W90N745CD/W90N745CDG

winbond

32-BIT ARM7TDMI-BASED MCU

W90N745
16/32-bit ARM microcontroller
Product Data Sheet

Revision History

REVISION	DATE	COMMENTS
A	2006/06/23	Draft
A1	2006/08/30	Add Electrical specification
A2	2006/09/22	Delete Chapter 6: BLOCK DIAGRAM

Table of Contents-

1. GENERAL DESCRIPTION

The W90N745 is built around an outstanding CPU core, the 16/32 ARM7TDMI RISC processor which designed by Advanced RISC Machines, Ltd. It offers 4K-byte I-cache/SRAM and 4K-byte D-cache/SRAM, is a low power, general purpose integrated circuits. Its simple, elegant, and fully static design is particularly suitable for cost sensitive and power sensitive applications.

One 100/10 Mbit MAC of Ethernet controller is built-in to reduce total system cost.

The W90N745 also provides one USB 1.1 host controller, one USB 1.1 device controller, one AC97/I²S controller, one 2-channel GDMA, four independent UARTs, one watchdog timer, two 24-bit timers with 8-bit pre-scale, up to 31 programmable I/O ports, PS2 keyboard controller and an advanced interrupt controller. The external bus interface (EBI) controller provides for SDRAM, ROM/SRAM, flash memory and I/O devices. The system manager includes an internal 32-bit system bus arbiter and a PLL clock controller.

With a wide range of serial communication and Ethernet interfaces, the W90N745 is suitable for communication gateways as well as many other general purpose applications.

2. FEATURES

Architecture

- Fully 16/32-bit RISC architecture
- Little/Big-Endian mode supported
- Efficient and powerful ARM7TDMI core
- Cost-effective JTAG-based debug solution

External Bus Interface

- 8/16-bit external bus support for ROM/SRAM, flash memory, SDRAM and external I/Os
- Support for SDRAM
- Programmable access cycle (0-7 wait cycle)
- Four-word depth write buffer for SDRAM write data
- Cost-effective memory-to-peripheral DMA interface

Instruction and Data Cache

- Two-way, set-associative, 4K-byte I-cache and 4K-byte D-cache
- Support for LRU (Least Recently Used) protocol
- Cache can be configured as internal SRAM
- Support cache lock function

Ethernet MAC Controller

- DMA engine with burst mode
- MAC Tx/Rx buffers (256 bytes Tx, 256 bytes Rx)
- Data alignment logic
- Endian translation
- 100/10 Mbit per second operation
- 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
- PAD generation

DMA Controller

- 2-channel general DMA for memory-to-memory data transfers without CPU intervention
- Initialed by a software or external DMA request
- Increments or decrements a source or destination address in 8-bit, 16-bit or 32-bit data transfers
- 4-data burst mode

UART

- Four 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, ½ or 2 stop bits
- Odd or even parity
- Break generation and detection
- Parity, overrun and framing error detection
- X16 clock mode
- UART1 supports Bluetooth, and UART2 supports IrDA1.0 SIR

Timers

- Two programmable 24-bit timers with 8-bit pre-scaler
- One programmable 20 bit with selectable additional 8-bit prescaler watchdog timer
- One-shot mode, periodical mode or toggle mode operation

Programmable I/Os

- 31 programmable I/O ports
- Pins individually configurable to input, output or I/O mode for dedicated signals
- I/O ports are configurable for multiple functions

Advanced Interrupt Controller

- 24 interrupt sources, including 4 external interrupt sources
- Programmable normal or fast interrupt mode (IRQ, FIQ)
- Programmable as either edge-triggered or level-sensitive for 4 external interrupt sources
- Programmable as either low-active or high-active for 4 external interrupt sources
- Priority methodology is encoded to allow for interrupt daisy-chaining
- Automatically mask out the lower priority interrupt during interrupt nesting

USB Host Controller

- USB 1.1 compliant
- Compatible with Open HCI 1.0 specification
- Supports low-speed and full speed devices
- Build-in DMA for real time data transfer
- Two on-chip USB transceivers with one optionally shared with USB device controller

USB Device Controller

- USB 1.1 compliant
- Support four USB endpoints including one control endpoint and 3 configurable endpoints for rich USB functions

Two PLLs

- The external clock can be multiplied by on-chip PLL to provide high frequency system clock
- The input frequency range is 3-30MHz; 15MHz is preferred.
- One PLL for both CPU and USB host/device controller
- One PLL for audio I²S 12.288/16.934MHz clock source
- Programmable clock frequency

4-Channel PWM

- Four 16-bit timers with PWM
- Two 8-bit pre-scalers & Two 4-bit dividers
- Programmable duty control of output waveform (PWM)
- Auto reload mode or one-shot pulse mode
- Dead-zone generator

I^2C Master

- 2-channel I^2C
- Compatible with Philips I^2C standard, support master mode only
- Support 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)

- 1-channel USI
- Support USI (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 or falling edge of serial clock independently
- Two slave/device select lines
- Fully static synchronous design with one clock domain

2-Channel AC97/I²S Audio Codec Host Interface

- AHB master port and an AHB slave port are offered in audio controller.
- Always 8-beat incrementing burst
- Always bus lock when 8-beat incrementing burst
- When reach middle and end address of destination address, a DMA_IRQ is requested to CPU automatically

KeyPad Scan Interface

- Scan up to 16 rows by 8 columns with an external 4 to 16 decoder and 4x8 array without auxiliary component
- Programmable debounce time
- One or two keys scan with interrupt and three keys reset function.
- Wakeup CPU from IDEL/Power Down mode

PS2 Host Interface

- APB slave consisted of PS2 protocol.
- Connect IBM keyboard or bar-code reader through PS2 interface.
- Provide hardware scan code to ASCII translation

Power management

- Programmable clock enables for individual peripheral
- IDLE mode to halt ARM core and keep peripheral working
- Power-Down mode to stop all clocks included external crystal oscillator.
- Exit IDLE by all interrupts
    - Exit Power-Down by keypad,USB device and external interrupts

Operation Voltage Range

- 3.0 ~ 3.6 V for IO buffer
- 1.62 ~ 1.98 V for core logic

Operation Temperature Range

- TBD

Operating Frequency

- Up to 80 MHz

Package Type

- 128-pin LQFP

3. PIN DIAGRAM

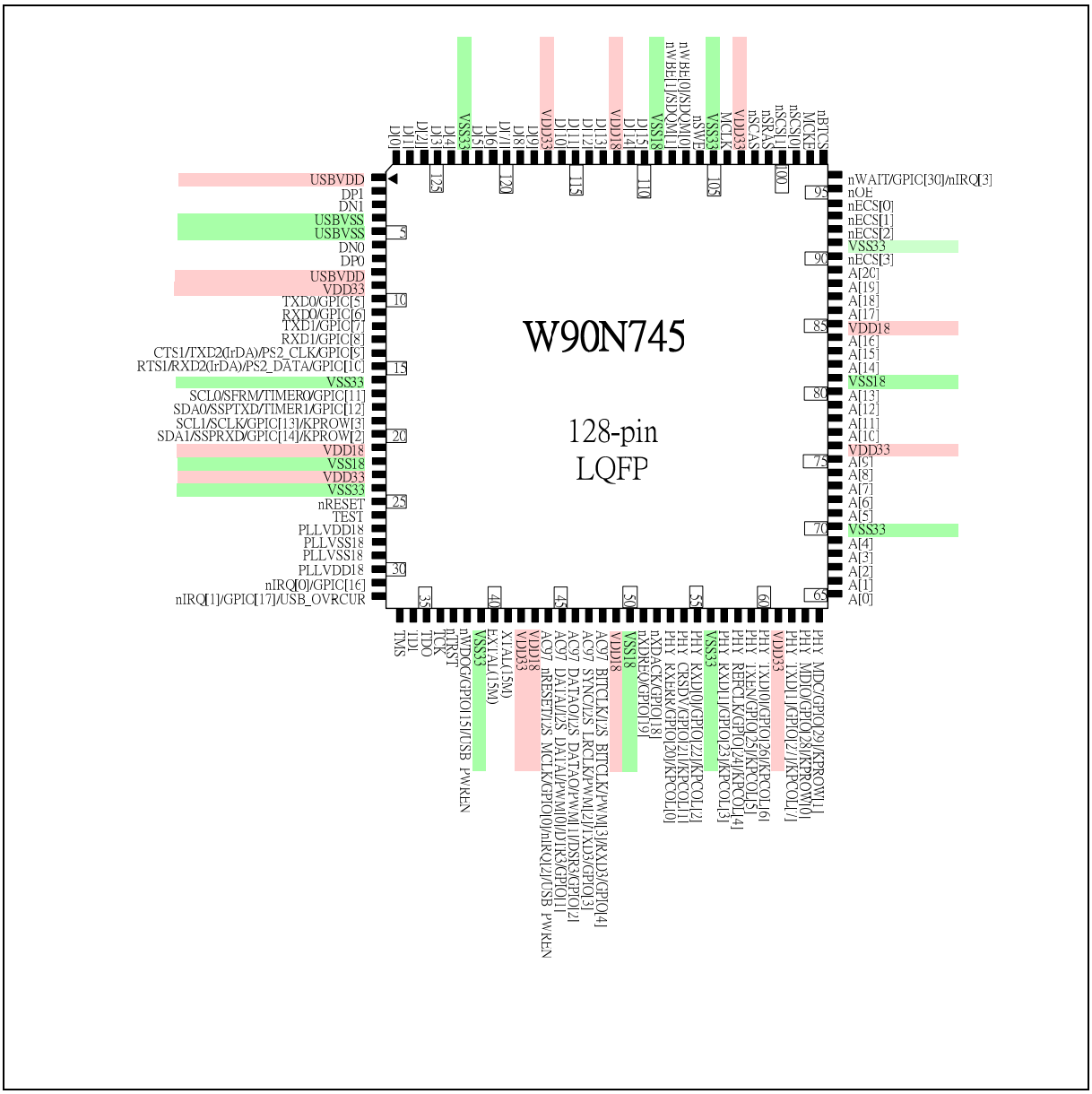


Figure 3.1 Pin Diagram

4. PIN ASSIGNMENT

Table 4.1 W90N745 Pins Assignment

PIN NAME	128-PIN LQFP
Clock & Reset	**(3 pins)**
EXTAL (15M)	40
XTAL (15M)	41
nRESET	25
JTAG Interface	**(5 pins)**
TMS	33
TDI	34
TDO	35
TCK	36
nTRST	37
External Bus Interface	**(53 pins)**
A [20:0]	89-86,84-82,80-77,75-71,69-65
D [15:0]	110-111,113-116,118-122,124-128
nWBE [1;0] / SDQM [1:0]	108,107
nSCS [1:0]	100,99
nSRAS	101
nSCAS	102
MCKE	98
nSWE	106
MCLK	104
nWAIT / GPIO [30] / nIRQ [3]	96
nBTCS	97
nECS [3:0]	90,92-94
nOE	95

Table 4.1 W90N745 Pins Assignment, continued

PIN NAME	128-PIN LQFP
Ethernet Interface	**(10 pins)**
PHY_MDC / GPIO [29] / KPROW [1]	64
PHY_MDIO / GPIO [28] / KPROW [0]	63
PHY_TXD [1:0] / GPIO [27:26] / KPCOL [7:6]	62,60
PHY_TXEN / GPIO [25] / KPCOL [5]	59
PHY_REFCLK / GPIO [24] / KPCOL [4]	58
PHY_RXD [1:0] / GPIO [23:22] / KPCOL [3:2]	57,55
PHY_CRSDV / GPIO [21] / KPCOL [1]	54
PHY_RXERR / GPIO [20] / KPCOL [0]	53
AC97/I²S/PWM/UART3	**(5 pins)**
AC97_nRESET / I²S_MCLK / GPIO [0] / nIRQ [2] / USB_PWREN	44

Table 4.1 W90N745 Pins Assignment, continued

PIN NAME	128-PIN LQFP
AC97/I²S/PWM/UART3	**(5 pins)**
AC97_DATAI / I²S_DATAI / PWM [0] / DTR3 / GPIO [1]	45
AC97_DATAO / I²S_DATAO / PWM [1] / DSR3 / GPIO [2]	46
AC97_SYNC / I²S_LRCLK / PWM [2] / TXD3 / GPIO [3]	47
AC97_BITCLK / I²S_BITCLK / PWM [3] / RXD3 GPIO [4]	48
USB Interface	**(4 pins)**
DP0	7
DN 0	6
DP1	2
DN1	3
Miscellaneous	**(7 pins)**
nIRQ [1] / GPIO [17] / USB_OVRCUR	32
nIRQ [0] / GPIO [16]	31
nWDOG / GPIO [15] / USB_PWREN	38
TEST	26

Table 4.1 W90N745 Pins Assignment, continued

PIN NAME	128-PIN LQFP
I²C/USI(SPI/MW)	**(4 pins)**
SCL0 / SFRM / TIMER0 / GPIO [11]	17
SDA0 / SSPTXD / TIMER1 / GPIO [12]	18
SCL1 / SCLK / GPIO [13] / KPROW [3]	19
SDA1 / SSPRXD / GPIO [14] / KPROW [2]	20
UART0/UART1/UART2/PS2	**(6 pins)**
TXD0 / GPIO [5]	10
RXD0 / GPIO [6]	11
TXD1 / GPIO [7]	12
RXD1 / GPIO [8]	13
CTS1 / TXD2(IrDA) / PS2_CLK / GPIO [9]	14
RTS1 / RXD2(IrDA) / PS2_DATA / GPIO [10]	15

Table 4.1 W90N745 Pins Assignment, continued

PIN NAME	128-PIN LQFP
XDMA	**(2 pins)**
nXDREQ / GPIO [19] /	51
nXDACK / GPIO [18] /	52
Power/Ground	**(36 pins)**
VDD18	21,43,49,85,112
VSS18	22,50,81,109
VDD33	9,23,42,61,76,103,117
VSS33	16,24,39,56,70,91,105,123
USBVDD	1,8
USBVSS	4,5
PLLVDD18	27,30
PLLVSS18	28,29

5. PIN DESCRIPTION

Table 5.1 W90N745 Pins Description

PIN NAME	IO TYPE	DESCRIPTION
Clock & Reset		
EXTAL (15M)	I	15MHz External Clock / Crystal Input
XTAL (15M)	O	15MHz Crystal Output
nRESET	IS	System Reset, active-low
JTAG Interface		
TMS	IUS	JTAG Test Mode Select, internal pull-up with 70K ohm
TDI	IUS	JTAG Test Data in, internal pull-up with 70K ohm
TDO	O	JTAG Test Data out
TCK	IDS	JTAG Test Clock, internal pull-down with 58K ohm
nTRST	IUS	JTAG Reset, active-low, internal pull-up with 70K ohm
External Bus Interface		
A [20:18]	O	Address Bus (MSB) of external memory and IO devices.
A [17:0]	IOS	Address Bus of external memory and IO devices.
D [15:0]	IOS	Data Bus (LSB) of external memory and IO device.
nWBE [1:0] / SDQM [1:0]	IOS	Write Byte Enable for specific device (nECS [1:0]). Data Bus Mask signal for SDRAM (nSCS [1:0]), active-low.
nSCS [1:0]	O	SDRAM chip select for two external banks, active-low.
nSRAS	O	Row Address Strobe for SDRAM, active-low.
nSCAS	O	Column Address Strobe for SDRAM, active-low.
MCKE	O	SDRAM Clock Enable, active-high
nSWE	O	SDRAM Write Enable, active-low
MCLK	O	System Master Clock Out, SDRAM clock, output with slew-rate control
nWAIT / GPIO[30] / nIRQ3	IUS	External Wait, active-low. This pin indicates that the external devices need more active cycle during access operation. General Programmable In/Out Port GPIO[30]. If memory and IO devices in EBI do not need wait request, it can be configured as GPIO[30] or nIRQ3.
nBTCS	O	ROM/Flash Chip Select, active-low.
nECS [3:0]	IO	External I/O Chip Select, active-low.
nOE	O	ROM/Flash, External Memory Output Enable, active-low.

Table 5.1 W90N745 Pins Description, continued

PIN NAME	IO TYPE	DESCRIPTION
Ethernet Interface		
PHY_MDC / GPIO [29] / KPROW [1]	IOU	RMII Management Data Clock for Ethernet. It is the reference clock of MDIO. Each MDIO data will be latched at the rising edge of MDC clock. General Programmable In/Out Port [29] Keypad ROW[1] scan output.
PHY_MDIO / GPIO [28] / KPROW [0]	IO	RMII Management Data I/O for Ethernet. It is used to transfer RMII control and status information between PHY and MAC. General Programmable In/Out Port [28] Keypad ROW[0] scan output.
PHY_TXD [1:0] / GPIO [27:26] / KPCOL [7:6]	IOU	2-bit Transmit Data bus for Ethernet. General programmable In/Out Port [27:26] Keypad column input [7:6], active low
PHY_TXEN / GPIO [25] / KPCOL [5]	IOU	PHY_TXEN shall be asserted synchronously with the first 2-bit of the preamble and shall remain asserted while all di-bits to be transmitted are presented. Of course, it is synchronized with PHY_REFCLK. General Programmable In/Out Port [25] Keypad column input [5], active low
PHY_REFCLK / GPIO [24] / KPCOL [4]	IOS	Reference Clock. The clock shall be 50MHz +/- 50 ppm with minimum 35% duty cycle at high or low state. General Programmable In/Out port [24] Keypad column input [4], active low
PHY_RXD [1:0] / GPIO [23:22] / KPCOL [3:2]	IOS	2-bit Receive Data bus for Ethernet. General Programmable In/Out Port [23:22] Keypad column input [3:2], active low
PHY_CRSDV / GPIO [21] / KPCOL [1]	IOS	Carrier Sense / Receive Data Valid for Ethernet. The PHY_CRSDV shall be asserted by PHY when the receive medium is non-idle. Loss of carrier shall result in the de-assertion of PHY_CRSDV synchronous to the cycle of PHY_REFCLK, and only on 2-bit receive data boundaries. General Programmable In/Out port [21] Keypad column input [1], active low
PHY_RXERR / GPIO [20] / KPCOL [0]	IOS	Receive Data Error for Ethernet. It indicates a data error detected by PHY.The assertion should be lasted for longer than a period of PHY_REFCLK. When PHY_RXERR is asserted, the MAC will report a CRC error. General programmable In/Out port [20] Keypad column input [0], active low

