75ns		20ns
T_{FRFM}	Full-speed rise/fall time matching	$T_{FRFM} = T_{FR} / T_{FF}$	90%		111.11 %

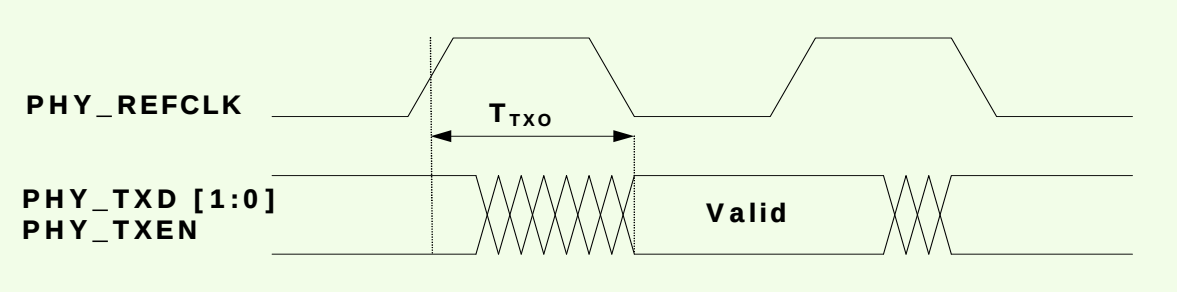
USB Transceiver: High-Speed AC Electrical Specifications

Symbol	Parameter	Conditions	MIN	TYP	MAX
T_{HSR}	High-speed driver rise time	Z_{HSDRV}=45Ω	500ps		900ps
T_{HSF}	High-speed driver fall time	Z_{HSDRV}=45Ω	500ps		900ps
	High-speed driver waveform requirement		Eye diagram of template 1**		
	High-speed receiver waveform requirement		Eye diagram of template 4††		
	High-speed jitter requirement	Data source end	Eye diagram of template 1**		
		Receiver end	Eye diagram of template 4††		

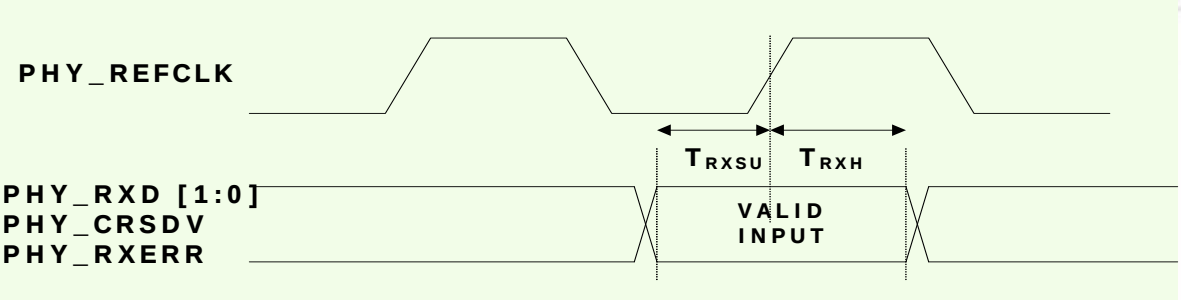
** Check "Universal Serial Bus Specification Revision 2.0" page 133.
†† Check "Universal Serial Bus Specification Revision 2.0" page 136.

8.3.9 EMC RMII AC Characteristics

The signal timing characteristics conforms to the guidelines specified in IEEE Std. 802.3.

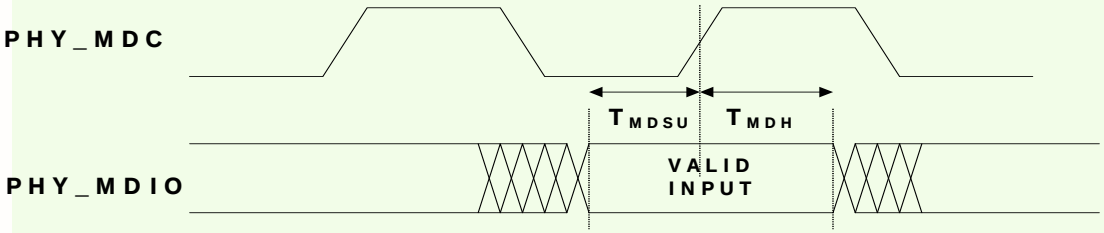


Transmit Signal Timing Relationships at RMII

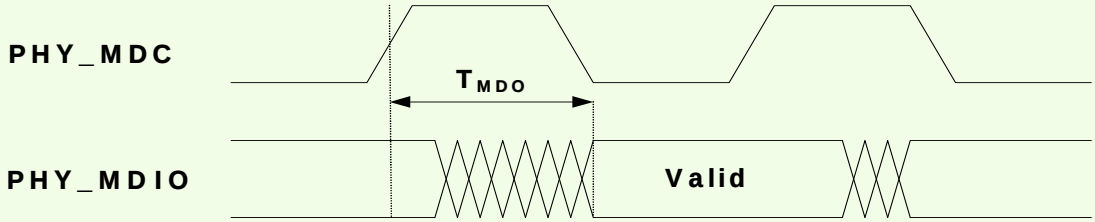


Receive Signal Timing Relationships at RMII

Symbol	Parameter	MIN	MAX	Unit
T_{TxO}	Transmit Output Delay Time	7	14	ns
T_{RxSU}	Receive Setup Time	4		ns
T_{RxH}	Receive Hold Time	2		ns