Table 5.1 W90N745 Pins Description, continued

PIN NAME	IO TYPE	DESCRIPTION
AC97/I²S/PWM/UART3		
AC97_nRESET / I²S_MCLK / GPIO [0] / nIRQ [2] / USB_PWREN	IOU	AC97 CODEC Host Interface RESET Output. I²S CODEC Host Interface System Clock Output. General Purpose In/Out port [0] External interrupt request. USB host power enable output
AC97_DATAI / I²S_DATAI / PWM [0] / DTR3 / GPIO [1]	IOU	AC97 CODEC Host Interface Data Input. I²S CODEC Host Interface Data Input. PWM Channel 0 output. Data Terminal Ready for UART3. General Purpose In /Out port [1]
AC97_DATAO / I²S_DATAO / PWM [1] / DSR3 / GPIO [2]	IOU	AC97 CODEC Host Interface Data Output. I²S CODEC Host Interface Data Output. PWM Channel 1 output. Data Set Ready for UART3. General Purpose In/Out port [2]
AC97_SYNC / I²S_LRCLK / PWM [2] / TXD3 / GPIO [3]	IOU	AC97 CODEC Host Interface Synchronous Pulse Output. I²S CODEC Host Interface Left/Right Channel Select Clock. PWM Channel 2 output. Transmit Data for UART3. General Purpose In/Out port [3]
AC97_BITCLK / I²S_BITCLK / PWM [3] / RXD3 / GPIO [4]	IOS	AC97 CODEC Host Interface Bit Clock Input. I²S CODEC Host Interface Bit Clock. PWM Channel 3 output. Receive Data for UART3. General Purpose In/Out port [4].
USB Interface		
DP0	IO	Differential Positive USB IO signal
DN0	IO	Differential Negative USB IO signal
DP1	IO	Differential Positive USB IO signal
DN1	IO	Differential Negative USB IO signal
Miscellaneous		
nIRQ [1:0] / GPIO [17:16] / USB_OVRCUR	IOU	External Interrupt Request General Purpose I/O nIRQ1 is used as USB host over-current detection input
nWDOG / GPIO [15] / USB_PWREN	IOU	Watchdog Timer Timeout Flag and Keypad 3-keys reset output, active low General Purpose In/output USB host power switch enable output
TEST	IDS	This test pin must be short to ground or left unconnected

Table 5.1 W90N745 Pins Description, continued

PIN NAME	IO TYPE	DESCRIPTION
I²C/USI		
SCL0 / SFRM / TIMER0 / GPIO [11]	IOU	I²C Serial Clock Line 0. USI Serial Frame. Timer0 time out output. General Purpose In/Out port [11].
SDA0 / SSPTXD / TIMER1 / GPIO [12]	IOU	I²C Serial Data Line 0 USI Serial Transmit Data Timer1 time out output General Purpose In/Out port [12]
SCL1 / SCLK / GPIO [13] / KPROW [3]	IOU	I²C Serial Clock Line 1 USI Serial Clock General Purpose In/Out port [13] Keypad row scan output [3]
SDA1 / SSPRXD / GPIO [14] / KPROW [2]	IDU	I²C Serial Data Line 1 USI Serial Receive Data General Purpose In/Out port [14] Keypad scan output [2]
UART0/UART1/UART2		
TXD0 / GPIO [5]	IOU	UART0 Transmit Data. General Purpose In/Out [5]
RXD0 / GPIO [6]	IOU	UART0 Receive Data. General Purpose In/Out [6]
TXD1 / GPIO [7]	IOU	UART1 Transmit Data. General Purpose In/Out [7]
RXD1 / GPIO [8]	IOU	UART1 Receive Data. General Purpose In/Out [8]
CTS1/ TXD2(IrDA) / PS2_CLK / GPIO [9]	IOU	UART1 Clear To Send for Bluetooth application UART2 Transmit Data supporting SIR IrDA. PS2 Interface Clock Input/Output General Purpose In/Out [9]
RTS1/ RXD2(IrDA) / PS2_DATA / GPIO [10]	IOU	UART1 Request To Send for Bluetooth application UART2 Receive Data supporting SIR IrDA. PS2 Interface Bi-Directional Data Line. General Purpose In/Out [10]
XDMA		
nXDREQ / GPIO [19] /	IO	External DMA Request. General Purpose In/Out [19]
nXDACK / GPIO [18] /	IO	External DMA Acknowledgement. General Purpose In/Out [18]

Table 5.1 W90N745 Pins Description, continued

PIN NAME	IO TYPE	DESCRIPTION
Power/Ground		
VDD18	P	Core Logic power (1.8V)
VSS18	G	Core Logic ground (0V)
VDD33	P	IO Buffer power (3.3V)
VSS33	G	IO Buffer ground (0V)
USBVDD	P	USB power (3.3V)
USBVSS	G	USB ground (0V)
DVDD18	P	PLL Digital power (1.8V)
DVSS18	G	PLL Digital ground (0V)
AVDD18	P	PLL Analog power (1.8V)
AVSS18	G	PLL Analog ground (0V)

Table 5.2 W90N745 128-pin LQFP Multi-function List

PIN NO.	DEFAULT	FUNCTION0	FUNCTION1	FUNCTION2	FUNCTION3
USB1.1 Host/Device Interface					
1	**USBVDD**	USBVDD	-	-	-
2	**DP1**	DP1	-	-	-
3	**DN1**	DN1	-	-	-
4	**USBVSS**	USBVSS	-	-	-
5	**USBVSS**	USBVSS	-	-	-
6	**DN0**	DN0	-	-	-
7	**DP0**	DP0	-	-	-
8	**USBVDD**	USBVDD	-	-	-
9	**VDD33**	VDD33	-	-	-
UART[2:0]/PS2 Interface					
10	**GPIO[5]**	GPIO[5]	UART_TXD0	-	-
11	**GPIO[6]**	GPIO[6]	UART_RXD0	-	-
12	**GPIO[7]**	GPIO[7]	UART_TXD1	-	-
13	**GPIO[8]**	GPIO[8]	UART_RXD1	-	-
14	**GPIO[9]**	GPIO[9]	UART_TXD2	UART_CTS1	PS2_CLK
15	**GPIO[10]**	GPIO[10]	UART_RXD2	UART_RTS1	PS2_DATA
16	**VSS33**	VSS33	-	-	-
I^2C/USI Interface					
17	**GPIO[11]**	GPIO[11]	I^2C_SCL0	SSP_FRAM	TIMER0
18	**GPIO[12]**	GPIO[12]	I^2C_SDA0	SSP_TXD	TIMER1
19	**GPIO[13]**	GPIO[13]	I^2C_SCL1	SSP_SCLK	KPI_ROW[3]
20	**GPIO[14]**	GPIO[14]	I^2C_SDA1	SSP_RXD	KPI_ROW[2]
21	**VDD18**	VDD18	-	-	-
22	**VSS18**	VSS18	-	-	-
23	**VDD33**	VDD33	-	-	-
24	**VSS33**	VSS33	-	-	-
System Reset & TEST					
25	**nRESET**	nRESET	-	-	-
26	**TEST**	TEST	-	-	-

Table 5.2 W90N745 128-pin LQFP Multi-function List, continued

PLL Power/Ground					
27	**PLL_VDD18**	PLL_VDD18	-	-	-
28	**PLL_VSS18**	PLL_VSS18	-	-	-
29	**PLL_VSS18**	PLL_VSS18	-	-	-
30	**PLL_VDD18**	PLL_VDD18	-	-	-
External IRQ[1:0]/USB Over Current					
31	**GPIO[16]**	GPIO[16]	nIRQ [0]	-	-
32	**GPIO[17]**	GPIO[17]	nIRQ [1]	USB_OVRCUR	-
JTAG Interface					
33	**TMS**	TMS	-	-	-
34	**TDI**	TDI	-	-	-
35	**TDO**	TDO	-	-	-
36	**TCK**	TCK	-	-	-
37	**nTRST**	nTRST	-	-	-
WatchDog/USB Power Enable					
38	**GPIO[15]**	GPIO[15]	nWDOG	USB_PWREN	-
39	**VSS33**	VSS33	-	-	-
System Clock					
40	**EXTAL(15M)**	EXTAL(15M)	-	-	-
41	**XTAL(15M)**	XTAL(15M)	-	-	-
42	**VDD33**	VDD33	-	-	-
43	**VDD18**	VDD18	-	-	-

Table 5.2 W90N745 128-pin LQFP Multi-function List, continued

PIN NO.	DEFAULT	FUNCTION0	FUNCTION1	FUNCTION2	FUNCTION3
AC97/I²S/PWM/UART3 Interface					
44	**GPIO[0]**	GPIO[0]	AC97_nRESET or I²SMCLK	nIRQ [2]	USB_PWREN
45	**GPIO[1]**	GPIO[1]	AC97_DATAI or I²SDATAI	PWM0	UART_DTR3
46	**GPIO[2]**	GPIO[2]	AC97_DATAO or I²SDATAO	PWM1	UART_DSR3
47	**GPIO[3]**	GPIO[3]	AC97_SYNC or I²SLRCLK	PWM2	UART_TXD3
48	**GPIO[4]**	GPIO[4]	AC97_BITCLK or I²SBITCLK	PWM3	UART_RXD3
49	**VDD18**	VDD18	-	-	-
50	**VSS18**	VSS18	-	-	-
XDMAREQ					
51	**GPIO[19]**	GPIO[19]	nXDREQ	-	-
52	**GPIO[18]**	GPIO[18]	nXDACK	-	-
Ethernet RMII/KeyPad Interface					
53	**GPIO[20]**	GPIO[20]	PHY_RXERR	KPI_COL[0]	-
54	**GPIO[21]**	GPIO[21]	PHY_CRSDV	KPI_COL[1]	-
55	**GPIO[22]**	GPIO[22]	PHY_RXD[0]	KPI_COL[2]	-
56	**VSS33**	VSS33	-	-	-
57	**GPIO[23]**	GPIO[23]	PHY_RXD[1]	KPI_COL[3]	-
58	**GPIO[24]**	GPIO[24]	PHY_REFCLK	KPI_COL[4]	-
59	**GPIO[25]**	GPIO[25]	PHY_TXEN	KPI_COL[5]	-
60	**GPIO[26]**	GPIO[26]	PHY_TXD[0]	KPI_COL[6]	-
61	**VDD33**	VDD33	-	-	-
62	**GPIO[27]**	GPIO[27]	PHY_TXD[1]	KPI_COL[7]	-
63	**GPIO[28]**	GPIO[28]	PHY_MDIO	KPI_ROW[0]	
64	**GPIO[29]**	GPIO[29]	PHY_MDC	KPI_ROW[1]	

Table 5.2 W90N745 128-pin LQFP Multi-function List, continued

PIN NO.	DEFAULT	FUNCTION0	FUNCTION1	FUNCTION2	FUNCTION3
Memory Address/Data/Control					
65	**A[0]**	A[0]	-	-	-
66	**A[1]**	A[1]	-	-	-
67	**A[2]**	A[2]	-	-	-
68	**A[3]**	A[3]	-	-	-
69	**A[4]**	A[4]	-	-	-
70	**VSS33**	VSS33	-	-	-
71	**A[5]**	A[5]	-	-	-
72	**A[6]**	A[6]	-	-	-
73	**A[7]**	A[7]	-	-	-
74	**A[8]**	A[8]	-	-	-
75	**A[9]**	A[9]	-	-	-
76	**VDD33**	VDD33	-	-	-
77	**A[10]**	A[10]	-	-	-
78	**A[11]**	A[11]	-	-	-
79	**A[12]**	A[12]	-	-	-
80	**A[13]**	A[13]	-	-	-
81	**VSS18**	VSS18	-	-	-
82	**A[14]**	A[14]	-	-	-
83	**A[15]**	A[15]	-	-	-
84	**A[16]**	A[16]	-	-	-
85	**VDD18**	VDD18	-	-	-
86	**A[17]**	A[17]	-	-	-
87	**A[18]**	A[18]	-	-	-
88	**A[19]**	A[19]	-	-	-
89	**A[20]**	A[20]	-	-	-
90	**nECS[3]**	nECS[3]	-	-	-
91	**VSS33**	VSS33	-	-	-

Table 5.2 W90N745 128-pin LQFP Multi-function List, continued

PIN NO.	DEFAULT	FUNCTION0	FUNCTION1	FUNCTION2	FUNCTION3
Memory Address/Data/Control					
92	**nECS[2]**	nECS[2]	-	-	-
93	**nECS[1]**	nECS[1]	-	-	-
94	**nECS[0]**	nECS[0]	-	-	-
95	**nOE**	nOE	-	-	-
96	**nWAIT**	GPIO[30]	nWAIT	nIRQ [3]	-
97	**nBTCS**	nBTCS	-	-	-
98	**MCKE**	MCKE	-	-	-
99	**nSCS[0]**	nSCS[0]	-	-	-
100	**nSCS[1]**	nSCS[1]	-	-	-
101	**nSRAS**	nSRAS	-	-	-
102	**nSCAS**	nSCAS	-	-	-
103	**VDD33**	VDD33	-	-	-
104	**MCLK**	MCLK	-	-	-
105	**VSS33**	VSS33	-	-	-
106	**nSWE**	nSWE	-	-	-
107	**nWBE/SDQM[0]**	nWBE or SDQM[0]			
108	**nWBE/SDQM[1]**	nWBE or SDQM[1]			
109	**VSS18**	VSS18	-	-	-
110	**D[15]**	D[15]	-	-	-
111	**D[14]**	D[14]	-	-	-
112	**VDD18**	VDD18	-	-	-
113	**D[13]**	D[13]	-	-	-
114	**D[12]**	D[12]	-	-	-
115	**D[11]**	D[11]	-	-	-
116	**D[10]**	D[10]	-	-	-
117	**VDD33**	VDD33	-	-	-
118	**D[9]**	D[9]	-	-	-
119	**D[8]**	D[8]	-	-	-
120	**D[7]**	D[7]	-	-	-

Table 5.2 W90N745 128-pin LQFP Multi-function List, continued

PIN NO.	DEFAULT	FUNCTION0	FUNCTION1	FUNCTION2	FUNCTION3
Memory Address/Data/Control					
121	**D[6]**	D[6]	-	-	-
122	**D[5]**	D[5]	-	-	-
123	**VSS33**	VSS33	-	-	-
124	**D[4]**	D[4]	-	-	-
125	**D[3]**	D[3]	-	-	-
126	**D[2]**	D[2]	-	-	-
127	**D[1]**	D[1]	-	-	-
128	**D[0]**	D[0]	-	-	-

6. FUNCTIONAL DESCRIPTION

6.1 ARM7TDMI CPU CORE

The ARM7TDMI CPU core is a member of the Advanced RISC Machines (ARM) family of general-purpose 32-bit microprocessors, which offer high performance for very low power consumption. The architecture is based on Reduced Instruction Set Computer (RISC) principles, and the instruction set and related decode mechanism are much simpler than those of micro-programmed Complex Instruction Set Computers. Pipelining is employed so that all parts of the processing and memory systems can operate continuously. The high instruction throughput and impressive real-time interrupt response are the major benefits.

The ARM7TDMI CPU core has two instruction sets:

(1) The standard 32-bit ARM set

(2) A 16-bit THUMB set

The THUMB set's 16-bit instruction length allows it to approach twice the density of standard ARM core while retaining most of the ARM's performance advantage over a traditional 16-bit processor using 16-bit registers. THUMB instructions operate with the standard ARM register configuration, allowing excellent interoperability between ARM and THUMB states. Each 16-bit THUMB instruction has a corresponding 32-bit ARM instruction with the same effect on the processor model.

ARM7TDMI CPU core has 31 x 32-bit registers. At any one time, 16 sets are visible; the other registers are used to speed up exception processing. All the register specified in ARM instructions can address any of the 16 registers. The CPU also supports 5 types of exception, such as two levels of interrupt, memory aborts, attempted execution of an undefined instruction and software interrupts.

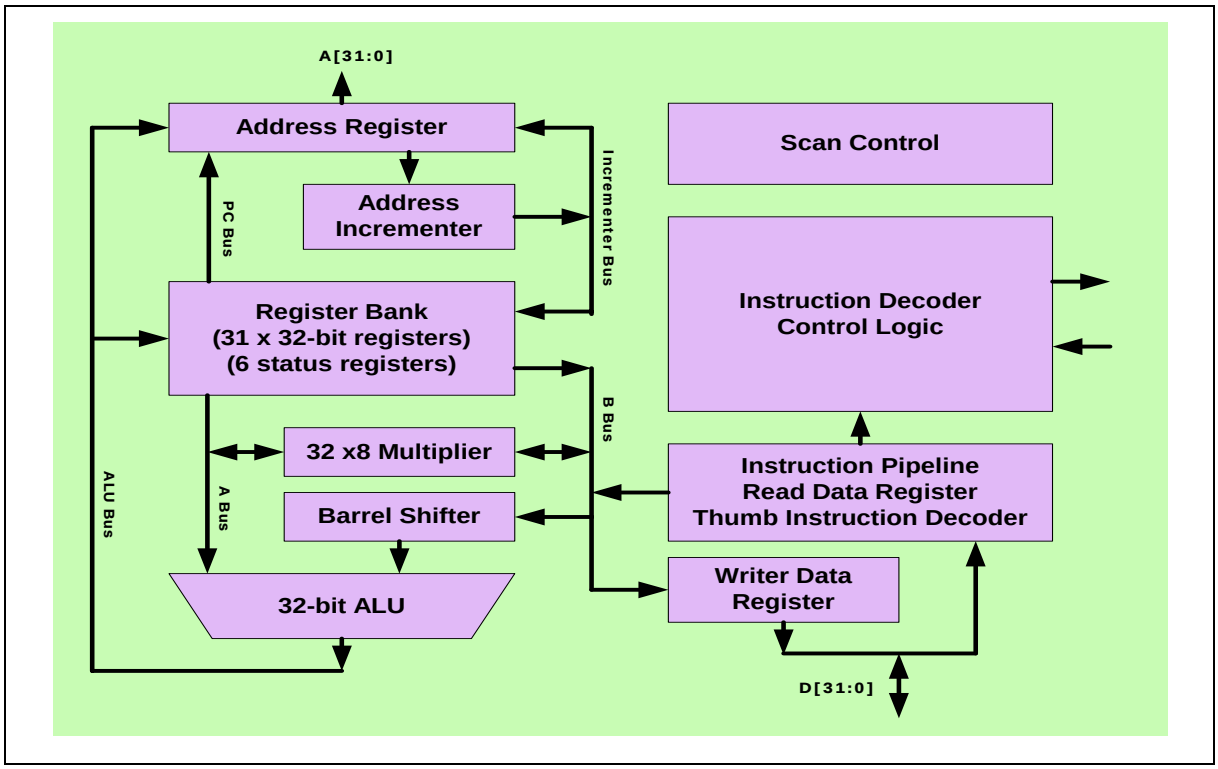


Figure 6.1.1 ARM7TDMI CPU Core Block Diagram

6.2 System Manager

6.2.1 Overview

The W90N745 system manager has the following functions.

- System memory map
- Data bus connection with external memory
- Product identifier register
- Bus arbitration
- PLL module
- Clock select and power saving control register
- Power-On setting

6.2.2 System Memory Map

W90N745 provides 2G bytes cacheable address space and the other 2G bytes are non-cacheable. The On-Chip Peripherals bank is on 1M bytes top of the space (0xFFF0_0000 – 0xFFFF_FFFF) and the On-Chip RAM bank's start address is 0xFFE0.0000, the other banks can be located anywhere (cacheable space:0x0000_0000~0x7FDF_FFFF if Cache ON; non-cacheable space: 0x8000_0000~0xFFDF_FFFF).

The size and location of each bank is determined by the register settings for "current bank base address pointer" and "current bank size". Please note that when setting the bank control registers, the address boundaries of consecutive banks must not overlap.

Except On-Chip Peripherals and On-Chip RAM, 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".

In the event of an access requested to an address outside any programmed bank size, an abort signal is generated. The maximum accessible memory size of each external IO bank is 4M bytes (by word format), and 64M bytes on each SDRAM bank.

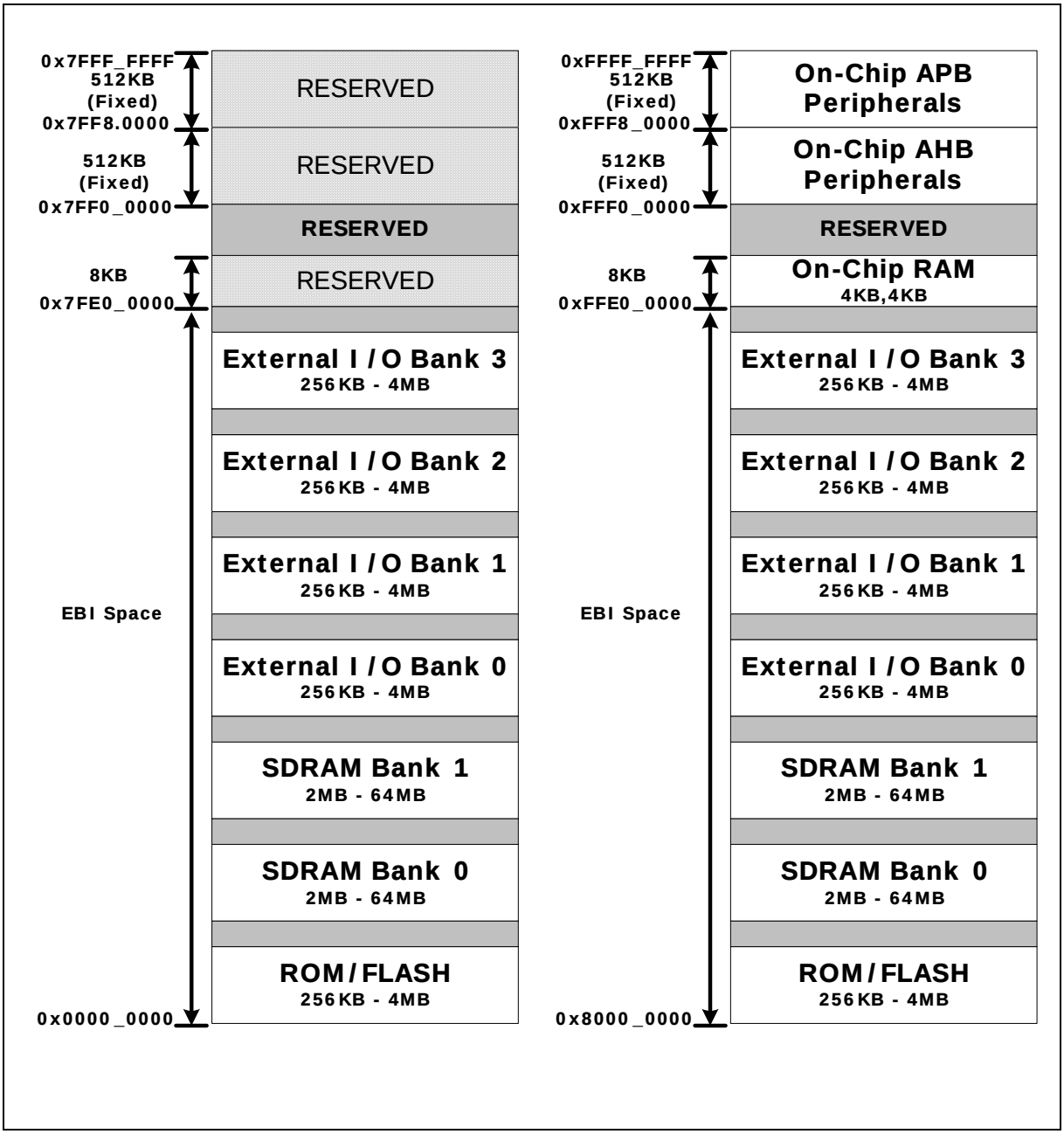


Figure 6.2.1 System Memory Map

Table 6.2.1 On-Chip Peripherals Memory Map

BASE ADDRESS	DESCRIPTION
AHB Peripherals	
0xFFF0_0000	Product Identifier Register (PDID)
0xFFF0_0004	Arbitration Control Register (ARBCON)
0xFFF0_0008	PLL Control Register 0(PLLCON0)
0xFFF0_000C	Clock Select Register (CLKSEL)
0xFFF0_0010	PLL Control Register 1 (PLLCON1)
0xFFF0_0014	Audio I²S Clock Control Register (I²SCKCON)
0xFFF0_0020	IRQ Wakeup Control Register (IRQWAKEUPCON)
0xFFF0_0024	IRQ Wakeup Flag Register (IRQWAKEFLAG)
0xFFF0_0028	Power Manager Control Register (PMCON)
0xFFF0_0030	USB Transceiver Control Register (USBTXRCON)
0xFFF0_1000	EBI Control Register (EBICON) Control Registers
0xFFF0_1004	ROM/FLASH (ROMCON) Control Registers
0xFFF0_1008	SDRAM bank 0 – 1 Control Registers
0xFFF0_1018	External I/O 0 – 3 Control Registers
0xFFF0_2000	Cache Controller Control Registers
0xFFF0_3000	Ethernet MAC Controller Control Registers
0xFFF0_4000	GDMA 0 – 1 Control Registers
0xFFF0_5000	USB Host Controller Control Registers
0xFFF0_6000	USB Device Controller Control Registers
0xFFF0_9000	AC97/I²S Controller Control Registers
APB Peripherals	
0xFFF8_0000	UART 0 (Tx, RX for console)
0xFFF8_0100	UART 1 (Tx, Rx, for bluetooth)
0xFFF8_0200	UART 2 (bluetooth CTS, RTS/ IrDA Tx, Rx)
0xFFF8_0300	UART 3 (micro-print DTR, DTS, Tx, Rx)
0xFFF8_1000	Timer 0 – 1, WDOG Timer
0xFFF8_2000	Interrupt Controller
0xFFF8_3000	GPIO
0xFFF8_6000	I^2C-0 Control Registers
0xFFF8_6100	I^2C-1 Control Registers
0xFFF8_6200	USI Control Registers
0xFFF8_7000	Pulse Width Modulation (PWM) Control Registers
0xFFF8_8000	KeyPad Interface Control Register (KPI)
0xFFF8_9000	PS2 Control Registers

6.2.3 Address Bus Generation

The W90N745 address bus generation is depended on the required data bus width of each memory bank. The data bus width is determined by **DBWD** bits in each bank's control register.

The maximum accessible memory size of each external IO bank is 4M bytes.

Table 6.2.2 Address Bus Generation Guidelines

DATA BUS	EXTERNAL ADDRESS PINS	MAXIMUM ACCESSIBLE MEMORY SIZE
WIDTH	A [20:0]	
8-bit	A20 – A0 (Internal)	2M bytes
16-bit	A21 – A1 (Internal)	2M half-words

6.2.4 Data Bus Connection with External Memory

6.2.4.1. Memory formats

The W90N745 can be configured as big endian or little endian mode by pull up or down the external data bus D14 pin. If D14 is pull up, then it is a little endian mode, otherwise, it is a big endian mode.

Little endian

In little endian format, the lowest addressed byte in a word is considered the least significant byte of the word and the highest addressed byte is the most significant. So the byte at address 0 of the memory system connects to data lines 7 through 0.

For a word aligned address A, Figure 6.2.2 shows how the half-word at addresses A and A+2, and the bytes at addresses A, A+1, A+2, and A+3 map on to each other when **D14** pin is High.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Half-word at address A															
Half-word at address A+2															
Byte at address A+1								Byte at address A							
Byte at address A+3								Byte at address A+2							

Figure 6.2.2 Little endian addresses of bytes and half-words within half words

Big endian

In Big endian format, the W90N745 stores the most significant byte of a word at the lowest numbered byte, and the least significant byte at the highest-numbered byte. So the byte at address 0 of the memory system connects to data lines 31 through 24.

For a word aligned address A, Figure 6.2.3 shows how the half-word at addresses A and A+2, and the bytes at addresses A, A+1, A+2, and A+3 map on to each other when the **D14** pin is Low.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Half-word at address A															
Half-word at address A+2															
Byte at address A								**Byte at address A+1**							
Byte at address A+2								**Byte at address A+3**							

Figure 6.2.3 Big endian addresses of bytes and half-words within half words

6.2.4.2. Connection of External Memory with Various Data Width

The system diagram for W90N745 connecting with the external memory is shown in Figure 6.2.4. Below tables (Table 6.2.3 through Table 6.2.14) show the program/data path between CPU register and the external memory using little / big endian and word/half-word/byte access.

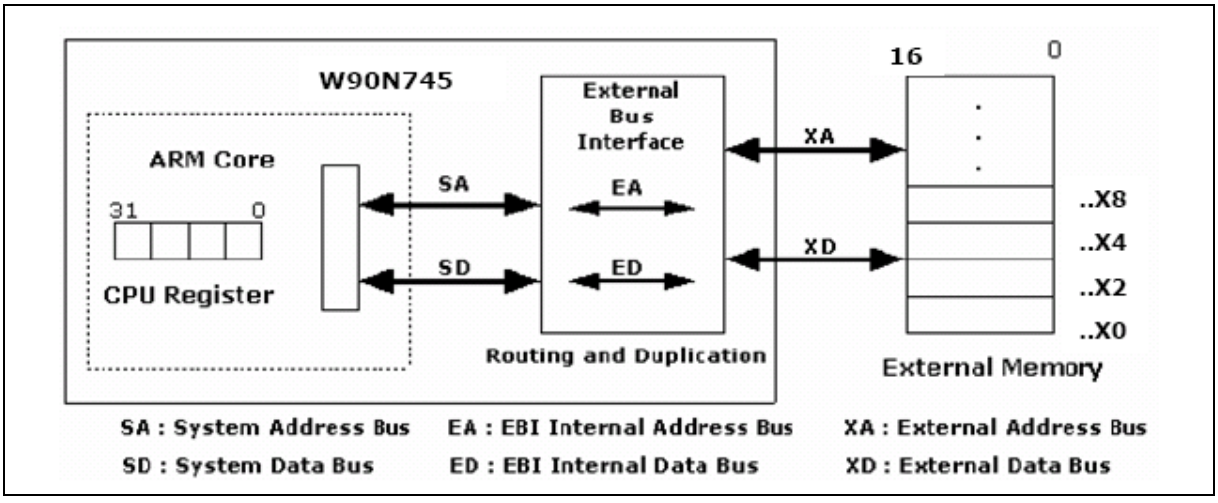


Figure 6.2.4 Address/Data bus connection with external memory

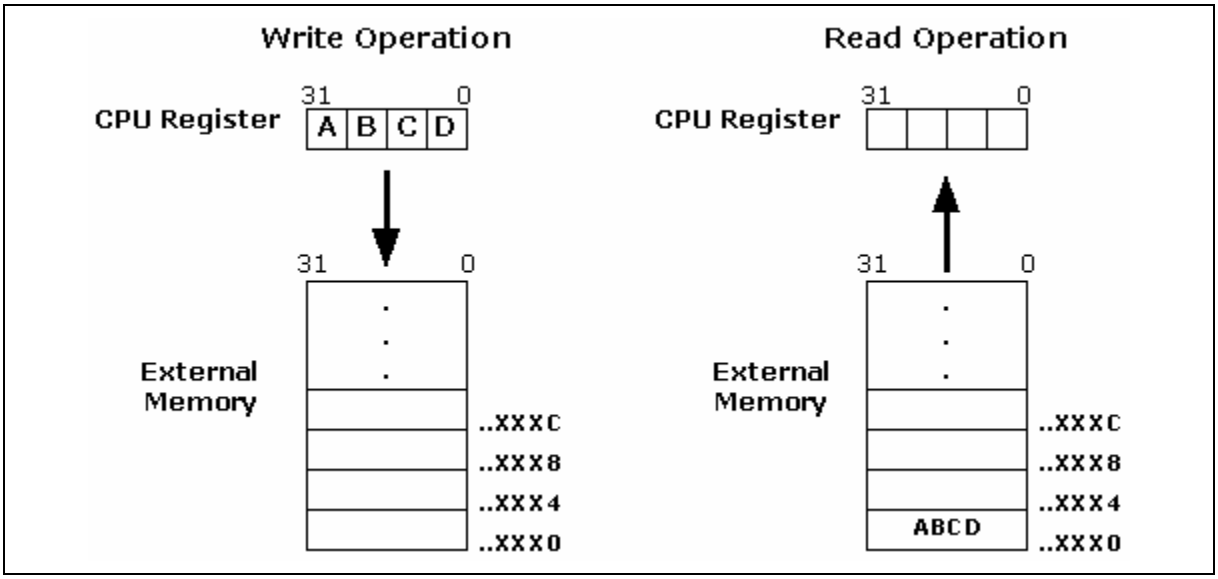


Figure 6.2.5 CPU registers Read/Write with external memory

Table 6.2.3 and Table 6.2.4

Using big-endian and word access, Program/Data path between register and external memory

WA = Address whose LSB is 0,4,8,C X = Don't care

nWBE [1-0] / SDQM [1-0] = A means active and U means inactive

Table 6.2.3 Word access write operation with Big Endian

ACCESS OPERATION	WRITE OPERATION (CPU REGISTER ➔ EXTERNAL MEMORY)					
XD WIDTH	**HALF WORD**		**BYTE**			
Bit Number **CPU Reg Data**	31 0 ABCD		31 0 ABCD			
SA	WA		WA			
Bit Number **SD**	31 0 AB CD		31 0 A B C D			
Bit Number **ED**	15 0 AB	15 0 CD	7 0 A	7 0 B	7 0 C	7 0 D
XA	WA	WA+2	WA	WA+1	WA+2	WA+3
nWBE [1-0] / **SDQM [1-0]**	AA	AA	XA	XA	XA	XA
Bit Number **XD**	15 0 AB	15 0 CD	7 0 A	7 0 B	7 0 C	7 0 D
Bit Number **Ext. Mem Data**	15 0 AB	15 0 CD	7 0 A	7 0 B	7 0 C	7 0 D
Timing Sequence	1st write	2nd write	1st write	2nd write	3rd write	4th write

Table 6.2.4 Word access read operation with Big Endian

ACCESS OPERATION	READ OPERATION (CPU REGISTER ← EXTERNAL MEMORY)					
XD WIDTH	**HALF WORD**		**BYTE**			
Bit Number **CPU Reg Data**	31 0 CDAB		31 0 DCBA			
SA	WA		WA			
Bit Number **SD**	31 0 CD AB		31 0 D C B A			
Bit Number **ED**	31 0 CD XX	31 0 CD AB	31 0 D X X X	31 0 D C X X	31 0 D C B X	31 0 D C B A
XA	WA	WA+2	WA	WA+1	WA+2	WA+3
SDQM [1-0]	AA	AA	XA	XA	XA	XA
Bit Number **XD**	15 0 CD	15 0 AB	7 0 D	7 0 C	7 0 B	7 0 A
Bit Number **Ext. Mem Data**	15 0 CD	15 0 AB	7 0 D	7 0 C	7 0 B	7 0 A
Timing Sequence	1st read	2nd read	1st read	2nd read	3rd read	4th read

Table 6.2.5 and Table 6.2.6

Using big-endian and half-word access, Program/Data path between register and external memory.

HA = Address whose LSB is 0,2,4,6,8,A,C,E X = Don't care

nWBE [1-0] / SDQM [1-0] = A means active and U means inactive

Table 6.2.5 Half-word access write operation with Big Endian

ACCESS OPERATION	WRITE OPERATION (CPU REGISTER → EXTERNAL MEMORY)		
XD WIDTH	**HALF WORD**	**BYTE**	
Bit Number **CPU Reg Data**	31 0 ABCD	31 0 ABCD	
SA	HA	HA	
Bit Number **SD**	31 0 CD CD	31 0 CD CD	31 0 CD CD
Bit Number **ED**	31 0 CD CD	7 0 C	7 0 D
XA	HA	HA	HA+1
nWBE [1-0] / **SDQM [1-0]**	AA	XA	XA
Bit Number **XD**	15 0 CD	7 0 C	7 0 D
Bit Number **Ext. Mem Data**	15 0 CD	7 0 C	7 0 D
Timing Sequence		1st write	2nd write

Table 6.2.6 Half-word access read operation with Big Endian

ACCESS OPERATION	READ OPERATION (CPU REGISTER ← EXTERNAL MEMORY)		
XD WIDTH	**HALF WORD**	**BYTE**	
Bit Number **CPU Reg Data**	15 0 CD	15 0 DC	
SA	HA	HA	
Bit Number **SD**	15 0 CD	15 0 DC	
Bit Number **ED**	15 0 CD	15 0 DX	15 0 DC
XA	HA	HA	HA+1
SDQM [1-0]	AA	XA	XA
Bit Number **XD**	15 0 CD	7 0 D	7 0 C
Bit Number **Ext. Mem Data**	15 0 CD	7 0 D	7 0 C
Timing Sequence		1st read	2nd read

Table 6.2.7 and Table 6.2.8

Using big-endian and byte access, Program/Data path between register and external memory.