PHY_MDIO Read from PHY Timing

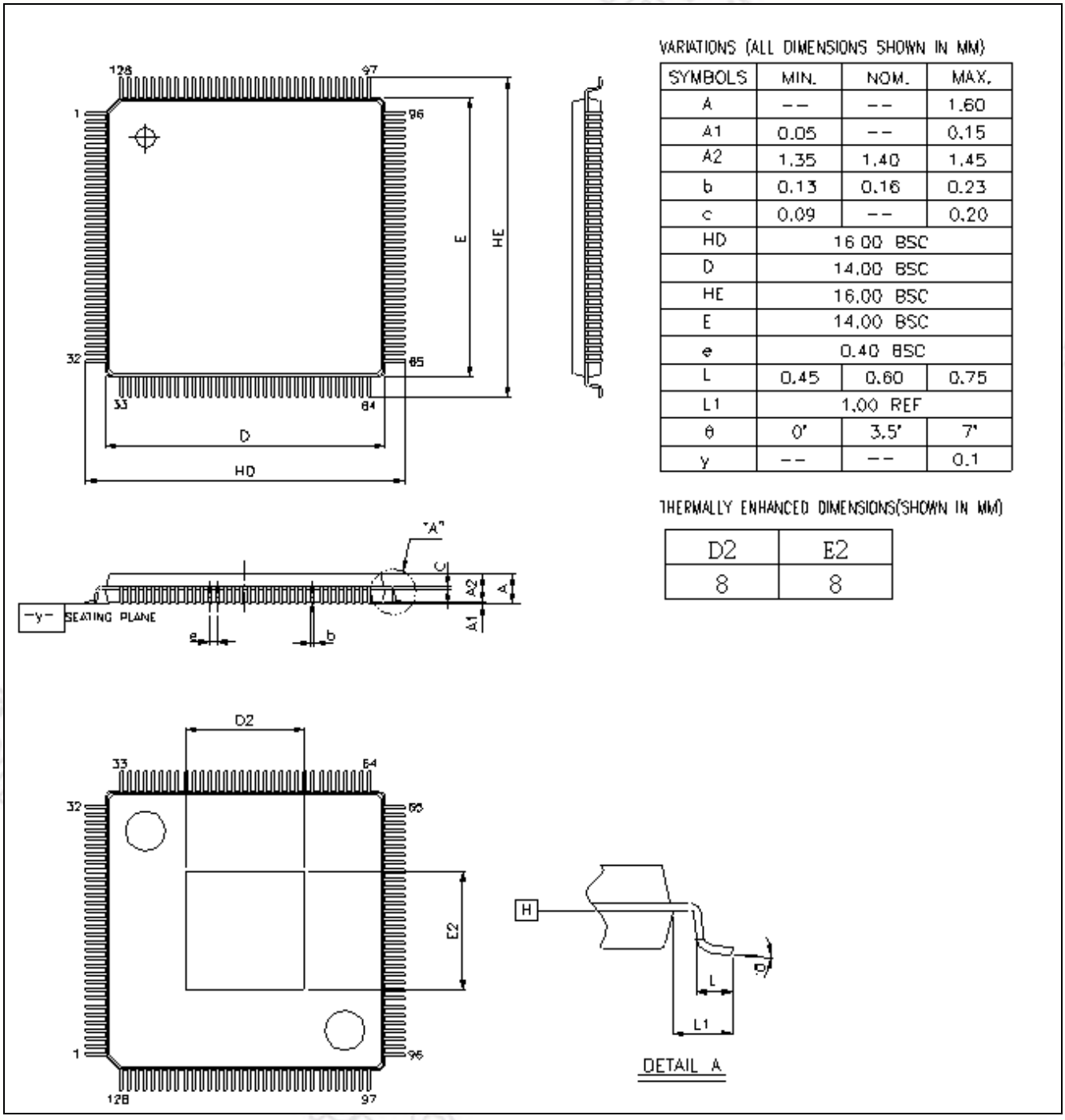


PHY_MDIO Write to PHY Timing

Symbol	Parameter	MIN	MAX	Unit
T_{MDO}	PHY_MDIO Output Delay Time	0	15	ns
T_{MDSU}	PHY_MDIO Setup Time	5		ns
T_{MDH}	PHY_MDIO Hold Time	5		ns

9 Package Specifications

NUC946ADN LQFP128L (14X14X1.4 mm, footprint 2.0mm)



VARIATIONS (ALL DIMENSIONS SHOWN IN MM)

SYMBOLS	MIN.	NOM.	MAX.
A	--	--	1.60
A1	0.05	--	0.15
A2	1.35	1.40	1.45
b	0.13	0.16	0.23
c	0.09	--	0.20
HD	16.00 BSC		
D	14.00 BSC		
HE	16.00 BSC		
E	14.00 BSC		
e	0.40 BSC		
L	0.45	0.60	0.75
L1	1.00 REF		
θ	0°	3.5°	7°
y	--	--	0.1

THERMALLY ENHANCED DIMENSIONS(SHOWN IN MM)

D2	E2
8	8

10 Revision History

Revision	Date	Comments
A4	2011/03/10	1. New release
A5	2011/7/26	1. Change footprint with ex-pad

nuvoTon

32-BIT ARM926EJ-S BASED MCU

Important Notice

Nuvoton Products are neither intended nor warranted for usage in systems or equipment, any malfunction or failure of which may cause loss of human life, bodily injury or severe property damage. Such applications are deemed, "Insecure Usage".

Insecure usage includes, but is not limited to: equipment for surgical implementation, atomic energy control instruments, airplane or spaceship instruments, the control or operation of dynamic, brake or safety systems designed for vehicular use, traffic signal instruments, all types of safety devices, and other applications intended to support or sustain life.

All Insecure Usage shall be made at customer's risk, and in the event that third parties lay claims to Nuvoton as a result of customer's Insecure Usage, customer shall indemnify the damages and liabilities thus incurred by Nuvoton.