BA = Address whose LSB is 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F

BAL = Address whose LSB is 0,2,4,6,8,A,C,E BAU = Address whose LSB is 1,3,5,7,9,B,D,F

Table 6.2.7 Byte access write operation with Big Endian

ACCESS OPERATION	WRITE OPERATION (CPU REGISTER EXTERNAL MEMORY)		
XD WIDTH	**HALF WORD**		**BYTE**
Bit Number **CPU Reg Data**	31 0 ABCD		31 0 ABCD
SA	BAL	BAU	BA
Bit Number **SD**	31 0 D D D D	31 0 D D D D	31 0 D D D D
Bit Number **ED**	15 8 D	7 0 D	7 0 D
XA	BAL	BAL	BA
nWBE [1-0] / **SDQM [1-0]**	AU	UA	XA
Bit Number **XD**	15 0 D X	15 0 X D	7 0 D
Bit Number **Ext. Mem Data**	15 8 D	7 0 D	7 0 D
Timing Sequence			

Table 6.2.8 Byte access read operation with Big Endian

ACCESS OPERATION	READ OPERATION (CPU REGISTER ← EXTERNAL MEMORY)		
XD WIDTH	**HALF WORD**		**BYTE**
Bit Number **CPU Reg Data**	7 0 C	7 0 D	7 0 D
SA	BAL	BAU	BA
Bit Number **SD**	7 0 C	7 0 D	7 0 D
Bit Number **ED**	7 0 C	15 8 D	7 0 D
XA	BAL	BAL	BA
SDQM [1-0]	AU	UA	XA
Bit Number **XD**	15 0 CD	15 0 CD	7 0 D
Bit Number **Ext. Mem Data**	15 0 CD		7 0 D
Timing Sequence			

Table 6.2.9 and Table 6.2.10

Using little-endian and word access, Program/Data path between register and external memory

WA = Address whose LSB is 0,4,8,C X = Don't care

nWBE [1-0] / SDQM [1-0] = A means active and U means inactive

Table 6.2.9 Word access write operation with little Endian

ACCESS OPERATION	WRITE OPERATION (CPU REGISTER → EXTERNAL MEMORY)					
XD WIDTH	**HALF WORD**		**BYTE**			
Bit Number **CPU Reg Data**	31 0 ABCD		31 0 ABCD			
SA	WA		WA			
Bit Number **SD**	31 0 AB CD		31 0 A B C D			
Bit Number **ED**	15 0 CD	15 0 AB	7 0 D	7 0 C	7 0 B	7 0 A
XA	WA	WA+2	WA	WA+1	WA+2	WA+3
nWBE [1-0] / **SDQM [1-0]**	AA	AA	XA	XA	XA	XA
Bit Number **XD**	15 0 CD	15 0 AB	7 0 D	7 0 C	7 0 B	7 0 A
Bit Number **Ext. Mem Data**	15 0 CD	15 0 AB	7 0 D	7 0 C	7 0 B	7 0 A
Timing Sequence	1st write	2nd write	1st write	2nd write	3rd write	4th write

Table 6.2.10 Word access read operation with Little Endian

ACCESS OPERATION	READ OPERATION (CPU REGISTER ← EXTERNAL MEMORY)					
XD Width	**Half Word**		**Byte**			
Bit Number **CPU Reg Data**	31 0 ABCD		31 0 ABCD			
SA	WA		WA			
Bit Number **SD**	31 0 AB CD		31 0 A B C D			
Bit Number **ED**	31 0 XX CD	31 0 AB CD	31 0 X X X D	31 0 X X C D	31 0 X B C D	31 0 A B C D
XA	WA	WA+2	WA	WA+1	WA+2	WA+3
SDQM [1-0]	AA	AA	XA	XA	XA	XA
Bit Number **XD**	15 0 CD	15 0 AB	7 0 D	7 0 C	7 0 B	7 0 A
Bit Number **Ext. Mem Data**	15 0 CD	15 0 AB	7 0 D	7 0 C	7 0 B	7 0 A
Timing Sequence	1st write	2nd write	1st write	2nd write	3rd write	4th write

Table 6.2.11 and Table 6.2.12

Using little-endian and half-word access, Program/Data path between register and external memory.

HA = Address whose LSB is 0,2,4,6,8,A,C,E X = Don't care

nWBE [1-0] / SDQM [1-0] = A means active and U means inactive

Table 6.2.11 Half-word access write operation with little Endian

ACCESS OPERATION	WRITE OPERATION (CPU REGISTER → EXTERNAL MEMORY)		
XD Width	**Half Word**	**Byte**	
Bit Number **CPU Reg Data**	31 0 ABCD	31 0 ABCD	
SA	HA	HA	
Bit Number **SD**	31 0 CD CD	31 0 CD CD	31 0 CD CD
Bit Number **ED**	31 0 CD CD	7 0 D	7 0 C
XA	HA	HA	HA+1
nWBE [1-0] / **SDQM [1-0]**	AA	XA	XA
Bit Number **XD**	15 0 CD	7 0 D	7 0 C
Bit Number **Ext. Mem Data**	15 0 CD	7 0 D	7 0 C
Timing Sequence		1st write	2nd write

Table 6.2.12 Half-word access read operation with Little Endian

ACCESS OPERATION	READ OPERATION (CPU REGISTER ← EXTERNAL MEMORY)		
XD Width	**Half Word**	**Byte**	
Bit Number **CPU Reg Data**	15 0 CD	15 0 CD	
SA	HA	HA	
Bit Number **SD**	15 0 CD	15 0 CD	
Bit Number **ED**	15 0 CD	15 0 XD	15 0 CD
XA	HA	HA	HA+1
SDQM [1-0]	AA	XA	XA
Bit Number **XD**	15 0 CD	7 0 D	7 0 C
Bit Number **Ext. Mem Data**	15 0 CD	7 0 D	7 0 C
Timing Sequence		1st read	2nd read

Table 6.2.13 and Table 6.2.14

Using little-endian and byte access, Program/Data path between register and external memory.
BA = Address whose LSB is 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F
BAL = Address whose LSB is 0,2,4,6,8,A,C,E BAU = Address whose LSB is 1,3,5,7,9,B,D,F

Table 6.2.13 Byte access write operation with little Endian

ACCESS OPERATION	WRITE OPERATION (CPU REGISTER → EXTERNAL MEMORY)		
XD Width	**Half Word**		**Byte**
Bit Number **CPU Reg Data**	31 0 ABCD		31 0 ABCD
SA	BAL	BAU	BA
Bit Number **SD**	31 0 D D D D	31 0 D D D D	31 0 D D D D
Bit Number **ED**	7 0 D	15 8 D	7 0 D
XA	BAL	BAL	BA
nWBE [1-0] / **SDQM [1-0]**	UA	AU	XA
Bit Number **XD**	15 0 X D	15 0 D X	7 0 D
Bit Number **Ext. Mem Data**	7 0 D	15 8 D	7 0 D
Timing Sequence			

Table 6.2.14 Byte access read operation with Little Endian

ACCESS OPERATION	READ OPERATION (CPU REGISTER ← EXTERNAL MEMORY)		
XD Width	**Half Word**		**Byte**
Bit Number **CPU Reg Data**	7 0 D	7 0 C	7 0 D
SA	BAL	BAU	BA
Bit Number **SD**	7 0 D	7 0 C	7 0 D
Bit Number **ED**	7 0 D	7 0 C	7 0 D
XA	BAL	BAL	BA
SDQM [1-0]	UA	AU	XA
Bit Number **XD**	15 0 CD	15 0 CD	7 0 D
Bit Number **Ext. Mem Data**	15 0 CD		7 0 D
Timing Sequence			

6.2.5 Bus Arbitration

The W90N745's internal function blocks or external devices can request mastership of the system bus and then hold the system bus in order to perform data transfers. Because the design of W90N745 bus allows only one bus master at a time, a bus controller is required to arbitrate when two or more internal units or external devices simultaneously request bus mastership. When bus mastership is granted to an internal function block or an external device, other pending requests are not acknowledged until the previous bus master has released the bus.

W90N745 supports two priority modes, the **Fixed Priority Mode** and the **Rotate Priority Mode**, depends on the ARBCON register **PRTMOD** bit setting.

6.2.5.1. Fixed Priority Mode

In **Fixed Priority Mode** (**PRTMOD**=0, default value), to facilitate bus arbitration, priorities are assigned to each internal W90N745 function block. The bus controller arbitration requests for the bus mastership according to these fixed priorities. In the event of contention, mastership is granted to the function block with the highest assigned priority. These priorities are listed in Table 6.2.15.

W90N745 allows raising ARM Core priority to second if an unmasked interrupt occurred. If **IPEN** bit, Bit 1 of the **Arbitration Control Register (ARBCON)**, is set to "0", the priority of ARM Core is fixed to lowest. If **IPEN** bit is set to "1" and if no unmasked interrupt request, then the ARM Core's priority is still lowest and the **IPACT**=0, Bit 2 of the **Arbitration Control Register (ARBCON)** ； If there is an unmasked interrupt request, then the ARM Core's priority is raised to first and **IPACT**=1.

If **IPEN** is set, an interrupt handler will normally clear **IPACT** at the end of the interrupt routine to allow an alternate bus master to regain the bus; however, if **IPEN** is cleared, no additional action need be taken in the interrupt handler. The **IPACT** bit can be read and written. Writing with "0", the **IPACT** bit is cleared, but it will be no effect as writing with "1".

Table 6.2.15 Bus Priorities for Arbitration in Fixed Priority Mode

BUS	FUNCTION BLOCK	
PRIORITY	IPACT = 0	IPEN = 1 AND IPACT = 1
1 (Highest)	Audio Controller (AC97 & I²S)	ARM Core
2	General DMA0	Audio Controller (AC97 & I²S)
3	General DMA1	General DMA0
4	EMC DMA	General DMA1
5	USB Host	EMC DMA
6	USB Device	USB Host
7(Lowest)	ARM Core	USB Device

6.2.5.2. Rotate Priority Mode

In **Rotate Priority Mode** (**PRTMOD**=1), the **IPEN** and **IPACT** bits have no function (i.e. can be ignored). W90N745 uses a round robin arbitration scheme ensures that all bus masters have equal chance to gain the bus and that a retracted master does not lock up the bus.

6.2.6 Power Management

W90N745 provide three power management scenarios to reduce power consumption. The peripherals' clocks can be enabled / disabled individually by controlling the co-responding bit in **CLKSEL** control register. Software can turn-off the unused modules' clocks to saving the unnecessary power consumption. It also provides **idle** and **power-down** modes to reduce power consumption.

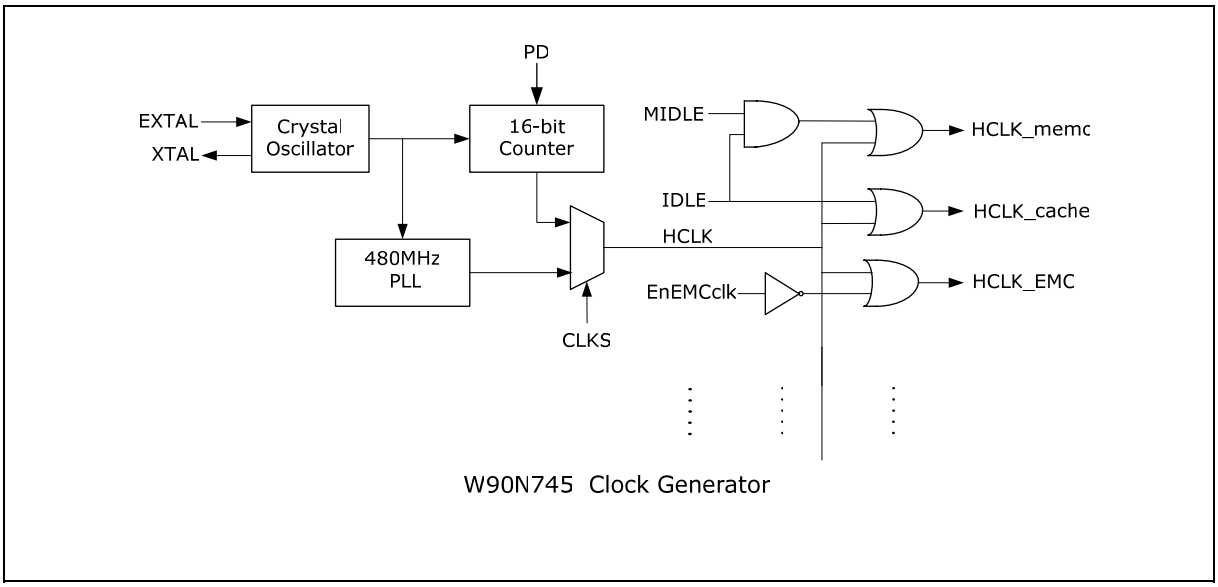


Figure 6.2.6 W90N745 system clock generation diagram

IDLE MODE

If the IDLE bit in Power Management Control Register (PMCON) is set, the ARM CORE clock source will be halted, the ARM CORE will not go forward. The AHB or APB clocks still active except the clock to cache controller and ARM are stopped. W90N745 will exit idle state when nIRQ or nFIQ from any peripheral is revived; like keypad, timer overflow interrupts and so on. The memory controller can also be forced to enter idle state if both MIDLE and IDLE bits are set. Software must switch SDRAM into self-refresh mode before forcing memory to enter idle mode.

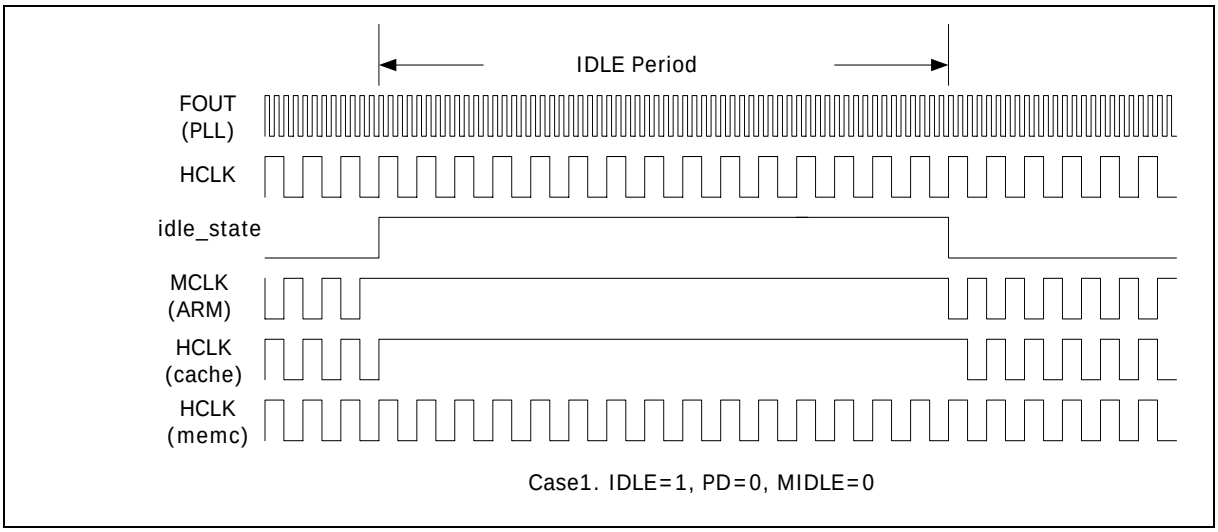


Figure 6.2.7 Clock management for system idle mode

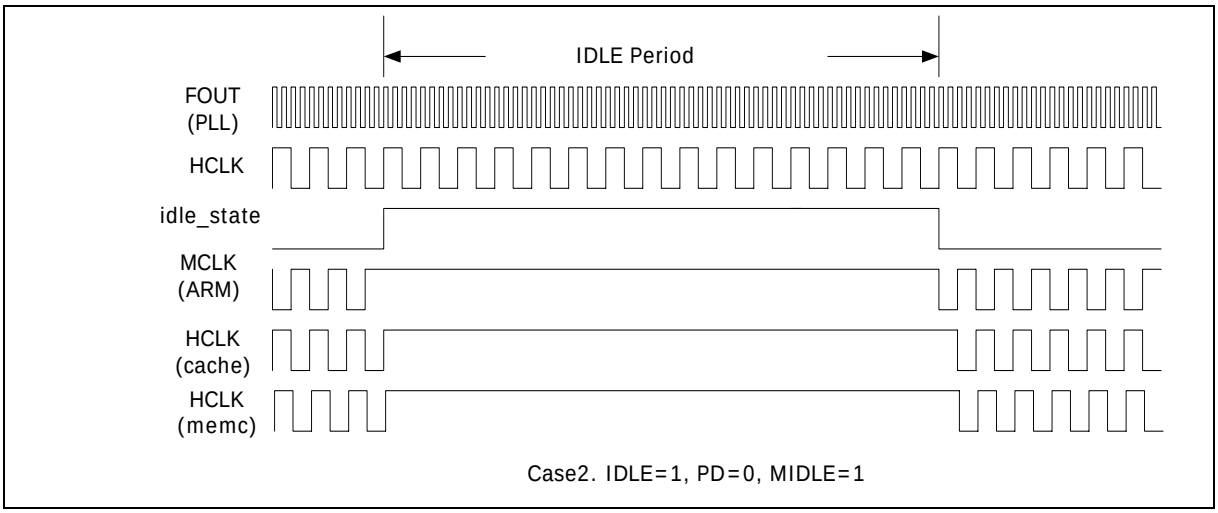


Figure 6.2.8 Clock management for system and memory idle mode

Power Down Mode

This mode provides the minimum power consumption. When the W90N745 system is not working or waiting an external event, software can write PD bit "1" to turn off all the clocks includes system crystal oscillator to let ARM CORE enter sleep mode. In this state, all peripherals are also in sleep mode since the clock source is stopped. W90N745 will exit power down state when nIRQ/nFIQ is detected. W90N745 provides external interrupt nIRQ[1:0], keypad, and USB device interfaces to wakeup the system clock.

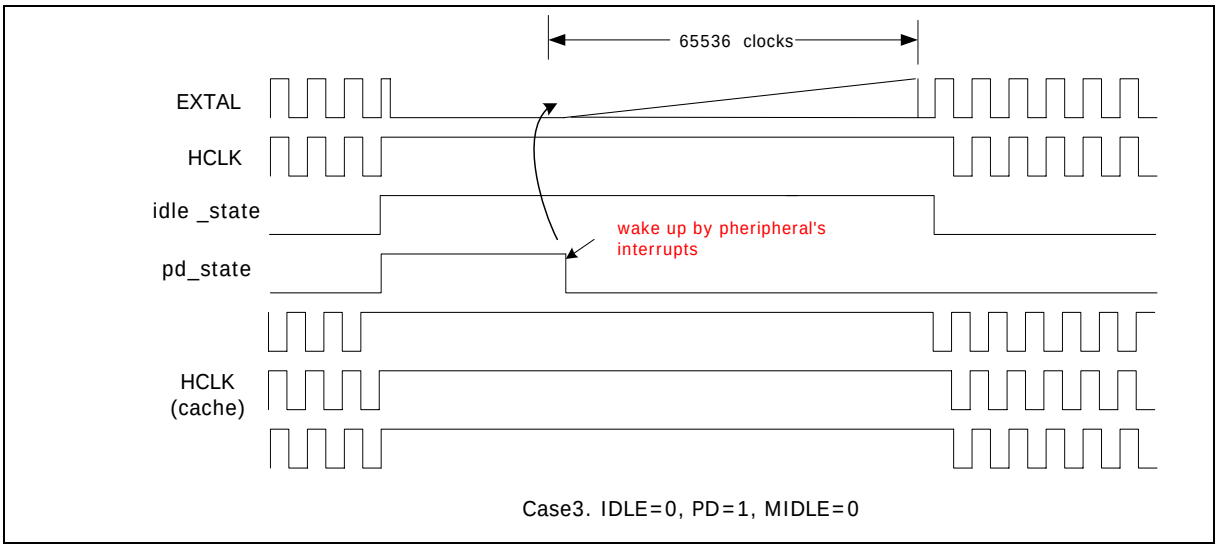


Figure 6.2.9 Clock management for system power down mode and wake up

6.2.7 Power-On Setting

After power on reset, there are eight Power-On setting pins to configure W90N745 system configuration.

POWER-ON SETTING	PIN
Internal System Clock Select	D15
Little/Big Endian Mode Select	D14
Boot ROM/FLASH Data Bus Width	D [13:12]
Default: Pull-Down in Normal Operation	D9
Default: Pull-Up in Normal Operation	D8

D15 pin：Internal System Clock Select

If pin D15 is pull-down, the external clock from EXTAL pin is served as internal system clock.

If pin D15 is pull-up, the PLL output clock is used as internal system clock.

D14 pin：Little/Big Endian Mode Select

If pin D14 is pull-down, the external memory format is Big Endian mode.

If pin D14 is pull-up, the external memory format is Little Endian mode.

D [13:12] ： Boot ROM/FLASH Data Bus Width

D [13:12]		BUS WIDTH
Pull-down	Pull-down	8-bit
Pull-down	Pull-up	16-bit
Pull-up	Pull-down	RESERVED
Pull-up	Pull-up	RESERVED

6.2.8 System Manager Control Registers Map

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
PDID	0xFFF0_0000	R	Product Identifier Register	0xX090_0745
ARBCON	0xFFF0_0004	R/W	Arbitration Control Register	0x0000_0000
PLLCON0	0xFFF0_0008	R/W	PLL Control Register 0	0x0000_2F01
CLKSEL	0xFFF0_000C	R/W	Clock Select Register	0x1FFF_3FX8
PLLCON1	0xFFF0_0010	R/W	PLL Control Register 1	0x0001_0000
I²SCKCON	0xFFF0_0014	R/W	Audio I²S Clock Control Register	0x0000_0000
IRQWAKECON	0xFFF0_0020	R/W	IRQ Wakeup Control register	0x0000_0000
IRQWAKEFLAG	0xFFFF_0024	R/W	IRQ wakeup Flag Register	0x0000_0000
PMCON	0xFFF0_0028	R/W	Power Manager Control Register	0x0000_0000
USBTxrCON	0xFFF0_0030	R/W	USB Transceiver Control Register	0x0000_0000

Product Identifier Register （PDID）

This register is read only and lets software can use it to recognize certain characteristics of the chip ID and the version number.

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
PDID	0xFFF0_0000	R	Product Identifier Register	0xX090_0745

31	30	29	28	27	26	25	24
PACKAGE		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	DESCRIPTION	
[31:30]	PACKAGE	**Package Type Select** These two bits are power-on setting latched from pin D[9:8] **Package [31:30]** = 0 1 → **Package Type** = 128-pin Package
[29:24]	VERSION	**Version of chip**
[23:0]	CHIPID	**The chip identifier** of W90N745 is 0x090.0745

Arbitration Control Register (ARBCON)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
ARBCON	0xFFF0_0004	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					IPACT	IPEN	PRTMOD

BITS	DESCRIPTION	
[31:3]	RESERVED	-
[2]	IPACT	**Interrupt priority active.** When **IPEN=**"1", this bit will be set when the ARM core has an unmasked interrupt request. This bit is available only when the **PRTMOD=**0.
[1]	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.
[0]	PRTMOD	**Priority mode select** 0 = Fixed Priority Mode (default) 1 = Rotate Priority Mode

PLL Control Register0 （PLLCON0）

W90N745 provides two clock generation options – crystal and oscillator. The external clock via **EXTAL(15**M) Minput pin as the reference clock input of **PLL** module. The external clock can bypass the **PLL** and be used to the internal system clock by pull-down the data D15 pin. Using **PLL**'s output clock for the internal system clock, D15 pin must be pull-up.

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
PLLCON	0xFFF0_0008	R/W	PLL Control Register	0x0000_2F01

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	DESCRIPTION	
[31:17]	RESERVED	-
[16]	PWDEN	**Power down mode enable** 0 = PLL is in normal mode (default) 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 [6:5] / 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.

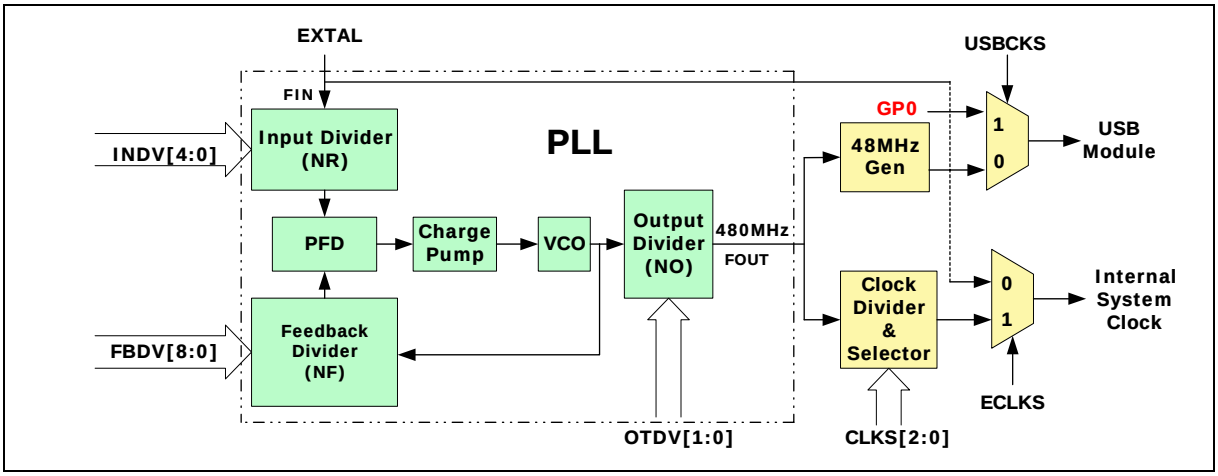


Figure 6.2.10 System PLL block diagram

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)

Clock Select Register (CLKSEL)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
CLKSEL	0xFFF0_000C	R/W	Clock Select Register	0x1FFF_7FX8

31	30	29	28	27	26	25	24
RESERVED			PS2	KPI	RESERVED		SSP
23	22	21	20	19	18	17	16
UART3	UART2	UART1	I^2C1	I^2C0	RESERVED	PWM	AC97
15	14	13	12	11	10	9	8
USBCKS	USBD	GDMA	RESERVED		EMC	RESERVED	WDT
7	6	5	4	3	2	1	0
USBH	TIMER	UART	ECLKS	CLKS			RESET

BITS	DESCRIPTION	
[31:29]	RESERVED	-
[28]	PS2	**PS2 controller clock enable bit** 0 = Disable PS2 controller clock 1 = Enable PS2 controller clock
[27]	KPI	**Keypad controller clock enable bit** 0 = Disable keypad controller clock 1 = Enable keypad controller clock
[26]	RESERVED	-
[25]	RESERVED	-
[24]	USI	**USI controller clock enable bit** 0 = Disable USI controller clock 1 = Enable USI controller clock
[23]	UART3	**UART3 controller clock enable bit** 0 = Disable UART3 controller clock 1 = Enable UART3 controller clock
[22]	UART2	**UART2 controller clock enable bit** 0 = Disable UART2 controller clock 1 = Enable UART2 controller clock
[21]	UART1	**UART1 controller clock enable bit** 0 = Disable UART1 controller clock 1 = Enable UART1 controller clock

Continued.

BITS	DESCRIPTION	
[20]	I^2C1	**I^2C1 controller clock enable bit** 0 = Disable I^2C1 controller clock 1 = Enable I^2C1 controller clock
[19]	I^2C0	**I^2C0 controller clock enable bit** 0 = Disable I^2C0 controller clock 1 = Enable I^2C0 controller clock
[18]	RESERVED	-
[17]	PWM	**PWM controller clock enable bit** **0 = Disable PWM controller clock** **1 = Enable PWM controller clock**
[16]	AC97	**Audio Controller clock enable bit** **0 = Disable AC97 controller clock** **1 = Enable AC97 controller clock**
[15]	USBCKS	**USB host/device 48MHz clock source Select bit** **0 = USB clock 48MHz input from internal PLL (480MHz/10)** **1 = USB clock 48MHz input from external GPIO0 pin, this pin direction must set to input.**
[14]	USBD	**USB device clock enable bit** **0 = Disable USB device controller clock** **1 = Enable USB device controller clock**
[13]	GDMA	**GDMA controller clock enable bit** **0 = Disable GDMA clock** **1 = Enable GDMA clock**
[12]	RESERVED	-
[11]	RESERVED	-
[10]	EMC	**EMC controller clock enable bit** **0 = Disable EMC controller clock** **1 = Enable EMC controller clock**
[9]	RESERVED	-
[8]	WDT	**WDT clock enable bit** **0 = Disable WDT counting clock** **1 = Enable WDT counting clock**

Continued.

<table>
<tr><th>BITS</th><th colspan="2">DESCRIPTION</th></tr>
<tr><td>[7]</td><td>USBH</td><td>USB host clock enable bit
0 = Disable USB host controller clock
1 = Enable USB host controller clock</td></tr>
<tr><td>[6]</td><td>TIMER</td><td>Timer clock enable bit
0 = Disable timer clock
1 = Enable timer clock</td></tr>
<tr><td>[5]</td><td>UART0</td><td>UART0 controller clock enable bit
0 = Disable UART0 controller clock
1 = Enable UART0 controller clock</td></tr>
<tr><td>[4]</td><td>ECLKS</td><td>External clock select
0 = External clock from EXTAL pin is used as system clock
1 = PLL output clock is used as system clock
After power on reset, the content of ECLKS is the Power-On Setting value. You can program this bit to change the system clock source.</td></tr>
<tr><td>[3:1]</td><td>CLKS</td><td>PLL output clock select
<table>
<tr><th colspan="3">CLKS [3:1]</th><th>System clock</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>58.594 KHz*</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>24 MHz</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>48 MHz</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>60 MHz</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>80 MHz</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>RESERVED</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>RESERVED</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>RESERVED</td></tr>
</table>
Note:
1. This values are based on PLL output(FOUT) is 480MHz.
2. When 24Mhz ~ 80MHz is selected, the ECLKS bit must be set to 1.
3. About 58.594KHz setting, two steps are needed. First, clear ECLKS bit, and then clear CLKS.</td></tr>
<tr><td>[0]</td><td>RESET</td><td>Software Reset bit
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.</td></tr>
</table>

PLL Control Register 1（PLLCON1）

W90N745 provides extra PLL to provide 12.288/16.934 MHz clock source to Audio Controller. It uses the same 15MHz crystal clock input source with system PLL mentioned above.

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
PLLCON1	0xFFF0_0010	R/W	PLL Control Register 1	0x0001_0000

31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							PWDEN1
15	14	13	12	11	10	9	8
FBDV1							
7	6	5	4	3	2	1	0
FBDV1	OTDV1		INDV1				

BITS	DESCRIPTION	
[31:17]	RESERVED	-
[16]	PWDEN1	**PLL1 power down enable** 0 = PLL1 is in normal mode 1 = PLL1 is in power down mode (default)
[15:7]	FBDV1	**PLL1 VCO output clock feedback divider** Feedback Divider divides the output clock from VCO of PLL1.
[6:5]	OTDV1	**PLL1 output clock divider** OTDV1 [6:5] / Divided by: 0 0 : 1 0 1 : 2 1 0 : 2 1 1 : 4
[4:0]	INDV1	**PLL1 input clock divider** Input divider divides the input reference clock into the PLL1.

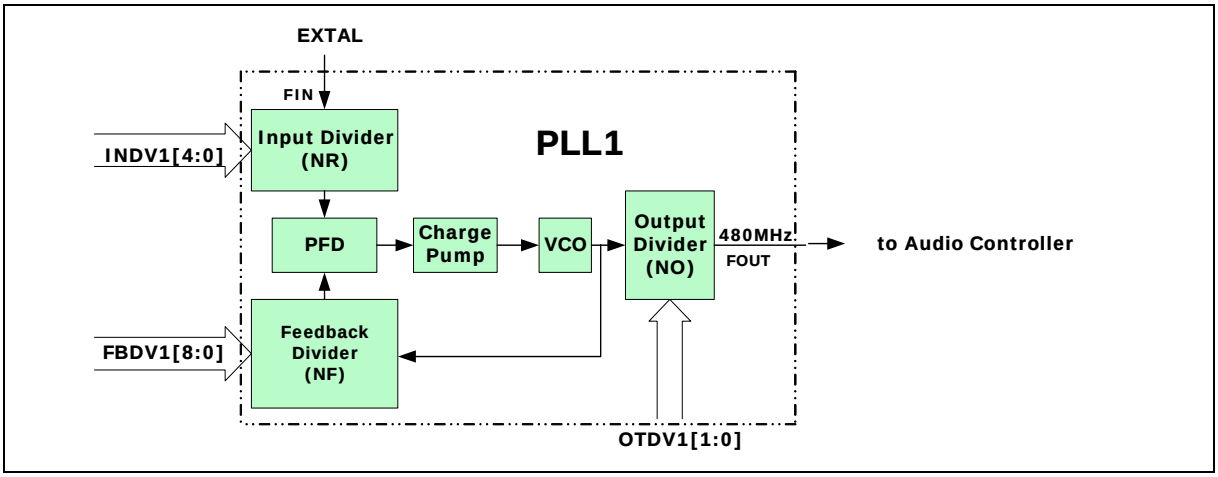


Figure 6.2.11 Audio PLL block diagram

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 = INDV1 + 2)

NF : Feedback divider value (NF = FBDV1 + 2)

NO : Output divider value (NO = OTDV1)

I²S Clock Control Register (I²SCKCON)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
I²SCKCON	0xFFF0_0014	R/W	I²S PLL clock 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							I²SPLLEN
7	6	5	4	3	2	1	0
PRESCALE							

BITS	DESCRIPTION	
[31:9]	RESERVED	-
[8]	I²SPLLEN	**I²S PLL clock source enable** Set this bit will enable PLL1 clock output to audio I²S clock input. 1 = Enable PLL1 clock source for audio I²S 0 = Disable PLL1 clock source for audio I²S
[7:0]	PRESCALE	The PLL1 is used by I²S, if in use, software can using this prescaler to generate an appropriate clock nearly 12.288M or 16.934M. The clock is generated as below, and if PRESCALE =0, the PLL_AUDIO is the same frequency as FOUT **"PLL_AUDIO = PLL_FOUT/(PRESCALE +1)"**

IRQ Wakeup Control Register (IRQWAKECON)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
IRQWAKECON	0xFFF0_0020	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							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED		**IRQWAKEUPPOL**		RESERVED		**IRQWAKEUPEN**	

BITS	DESCRIPTION	
[31:6]	RESERVED	
[5]	IRQWAKEUPPOL[1]	**nIRQ1 wake up polarity** 1 = nIRQ1 is high level wake up 0 = nIRQ1 is low level wake up
[4]	IRQWAKEUPPOL[0]	**nIRQ0 wake up polarity** 1 = nIRQ0 is high level wake up 0 = nIRQ0 is low level wake up
[3:2]	RESERVED	
[1]	IRQWAKEUPEN[1]	**nIRQ1 wake up enable bit** 1 = nIRQ1 wake up enable 0 = nIRQ1 wake up disable
[0]	IRQWAKEUPEN[0]	**nIRQ0 wake up enable bit** 1 = nIRQ0 wake up enable 0 = nIRQ0 wake up disable

IRQ Wakeup Flag Register (IRQWAKEFLAG)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
IRQWAKEFLAG	0xFFF0_0024	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
RESERVED						IRQWAKEFLAG	

This register is used to record the wakeup event, after clock recovery, software should check these flags to identify which nIRQ is used to wakeup the system. And clear the flags in IRQ interrupt sevice routine.

BITS	DESCRIPTION	
[31:2]	RESERVED	-
[1]	IRQWAKEFLAG[1]	**nIRQ1 wake up flag** 1 = chip is waked up by nIRQ1 0 = no active
[0]	IRQWAKEFLAG[0]	**nIRQ0 wake up flag** 1 = chip is waked up by nIRQ0 0 = no active

Power Management Control Register (PMCON)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
PMCON	0xFFF0_0028	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					MIDLE	PD	IDLE

BITS	DESCRIPTION	
[31:3]	RESERVED	
[2]	MIDLE	**Memory controller IDLE enable** Setting both MIDLE and IDLE bits HIGH will let memory controller enter IDLE mode, the clock source of memory controller will be halted while ARM CORE enter IDLE mode. 1=memory controller will be forced into IDLE mode, (clock of memory controller will be halted), when IDLE bit is set. 0 = memory controller still active when IDLE bit is set. **NOTE:** Software must let SDRAM enter self-refresh mode before enable this function because SDRAM MCLK will be stopped.
[1]	PD	**Power down enable** Setting this bit HIGH will let W90N745 enter power saving mode. The clock source 15M crystal oscillator and PLLs are stopped to generate clock. User can use nIRQ[3:0], keypad and external RESET to wakeup W90N745. 1 = Enable power down 0 = Disable
[0]	IDLE	**IDLE mode enable** Setting this bit HIGH will let ARM Core enter power saving mode. The peripherals can still keep working if the clock enable bit in CLKSEL is set. Any nIRQ or nFIQ to ARM Core will let ARM CORE to exit IDLE state. 1 = IDLE mode 0 = Disable

USB Transceiver Control Register (USBTXRCON)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
USBTXRCON	0xFFF0_0030	R/W	USB Transceiver 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							USBHnD

BITS	DESCRIPTION	
[31:1]	RESERVED	-
[0]	**USBHnD**	**USBHnD[0]: USB transceiver control** There are two USB1.1 built-in transceivers for data transmission. One is dedicated for USB host and the other is shared with USB device. Software can program this bit to switch the transceiver path. 1 = HOST 0 = Device

6.3 External Bus Interface

6.3.1 EBI Overview

W90N745 supports External Bus Interface (**EBI**), which controls the access to the external memory (ROM/FLASH, SDRAM) and External I/O devices. The **EBI** has seven chip selects to select one ROM/FLASH bank, two SDRAM banks, and four External I/O banks.The address bus is 21 bits. It supports 8-bit, 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
- External bus mastership

6.3.2 SDRAM Controller

The SDRAM controller module within W90N745 contains configuration registers、 timing control registers、common control register and other logic to provide 8、16 bits SDRAM interface with a single 8、16 bits SDRAM device or two 8-bit devices wired to give a 16-bit data path. The maximum size of each bank is 64M bytes, and maximum memory size can span up to 128MB.

The SDRAM controller has the following features：

- Supports up to 2 external SDRAM banks
- Maximum size of each bank is 64M bytes
- 8、16-bit data interface
- Programmable CAS Latency： 1、2 and 3
- Fixed Burst Length： 1
- Sequential burst type
- Auto Refresh Mode and Self Refresh Mode
- Adjustable Refresh Rate
- Power up sequence

6.3.2.1. SDRAM Components Supported

Table 6.3.1 SDRAM supported by W90N745

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

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.
A14 ~ A0 are the Address pins of the W90N745 EBI interface;
A14 and A13 are the Bank Select Signals of SDRAM.

SDRAM Data Bus Width: 16-bit

Total	Type	R x C	R/C	A14 (BS1)	A13 (BS0)	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
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

SDRAM Data Bus Width: 8-bit

Total	Type	R x C	R/C	A14 (BS1)	A13 (BS0)	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
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
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
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
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

6.3.2.2. SDRAM Power Up Sequence

The SDRAM must be initialized predefined manner after power on.W90N745 SDRAM Controller automatically executes the commands needed for initialion 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.

6.3.2.3. SDRAM Interface

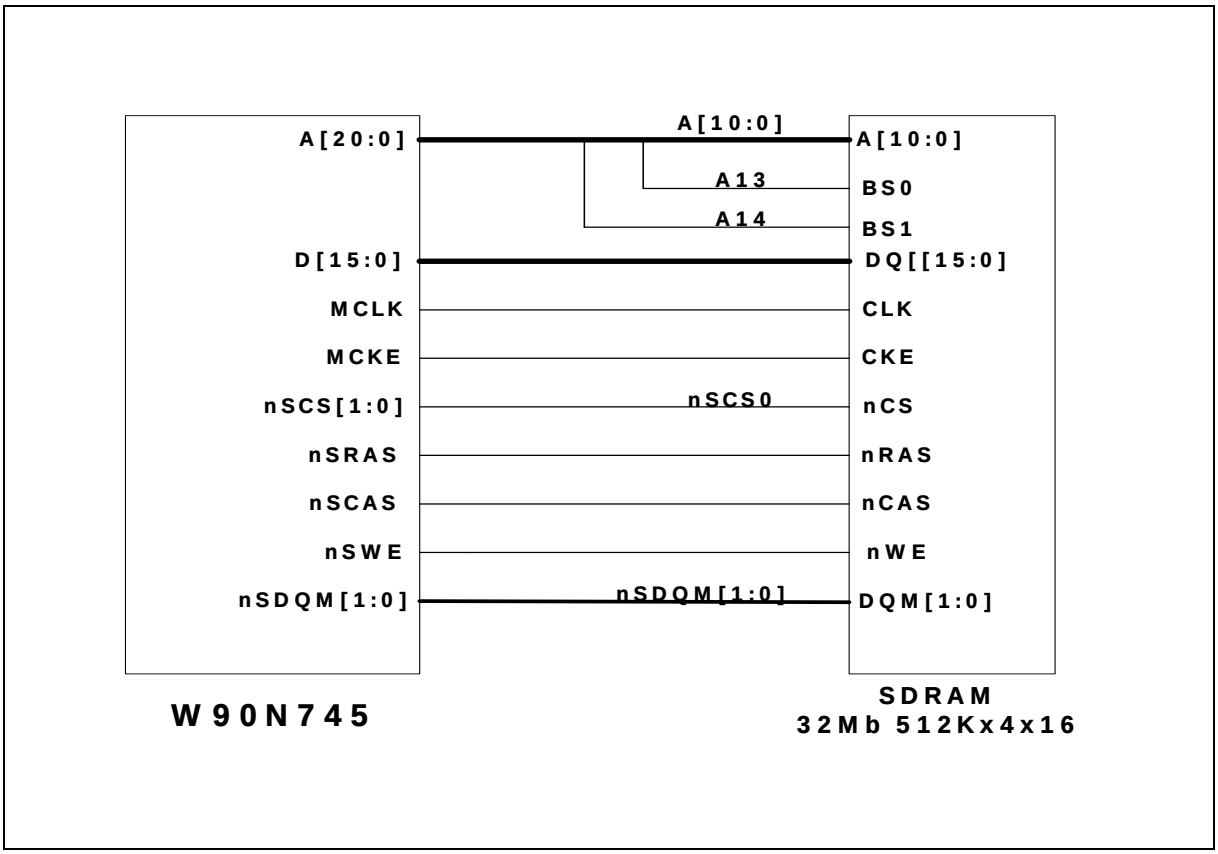


Figure 6.3.1 SDRAM Interface

6.3.3 EBI Control Registers Map

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
EBICON	0xFFF0_1000	R/W	EBI control register	0x0001_0000
ROMCON	0xFFF0_1004	R/W	ROM/FLASH control register	0x0000_0XFC
SDCONF0	0xFFF0_1008	R/W	SDRAM bank 0 configuration register	0x0000_0800
SDCONF1	0xFFF0_100C	R/W	SDRAM bank 1 configuration register	0x0000_0800
SDTIME0	0xFFF0_1010	R/W	SDRAM bank 0 timing control register	0x0000_0000
SDTIME1	0xFFF0_1014	R/W	SDRAM bank 1 timing control register	0x0000_0000
EXT0CON	0xFFF0_1018	R/W	External I/O 0 control register	0x0000_0000
EXT1CON	0xFFF0_101C	R/W	External I/O 1 control register	0x0000_0000
EXT2CON	0xFFF0_1020	R/W	External I/O 2 control register	0x0000_0000
EXT3CON	0xFFF0_1024	R/W	External I/O 3 control register	0x0000_0000
CKSKEW	0xFFF0_1F00	R/W	Clock skew control register (for testing)	0xXXXX_0038

EBI Control Register (EBICON)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
EBICON	0xFFF0_1000	R/W	EBI control register	0x0001_0000

31	30	29	28	27	26	25	24
RESERVED				EXBE3	EXBE2	EXBE1	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	DESCRIPTION	
[31:27]	RESERVED	
[27]	EXBE3	**External IO bank 3 byte enable** This function is used for some devices that with high and low bytes enable signals to control which byte will be write or mask data output when read. For this kind device, software can set this bit HIGH to implement this function. Detail pin interconnection is showed as Figure 6.3.8. 1 = nWBE[1:0] pin is byte enable signals, nWE will be used as write strobe signal to SRAM. 0 = nWBE[1:0] pin is byte write strobe signal.
[26]	EXBE2	**External IO bank 2 byte enable** The bit function description is the same as EXBE3 above. 1 = nWBE[1:0] pin is byte enable signals, nWE will be used as write strobe signal to SRAM. 0 = nWBE[1:0] pin is byte write strobe signal.
[25]	EXBE1	**External IO bank 1 byte enable** The bit function description is the same as EXBE3 above. 1 = nWBE[1:0] pin is byte enable signals, nWE will be used as write strobe signal to SRAM 0 = nWBE[1:0] pin is byte write strobe signal

Continued.

BITS	DESCRIPTION	
[24]	EXBE0	**External IO bank 0 byte enable** This bit function description is the same as EXBE3 above. 1 = nWBE[1:0] pin is byte enable signals, nWE will be used as write strobe signal to SRAM 0 = nWBE[1:0] pin is byte write strobe signal
[23:19]	RESERVED	-
[18]	REFEN	**Enable SDRAM refresh cycle for SDRAM bank0 & bank1** This bit set will start the auto-refresh cycle to SDRAM. The refresh rate is according to REFRAT bits. 1 = enable refresh function 0 = disable refresh function
[17]	REFMOD	**Refresh mode of SDRAM for SDRAM bank** Defines the refresh mode type of external SDRAM bank Software can write this bit "1" to force SDRAM enter self-refresh mode. **0** = Auto refresh mode **1** = Self refresh mode **NOTE:** If any read/write to SDRAM occurs then this bit will be cleared to "0" by hardware automatically and SDRAM will enters auto-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 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 The refresh period is calculated as $period = \frac{value}{fMCLK}$

Continued.

<table>
<tr><th>BITS</th><th colspan="2">DESCRIPTION</th></tr>
<tr><td>[2:1]</td><td>WAITVT</td><td>Valid time of nWAIT signal
W90N745 recognizes the nWAIT signal at the next “nth” MCLK rising edge after the nOE or nWBE active cycle. WAITVT bits determine the n.
<table>
<tr><th colspan="2">WAITVT [2:1]</th><th>nth MCLK</th></tr>
<tr><td>0</td><td>0</td><td>1</td></tr>
<tr><td>0</td><td>1</td><td>2</td></tr>
<tr><td>1</td><td>0</td><td>3</td></tr>
<tr><td>1</td><td>1</td><td>4</td></tr>
</table></td></tr>
<tr><td>[0]</td><td>LITTLE</td><td>Little Endian mode
After power on reset, the content of LITTLE is the Power-On Setting value from D14 pin. If pin D14 is pull-down, the external memory format is Big Endian mode. If pin D14 is pull-up, the external memory format is Little Endian mode. For more detail, refer to Power-On Setting of System Manager.
NOTE: This bit is read only.</td></tr>
</table>

ROM/Flash Control Register （ROMCON）

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
ROMCON	0xFFF0_1004	R/W	ROM/FLASH control register	0x0000_0XFC

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
RESERVED				tPA			
7	6	5	4	3	2	1	0
tACC				BTSIZE		PGMODE	

BITS	DESCRIPTION	
[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.
[18:16]	SIZE	**The size of ROM/FLASH memory** SIZE [10:8] / Byte: 0 0 0 — 256K 0 0 1 — 512K 0 1 0 — 1M 0 1 1 — 2M 1 0 0 — 4M 1 0 1 — RESERVED 1 1 0 — RESERVED 1 1 1 — RESERVED
[15:12]	RESERVED	-

Continued.

<table>
<tr><th>BITS</th><th colspan="2">DESCRIPTION</th></tr>
<tr><td>[11:8]</td><td>tPA</td><td>
Page mode access cycle time
<table>
<tr><th colspan="4">tPA[11:8]</th><th>MCLK</th><th colspan="4">tPA[11:8]</th><th>MCLK</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>10</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>1</td><td>2</td><td>1</td><td>0</td><td>0</td><td>1</td><td>12</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>0</td><td>3</td><td>1</td><td>0</td><td>1</td><td>0</td><td>14</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>1</td><td>4</td><td>1</td><td>0</td><td>1</td><td>1</td><td>16</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>0</td><td>5</td><td>1</td><td>1</td><td>0</td><td>0</td><td>18</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>1</td><td>6</td><td>1</td><td>1</td><td>0</td><td>1</td><td>20</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>0</td><td>7</td><td>1</td><td>1</td><td>1</td><td>0</td><td>22</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>1</td><td>8</td><td>1</td><td>1</td><td>1</td><td>1</td><td>24</td></tr>
</table>
</td></tr>
<tr><td>[7:4]</td><td>tACC</td><td>
Access cycle time
<table>
<tr><th colspan="4">tACC[11:8]</th><th>MCLK</th><th colspan="4">tACC[11:8]</th><th>MCLK</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>10</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>1</td><td>2</td><td>1</td><td>0</td><td>0</td><td>1</td><td>12</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>0</td><td>3</td><td>1</td><td>0</td><td>1</td><td>0</td><td>14</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>1</td><td>4</td><td>1</td><td>0</td><td>1</td><td>1</td><td>16</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>0</td><td>5</td><td>1</td><td>1</td><td>0</td><td>0</td><td>18</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>1</td><td>6</td><td>1</td><td>1</td><td>0</td><td>1</td><td>20</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>0</td><td>7</td><td>1</td><td>1</td><td>1</td><td>0</td><td>22</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>1</td><td>8</td><td>1</td><td>1</td><td>1</td><td>1</td><td>24</td></tr>
</table>
</td></tr>
<tr><td>[3:2]</td><td>BTSIZE</td><td>
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 the data bus signals D [13:12] power-on setting.
<table>
<tr><th colspan="2">BTSIZE [3:2]</th><th>Bus Width</th></tr>
<tr><td>0</td><td>0</td><td>8-bit</td></tr>
<tr><td>0</td><td>1</td><td>16-bit</td></tr>
<tr><td>1</td><td>0</td><td>RESERVED</td></tr>
<tr><td>1</td><td>1</td><td>RESERVED</td></tr>
</table>
<table>
<tr><th colspan="2">D [13:12]</th><th>Bus Width</th></tr>
<tr><td>Pull-down</td><td>Pull-down</td><td>8-bit</td></tr>
<tr><td>Pull-down</td><td>Pull-up</td><td>16-bit</td></tr>
<tr><td>Pull-up</td><td>Pull-down</td><td>RESERVED</td></tr>
<tr><td>Pull-up</td><td>Pull-up</td><td>RESERVED</td></tr>
</table>
</td></tr>
<tr><td>[1:0]</td><td>PGMODE</td><td>
Page mode configuration
<table>
<tr><th colspan="2">PGMODE [1:0]</th><th>Mode</th></tr>
<tr><td>0</td><td>0</td><td>Normal ROM</td></tr>
<tr><td>0</td><td>1</td><td>4 word page</td></tr>
<tr><td>1</td><td>0</td><td>8 word page</td></tr>
<tr><td>1</td><td>1</td><td>16 word page</td></tr>
</table>
</td></tr>
</table>

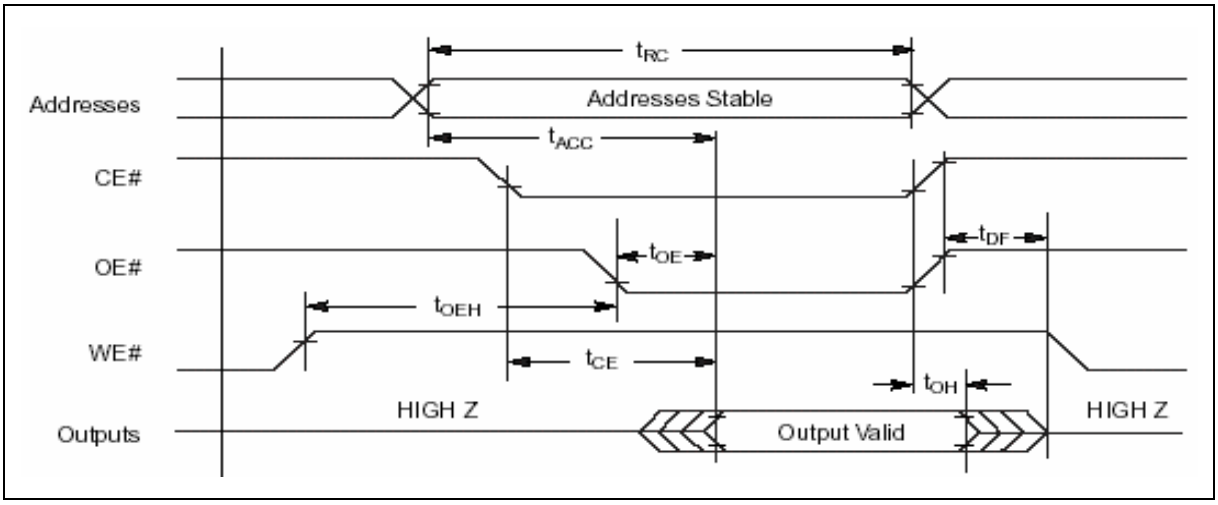


Figure 6.3.2 ROM/FLASH Read Operation Timing

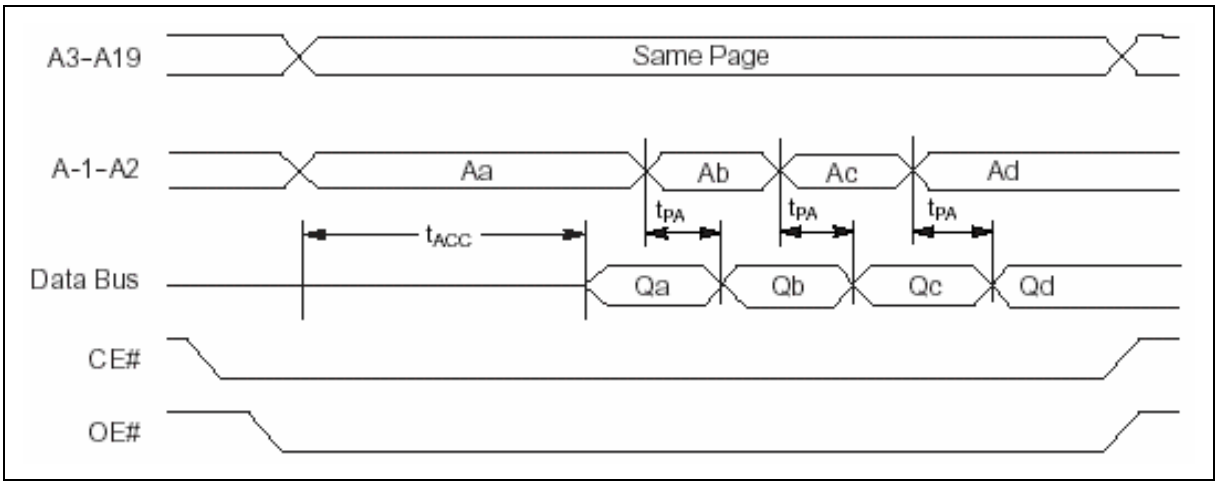


Figure 6.3.3 ROM/FLASH Page Read Operation Timing

Configuration Registers（SDCONF0/1）

The configuration registers enable software to set a number of operating parameters for the SDRAM controller. There are two configuration registers SDCONF0、SDCONF1 for SDRAM bank 0、bank 1 respectively. Each bank can have a different configuration.

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
SDCONF0	0xFFF0_1008	R/W	SDRAM bank 0 configuration register	0x0000_0800
SDCONF1	0xFFF0_100C	R/W	SDRAM bank 1 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		DESCRIPTION
[31:19]	BASADDR	**Base address pointer of SDRAM bank 0/1** The start address is calculated as SDRAM bank 0/1 base pointer << 18. The SDRAM base address pointer together with the "**SIZE**" bits constitutes the whole address range of each SDRAM bank.
[18:16]	RESERVED	-
[15]	MRSET	**SDRAM Mode register set command for SDRAM bank 0/1** This bit set will issue a mode register set command to SDRAM.
[14]	RESERVED	-
[13]	AUTOPR	**Auto pre-charge mode of SDRAM for SDRAM bank 0/1** Enable the auto pre-charge function of external SDRAM bank 0/1 0 = Auto pre-charge 1 = No auto pre-charge
[12:11]	LATENCY	**The CAS Latency of SDRAM bank 0/1** Defines the CAS latency of external SDRAM bank 0/1

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

Continued.

<table>
<tr><th>BITS</th><th colspan="2">DESCRIPTION</th></tr>
<tr><td>[10:8]</td><td>RESERVED</td><td>-</td></tr>
<tr><td>[7]</td><td>COMPBK</td><td>Number of component bank in SDRAM bank 0/1
Indicates the number of component bank (2 or 4 banks) in external SDRAM bank 0/1.
0 = 2 banks
1 = 4 banks</td></tr>
<tr><td>[6:5]</td><td>DBWD</td><td>Data bus width for SDRAM bank 0/1
Indicates the external data bus width connect with SDRAM bank 0/1
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>REVERSED</td></tr>
</table></td></tr>
<tr><td>[4:3]</td><td>COLUMN</td><td>Number of column address bits in SDRAM bank 0/1
Indicates the number of column address bits in external SDRAM bank 0/1.
<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>REVERSED</td></tr>
</table></td></tr>
<tr><td>[2:0]</td><td>SIZE</td><td>Size of SDRAM bank 0/1
Indicates the memory size of external SDRAM bank 0/1
<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>REVERSED</td></tr>
</table></td></tr>
</table>

Timing Control Registers （SDTIME0/1）

W90N745 offers the flexible timing control registers to control the generation and processing of the control signals and can achieve you use different speed of SDRAM

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
SDTIME0	0xFFF0_1010	R/W	SDRAM bank 0 timing control register	0x0000_0000
SDTIME1	0xFFF0_1014	R/W	SDRAM bank 1 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	DESCRIPTION	
[31:11]	RESERVED	-
[10:8]	tRCD	**SDRAM bank 0/1, /RAS to /CAS delay** 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
[7:6]	tRDL	**SDRAM bank 0/1, Last data in to pre-charge command** tRDL [7:6] / MCLK 0 0 : 1 0 1 : 2 1 0 : 3 1 1 : 4

Continued.

<table>
<tr><th>BITS</th><th colspan="2">DESCRIPTION</th></tr>
<tr><td>[5:3]</td><td>tRP</td><td>SDRAM bank 0/1, 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 0/1, 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>

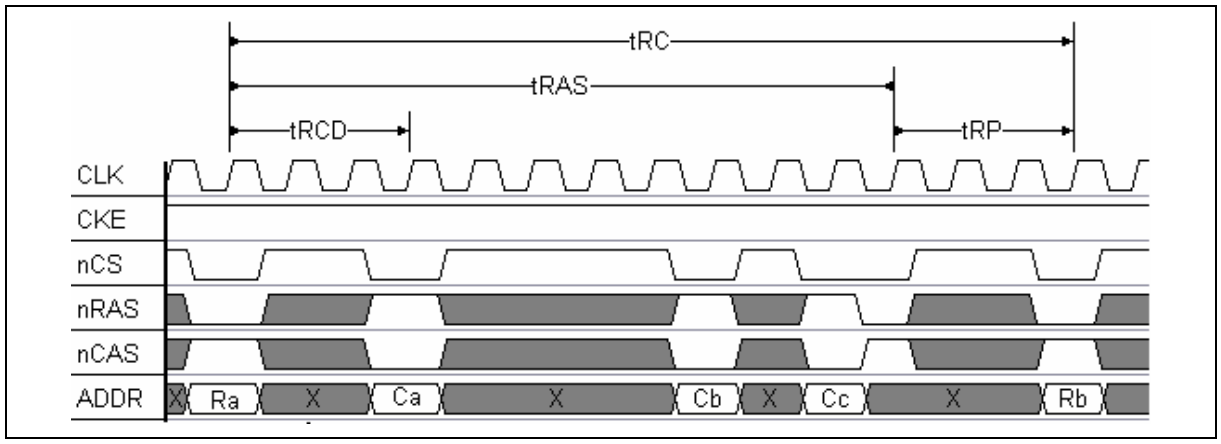


Figure 6.3.4 Access timing 1 of SDRAM

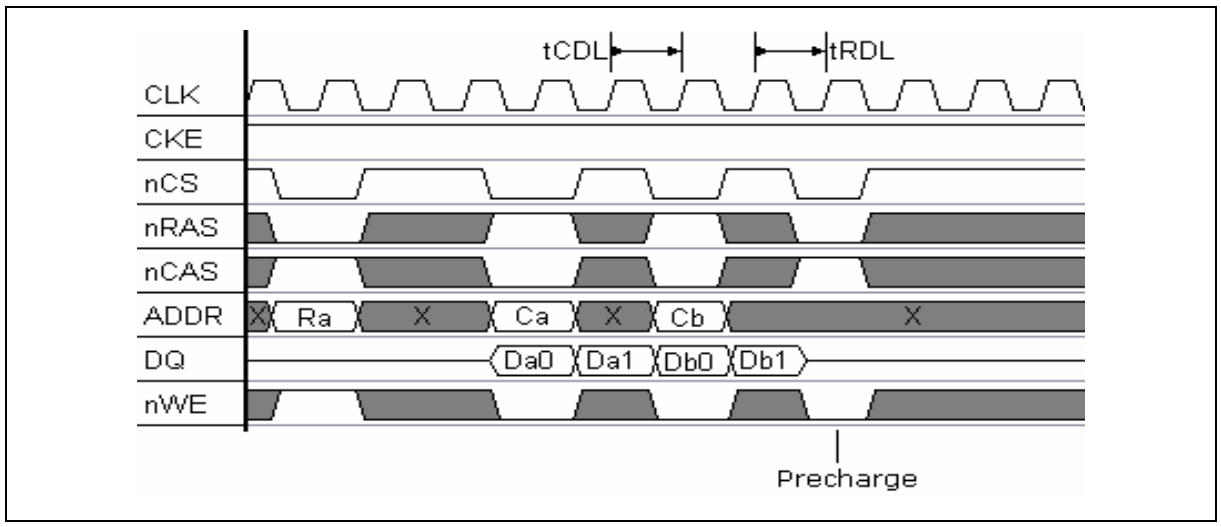


Figure 6.3.5 Access timing 2 of SDRAM

External I/O Control Registers（EXT0CON – EXT3CON）

The W90N745 supports an external device control without glue logic. It is very cost effective because address decoding and control signals timing logic are not needed. Using these control registers you can configure special external I/O devices for providing the low cost external devices control solution.

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
EXT0CON	0xFFF0_1018	R/W	External I/O 0 control register	0x0000_0000
EXT1CON	0xFFF0_101C	R/W	External I/O 1 control register	0x0000_0000
EXT2CON	0xFFF0_1020	R/W	External I/O 2 control register	0x0000_0000
EXT3CON	0xFFF0_1024	R/W	External I/O 3 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	DESCRIPTION	
[31:11]	BASADDR	**Base address pointer of external I/O bank 0~3** 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 0~3**

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
1	1	0	REVERSED
1	1	1	REVERSED

Continued.

<table>
<tr><th>BITS</th><th colspan="2">DESCRIPTION</th></tr>
<tr><td>[15]</td><td>ADRS</td><td>Address bus alignment for external I/O bank 0~3
When ADRS is set, external address (A20~A0) bus is alignment to byte address format, that is, A0 is internal AHB address bus HADDR[0] and A1 is AHB bus HADDR[1] and so forth. And it ignores DBWD [1:0] setting.</td></tr>
<tr><td>[14:11]</td><td>tACC</td><td>Access cycles of external I/O bank 0~3
This parameter means nWE, nWBE and nOE active time clock. Detail timing diagram please refer to Figure 6.3.6 and 6.3.7
<table>
<tr><th colspan="4">tACC[14:11]</th><th>MCLK</th><th colspan="4">tACC[14:11]</th><th>MCLK</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>Reversed</td><td>1</td><td>0</td><td>0</td><td>0</td><td>9</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td><td>1</td><td>11</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>0</td><td>2</td><td>1</td><td>0</td><td>1</td><td>0</td><td>13</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>1</td><td>3</td><td>1</td><td>0</td><td>1</td><td>1</td><td>15</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>0</td><td>4</td><td>1</td><td>1</td><td>0</td><td>0</td><td>17</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>1</td><td>5</td><td>1</td><td>1</td><td>0</td><td>1</td><td>19</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>0</td><td>6</td><td>1</td><td>1</td><td>1</td><td>0</td><td>21</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>1</td><td>7</td><td>1</td><td>1</td><td>1</td><td>1</td><td>23</td></tr>
</table></td></tr>
<tr><td>[10:8]</td><td>tCOH</td><td>Chip selection hold time of external I/O bank 0~3
This parameters control nWBE and nOE hold time. Detail timing diagram please refer to Figure 6.3.6 and 6.3.7
<table>
<tr><th colspan="3">tCOH [10:8]</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>
</table>

Continued.

<table>
<tr><th>BITS</th><th colspan="2">DESCRIPTION</th></tr>
<tr><td>[7:5]</td><td>tACS</td><td>
Address set-up before nECS for external I/O bank 0~3
<table>
<tr><th colspan="3">tACS [7:5]</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>[4:2]</td><td>tCOS</td><td>
Chip selection set-up time of external I/O bank 0~3
When ROM/Flash memory 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 0~3
<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>

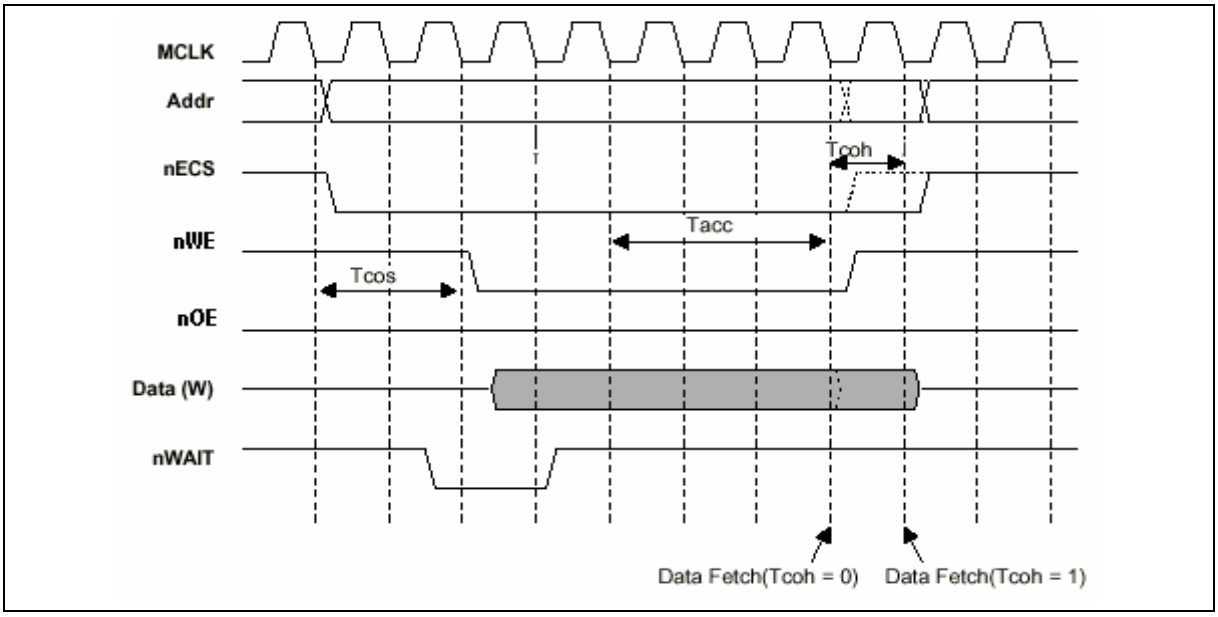


Figure 6.3.6 External I/O write operation timing

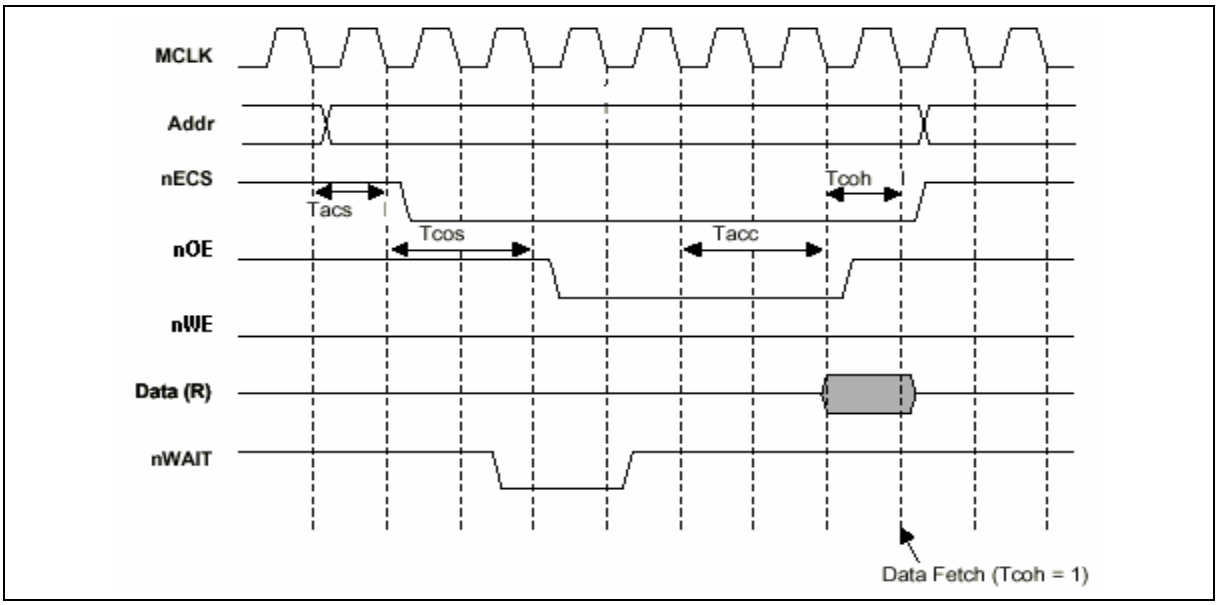


Figure 6.3.7 External I/O read operation timing

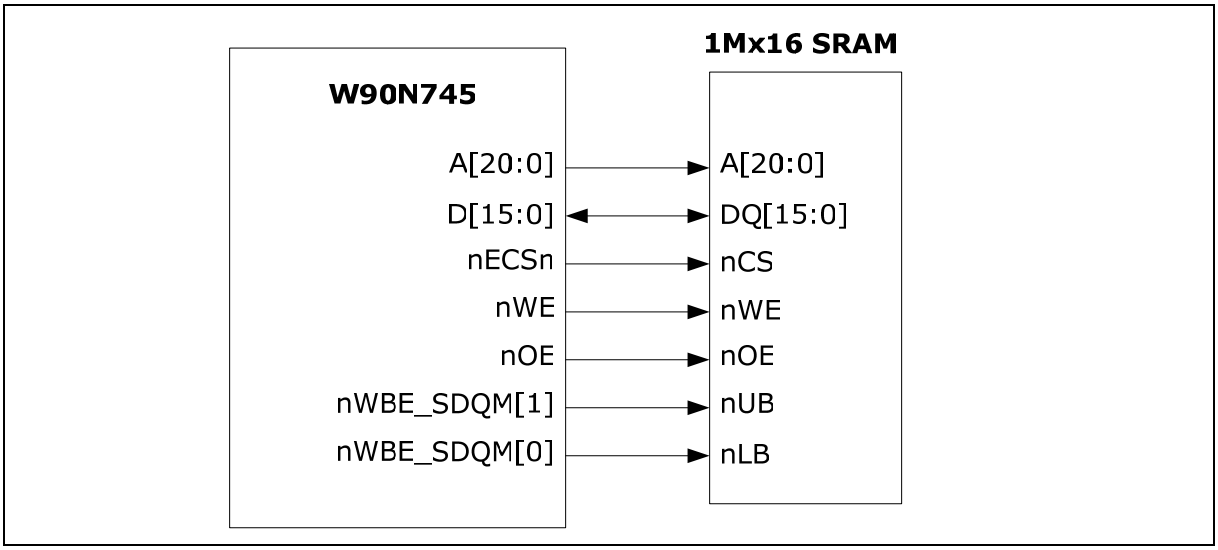


Figure 6.3.8 External IO bank with 16-bit SRAM

Clock Skew Control Register （CKSKEW）

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
CKSKEW	0xFFF0_1F00	R/W	Clock skew control register	0xXXXX_0018

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

<table>
<tr><th>BITS</th><th colspan="2">DESCRIPTION</th></tr>
<tr><td>[31:16]</td><td>DLH_CLK_REF</td><td>Latch DLH_CLK clock tree by HCLK positive edge
The SDRAM MCLK is generated by inserting a delay (XOR2) chain in HCLK positive or negedge edge to adjust the MCLK skew. So software can read these bits to expore MCLK and HCLK relationship. [31:24] is used for positive edge and [23:16] is for negedge edge.</td></tr>
<tr><td>[15:9]</td><td>RESERVED</td><td>-</td></tr>
<tr><td>[8]</td><td>SWPON</td><td>SDRAM Initialization by Software
Set this bit “1” will issue a SDRAM power on default setting command sequence like system power on, this bit will be auto-clear by hardware while SDRAM initialization finish.</td></tr>
<tr><td>[7:4]</td><td>DLH_CLK_SKEW</td><td>Data latch Clock Skew Adjustment
Due to PC board loading or too many devices connect to external address and data bus, it may causes SDRAM can not work correctly at high frequency (usually, > 80MHz) software can control MCLK_O_D[3:0] to adjust address and data bus to adjust setup/hold time.

<table>
<tr><th colspan="4">DLH_CLK_SKEW[7:4]</th><th>Gate Delay</th><th colspan="4">DLH_CLK_SKEW[7:4]</th><th>Gate Delay</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>P-0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>N-0</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>1</td><td>P-1</td><td>1</td><td>0</td><td>0</td><td>1</td><td>N-1</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>0</td><td>P-2</td><td>1</td><td>0</td><td>1</td><td>0</td><td>N-2</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>1</td><td>P-3</td><td>1</td><td>0</td><td>1</td><td>1</td><td>N-3</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>0</td><td>P-4</td><td>1</td><td>1</td><td>0</td><td>0</td><td>N-4</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>1</td><td>P-5</td><td>1</td><td>1</td><td>0</td><td>1</td><td>N-5</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>0</td><td>P-6</td><td>1</td><td>1</td><td>1</td><td>0</td><td>N-6</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>1</td><td>P-7</td><td>1</td><td>1</td><td>1</td><td>1</td><td>N-7</td></tr>
</table>
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.</td></tr>
</table>

Continued.

<table>
<tr><th>BITS</th><th>DESCRIPTION</th></tr>
<tr><td>[3:0]</td><td>MCLK_O_D</td><td>
MCLK output delay adjustment
<table>
<tr><th colspan="4">MCLK_O_D [3:0]</th><th>Gate Delay</th><th colspan="4">MCLK_O_D [3:0]</th><th>Gate Delay</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>P-0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>N-0</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>1</td><td>P-1</td><td>1</td><td>0</td><td>0</td><td>1</td><td>N-1</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>0</td><td>P-2</td><td>1</td><td>0</td><td>1</td><td>0</td><td>N-2</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>1</td><td>P-3</td><td>1</td><td>0</td><td>1</td><td>1</td><td>N-3</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>0</td><td>P-4</td><td>1</td><td>1</td><td>0</td><td>0</td><td>N-4</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>1</td><td>P-5</td><td>1</td><td>1</td><td>0</td><td>1</td><td>N-5</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>0</td><td>P-6</td><td>1</td><td>1</td><td>1</td><td>0</td><td>N-6</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>1</td><td>P-7</td><td>1</td><td>1</td><td>1</td><td>1</td><td>N-7</td></tr>
</table>
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.
</td></tr>
</table>

6.4 Cache Controller

The W90N745 incorporates a 4KB Instruction cache, 4KB Data cache and 8 words write buffer. The I-Cache and D-Cache have similar organization except the cache size. To raise the cache-hit ratio, these two caches are configured two-way set associative addressing. Each cache has four words cache line size. When a miss occurs, four words must be fetched consecutively from external memory. The replacement algorithm is a LRU (Least Recently Used).

If disabling the I-Cache / D-Cache, these cache memories can be treated as On-Chip RAM. The W90N745 also provides a write buffer to improve system performance. The write buffer can buffer up to eight words of data.

6.4.1 On-Chip RAM

If I-Cache or D-Cache is disabled, it can be served as On-Chip RAM. If D-Cache is disabled, there has 4KB On-Chip RAM, its start address is 0xFFE01000. If I-Cache is disabled, there has 4KB On-Chip RAM and the start address of this RAM is 0xFFE00000. If both the I-Cache and D-Cache are disabled, it has 8KB On-Chip RAM starting from 0xFFE00000.

The size of On-Chip RAM is depended on the I-Cache and D-Cache enable bits **ICAEN**, **DCAEN** in Cache Control Register (CAHCON).

Table 6.4.1 The size and start address of On-Chip RAM

ICAEN	DCAEN	ON-CHIP RAM	
		SIZE	START ADDRESS
0	0	8KB	0xFFE0_0000
0	1	4KB	0xFFE0_0000
1	0	4KB	0xFFE0.1000
1	1	Unavailable	

6.4.2 Non-Cacheable Area

Although the cache affects the entire 2GB system memory, it is sometimes necessary to define non-cacheable areas when the consistency of data stored in memory and the cache must be ensured. To support this, the W90N745 provides a non-cacheable area control bit in the address field, A[31].

If A[31] in the ROM/FLASH, SDRAM, or external I/O bank's access address is "0", then the accessed data is cacheable. If the A [31] value is "1", the accessed data is non-cacheable.

6.4.3 Instruction Cache

The Instruction cache (I-cache) is a 4K bytes two-way set associative cache. The cache organization is 128 sets, two lines per set, and four words per line. Cache lines are aligned on 4-word boundaries in memory. The cache access cycle begins with an instruction request from the instruction unit in the core. In the case of a cache hit, the instruction is delivered to the instruction unit. In case of a cache miss, the cache initiates a burst read cycle on the internal bus with the address of the requested instruction. The first word received from the bus is the requested instruction. The cache forwards this instruction to the instruction unit of the core as soon as it is received from the internal bus. A cache line is then selected to receive the data that will be coming from the bus. A least recently used (LRU) replacement algorithm is used to select a line when no empty lines are available. When I-Cache is disabled, the cache memory is served as 4KB On-chip RAM. The I-Cache is always disabled on reset.

The following is a list of the instruction cache features：

- 4K bytes instruction cache
- Two-way set associative
- Four words in a cache line
- LRU replacement policy
- Lockable on a per-line basis
- Critical word first, burst access

Instruction Cache Operation

On an instruction fetch, bits 10-4 of the instruction's address point into the cache to retrieve the tags and data of one set. The tags from both ways are then compared against bits 30-11 of the instruction's address. If a match is found and the matched entry is valid, then it is a cache hit. If neither tags match nor the matched tag is not valid, it is a cache miss.

Instruction Cache Hit

In case of a cache hit, bits 3-2 of the instruction address is used to select one word from the cache line whose tag matches. The instruction is immediately transferred to the instruction unit of the core.

Instruction Cache Miss

On an instruction cache miss, the address of the missed instruction is driven on the internal bus with a 4-word burst transfer read request. A cache line is then selected to receive the data that will be coming from the bus. The selection algorithm gives first priority to invalid lines. If neither of the two lines in the selected set is invalid, then the least recently used line is selected for replacement. Locked lines are never replaced. The transfer begins with the word requested by the instruction unit (critical word first), followed by the remaining words of the line, then by the word at the beginning of the lines (wraparound).

Instruction Cache Flushing

The W90N745 does not support external memory snooping. Therefore, if self-modifying code is written, the instructions in the I-Cache may become invalid. The entire I-Cache can be flushed by software in one operation, or can be flushed one line at a time by setting the **CAHCON** register bit **FLHS** or **FLHA** with the **ICAH** bit is set. As flushing the cache line, the "**V**" bit of the line is cleared to "0". The I-Cache is automatically flushed during reset.

Instruction Cache Load and Lock

The W90N745 supports a cache-locking feature that can be used to lock critical sections of code into I-Cache to guarantee quick access. Lockdown can be performed with a granularity of one cache line. The smallest space, which can be locked down, is 4 words. After a line is locked, it operates as a regular instruction SRAM. Lines locked are not replaced during misses and not affected by flush per line command.

To load and lock instruction, the following sequence should be followed:

1. Write the start address of the instructions to be locked into **CAHADR** register.
2. Set **LDLK** and **ICAH** bits in the **CAHCON** register.
3. Increased the address by 16 and written into **CAHADR** register.
4. Set **LDLK** and **ICAH** bits in the **CAHCON** register.
5. Repeat the steps 3 and 4, until the desired instructions are all locked.

When using I-Cache load and lock command, there are some notes should be cared.

- The programs executing load and lock operation should be held in a non-cacheable area of memory.
- The cache should be enabled and interrupts should be disabled.
- Software must flush the cache before execute load and lock to ensure that the code to be locked down is not already in the cache.

Instruction Cache Unlock

The unlock operation is used to unlock previously locked cache lines. After unlock, the "**L**" bit of the line is cleared to "0". W90N745 has two unlock command, unlock line and unlock all.

The unlock line operation is performed on a cache line granularity. In case the line is found in the cache, it is unlocked and starts to operate as a regular valid cache line. In case the line is not found in the cache, no operation is done and the command terminates with no exception. To unlock one line the following unlock line sequence should be followed:

1. Write the address of the line to be unlocked into the **CAHADR** Register.
2. Set the **ULKS** and **ICAH** bits in the **CAHCON** register.

The unlock all operation is used to unlock the whole I-Cache. This operation is performed on all cache lines. In case a line is locked, it is unlocked and starts to operate as regular valid cache line. In case a line is not locked or if it is invalid, no operation is performed. To unlock the whole cache, set the **ULKA** and **ICAH** bits.

6.4.4 Data Cache

The W90N745 data cache (D-Cache) is a 4KB two-way set associative cache. The cache organization is 128 sets, two lines per set, and four words per line. Cache lines are aligned on 4-word boundaries in memory. The cache is designed for **buffer write-through** mode of operation and a least recently used (LRU) replacement algorithm is used to select a line when no empty lines are available.

When D-Cache is disabled, the cache memory is served as 4KB On-chip RAM.

The D-Cache is always disabled on reset.

The following is a list of the data cache features：

- 4K bytes data cache
- Two-way set associative
- Four words in a cache line
- LRU replacement policy
- Lockable on a per-line basis
- Critical word first, burst access
- Buffer Write-through mode
- 8 words write buffer
- Drain write buffer

Data Cache Operation

On a data fetch, bits 10-4 of the data's address point into the cache to retrieve the tags and data of one set. The tags from both ways are then compared against bits 30-11 of the data's address. If a match is found and the matched entry is valid, then it is a cache hit. If neither tags match nor the matched tag is not valid, it is a cache miss.

Data Cache Read

Read Hit：On a cache hit, the requested word is immediately transferred to the core.

Read Miss：A line in the cache is selected to hold the data, which will be fetched from memory. The selection algorithm gives first priority to invalid lines and if both lines are invalid the line in way zero is selected first. If neither of the two candidate lines in the selected set is invalid, then one of the lines is selected by the LRU algorithm to replace. The transfer begins with the aligned word containing the missed data (critical word first), followed by the remaining word in the line, then by the word at the beginning of the line (wraparound). As the missed word is received from the bus, it is delivered directly to the core.

Data Cache Write

As buffer write-through mode, store operations always update memory. The buffer write-through mode is used when external memory and internal cache images must always agree.

Write Hit：Data is written into both the cache and write buffer. The processor then continues to access the cache, while the cache controller simultaneously downloads the contents of the write buffer to main memory. This reduces the effective write memory cycle time from the time required for a main memory cycle to the cycle time of the high-speed cache.

Write Miss：Data is only written into write buffer, not to the cache (write no allocate).

Data Cache Flushing

The W90N745 allows flushing of the data cache under software control. The data cache may be invalidated through writing flush line (**FLHS**) or flush all (**FLHA**) commands to the **CAHCON** register. Flushing the entire D-Cache also flushed any locked down code. As flushing the data cache, the "**V**" bit of the line is cleared to "0". The D-cache is automatically flushed during reset.

Data Cache Load and Lock

The W90N745 supports a cache-locking feature that can be used to lock critical sections of data into D-Cache to guarantee quick access. Lockdown can be performed with a granularity of one cache line. The smallest space, which can be locked down, is 4 words. After a line is locked, it operates as a regular instruction SRAM. The locked lines are not replaced during misses and it is not affected by flush per line command.

To load and lock data, the following sequence should be followed:

1. Write the start address of the data to be locked into **CAHADR** register.
2. Set **LDLK** and **DCAH** bits in the **CAHCON** register.
3. Increased the address by 16 and written into **CAHADR** register.
4. Set **LDLK** and **DCAH** bits in the **CAHCON** register.
5. Repeat the steps 3 and 4, until the desired data are all locked.

When using D-Cache load and lock command, there are some notes should be cared.

- The programs executing load and lock operation should be held in a non-cacheable area of memory.
- The cache should be enabled and interrupts should be disabled.
- Software must flush the cache before execute load and lock to ensure that the data to be locked down is not already in the cache.

Data Cache Unlock

The unlock operation is used to unlock previously locked cache lines. After unlock, the "**L**" bit of the line is cleared to "0". W90N745 has two unlock command, unlock line and unlock all.

The unlock line operation is performed on a cache line granularity. In case the line is found in the cache, it is unlocked and starts to operate as a regular valid cache line. In case the line is not found in the cache, no operation is done and the command terminates with no exception. To unlock one line the following unlock line sequence should be followed:

1. Write the address of the line to be unlocked into the **CAHADR** Register.
2. Set the **ULKS** and **DCAH** bits in the **CAHCON** register.

The unlock all operation is used to unlock the whole D-Cache. This operation is performed on all cache lines. In case a line is locked, it is unlocked and starts to operate as regular valid cache line. In case a line is not locked or if it is invalid, no operation is performed. To unlock the whole cache, set the **ULKA** and **DCAH** bits.

6.4.5 Write Buffer

The W90N745 provides a write buffer to improve system performance. The write buffer can buffer up to eight words of data. The write buffer may be enabled or be disabled via the **WRBEN** bit in the **CAHCNF** register, and the buffer is disabled and flushed on reset.

Drain write buffer

To force data, this is in write buffer, to be written to external main memory. This operation is useful in real time applications where the processor needs to be sure that a write to a peripheral has completed before program execution continues.

To perform this command, you can set the **DRWB** and **DCAH** bits in **CAHCON** register.

6.4.6 Cache Control Registers Map

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
CAHCNF	0xFFF0_2000	R/W	Cache configuration register	0x0000_0000
CAHCON	0xFFF0_2004	R/W	Cache control register	0x0000_0000
CAHADR	0xFFF0_2008	R/W	Cache address register	0x0000_0000
CTEST0	0xFFF6_0000	R/W	Cache test register 0	0x0000_0000
CTEST1	0xFFF6_0004	R	Cache test register 1	0x0000_0000

Configuration Register (CAHCNF)

Cache controller has a configuration register to enable or disable the I-Cache, D-Cache, and Write buffer.

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
CAHCNF	0xFFF0_2000	R/W	Cache configuration 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					WRBEN	DCAEN	ICAEN

BITS	DESCRIPTION	
[31:3]	RESERVED	-
[2]	WRBEN	**Write buffer enable** Write buffer is disabled after reset. 1 = enable write buffer 0 = disable write buffer
[1]	DCAEN	**D-Cache enable** D-Cache is disabled after reset. 1 = enable D-cache 0 = disable D-cache
[0]	ICAEN	**I-Cache enable** I-Cache is disabled after reset. 1 = enable I-cache 0 = disable I-cache

Control Register (CAHCON)

Cache controller supports one Control register used to control the following operations.

- Flush I-Cache and D-Cache
- Load and lock I-Cache and D-Cache
- Unlock I-Cache and D-Cache
- Drain write buffer

These command set bits in **CAHCON** register are auto-clear bits. As the end of execution, that command set bit will be cleared to “0” automatically.

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
CAHCON	0xFFF0_2004	R/W	Cache 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
DRWB	ULKS	ULKA	LDLK	FLHS	FLHA	DCAH	ICAH

BITS	DESCRIPTION	
[31:8]	RESERVED	-
[7]	DRWB	**Drain write buffer** Forces write buffer data to be written to main memory.
[6]	ULKS	**Unlock I-Cache/D-Cache single line** Unlocks the I-Cache/D-Cache per line. Both **WAY** and **ADDR** bits in **CAHADR** register must be specified.
[5]	ULKA	**Unlock I-Cache/D-Cache entirely** Unlocks the entire I-Cache/D-Cache, the lock bit “**L**” will be cleared to 0.
[4]	LDLK	**Load and Lock I-Cache/D-Cache** Loads the instruction or data from external memory and locks into cache. Both **WAY** and **ADDR** bits in **CAHADR** register must be specified.

Continued.

BITS	DESCRIPTION	
[3]	FLHS	**Flush I-Cache/D-Cache single line** Flushes the entire I-Cache/D-Cache per line. Both **WAY** and **ADDR** bits in **CAHADR** register must be specified.
[2]	FLHA	**Flush I-Cache/D-Cache entirely** To flush the entire I-Cache/D-Cache, also flushes any locked-down code. If the I-Cache/D-Cache contains locked down code, the programmer must flush lines individually
[1]	DCAH	**D-Cache selected** When set to "1", the command set is executed with D-Cache.
[0]	ICAH	**I-Cache selected** When set to "1", the command set is executed with I-Cache.

NOTE：When using the **FLHA** or **ULKA** command, you can set **both ICAH** and **DCAH** bits to execute entire I-Cache **and** D-Cache flushing or unlocking. But, **FLHS** and **ULKS** commands can only be executed with a cache line specified by **CAHADR** register in I-Cache **or** D-Cache at a time. If you set **both ICAH** and **DCAH** bits, and set **FLHS** or **ULKS** command bit, it will be treated as an invalid command and no operation is done and the command terminates with no exception.

The **Drain Write Buffer** operation is only for D-Cache. To perform this operation, you must set **DRWB** and **DCAH** bits. If the **ICAH** bit is set when using **DRWB** command, it will be an invalid command and no operation is done and the command terminates with no exception.

Address Register (CAHADR)

W90N745 Cache Controller supports one address register. This address register is used with the command set in the control register (**CAHCON**) by specifying instruction/data address.

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
CAHADR	0xFFF0_2008	R/W	Cache address register	0x0000_0000

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

BITS	DESCRIPTION	
[31]	WAY	**Way selection** 0 = Way0 is selected 1 = Way1 is selected
[30:0]	ADDR	The absolute address of instruction or data

Cache Test Register 0 (CTEST0)

Cache test control register that configures the cache and tag ram testing enable or disable. In addition, this register controls the built-in-self-test (BIST) function of SRAM.

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
CTEST0	0xFFF6_0000	R/W	Cache test register 0	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
BISTEN	RESERVED			BST_GP3	BST_GP2	BST_GP1	BST_GP0
7	6	5	4	3	2	1	0
RESERVED							CATEST

BITS	DESCRIPTION	
[31:16]	RESERVED	-
[15]	BISTEN	**BIST mode enable** When set to “1”, BIST mode will be enabled, the selected memory groups begins to be tested by BIST.
[14:12]	RESERVED	-
[11]	BIST_GP3	**Memory group 3 is selected to test by BIST** When set to “1”, memory group 3, including data cache tag ram way 0 and way 1, are selected to be tested by BIST.
[10]	BIST_GP2	**Memory group 2 is selected to test by BIST** When set to “1”, memory group 2, including program cache tag ram way 0 and way 1, are selected to be tested by BIST.
[9]	BIST_GP1	**Memory group 1 is selected to test by BIST** When set to “1”, memory group 1, including data cache ram way 0 and way 1, are selected to be tested by BIST.
[8]	BIST_GP0	**Memory group 0 is selected to test by BIST** When set to “1”, memory group 0, including program cache ram way 0 and way 1, are selected to be tested by BIST.
[7:0]	RESERVED	-

** **Note**: The 4 memory groups can be selected and tested simultaneously by BIST.

Cache Test Register 1 (CTEST1)

Cache Test Register that will be read back to provide the status of cache RAM BIST. Whether the BIST is finish and all of bank of SRAM are tested successfully will be presented in this register.

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
CTEST1	0xFFF6_0004	R	Cache test register 1	0x0000_0000

31	30	29	28	27	26	25	24
FINISH	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
BFAIL7	BFAIL6	BFAIL5	BFAIL4	BFAIL3	BFAIL2	BFAIL1	BFAIL0

BITS	DESCRIPTION	
[31]	FINISH	**BIST completed** This bit is "0" initially. When BIST mode enabled, this bit will be "1" after BIST test completed. The values of BFAIL0-7 are valid only after FINISH = 1.
[30:8]	RESERVED	-
[7]	BFAIL7	**BIST test fail for data cache tag ram way 1** If this bit equals to "1", it indicates the data cache tag ram for way 1 is tested fail by BIST. "0" means the test is passed.
[6]	BFAIL6	**BIST test fail for data cache tag ram way 0** If this bit equals to "1", it indicates the data cache tag ram for way 0 is tested fail by BIST. "0" means the test is passed.
[5]	BFAIL5	**BIST test fail for instruction cache tag ram way 1** If this bit equals to "1", it indicates the instruction cache tag ram for way 1 is tested fail by BIST. "0" means the test is passed.
[4]	BFAIL4	**BIST test fail for instruction cache tag ram way 0** If this bit equals to "1", it indicates the instruction cache tag ram for way 0 is tested fail by BIST. "0" means the test is passed.
[3]	BFAIL3	**BIST test fail for data cache ram way 1** If this bit equals to "1", it indicates the data cache ram for way 1 is tested fail by BIST. "0" means the test is passed.

Continued.

BITS	DESCRIPTION	
[2]	BFAIL2	**BIST test fail for data cache ram way 0** If this bit equals to “1”, it indicates the data cache ram for way 0 is tested fail by BIST. “0” means the test is passed.
[1]	BFAIL1	**BIST test fail for instruction cache ram way 1** If this bit equals to “1”, it indicates the instruction cache ram for way 1 is tested fail by BIST. “0” means the test is passed.
[0]	BFAIL0	**BIST test fail for instruction cache ram way 0** If this bit equals to “1”, it indicates the instruction cache ram for way 0 is tested fail by BIST. “0” means the test is passed.

6.5 Ethernet MAC Controller

Overview

The W90N745 provides an Ethernet MAC Controller (EMC) for 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.

6.5.1 EMC Functional Description

MII Management State Machine

The MII management function of EMC is compliant to IEEE 802.3 Std. Through the MII management interface, software can access the control and status registers of the external PHY chip. Tow programmable register MIID (MAC MII Management Data Register) and MIIDA (MAC MII Management Data Control and Address Register) are for MII management function. Set the bit BUSY of MIIDA register will trigger the MII management state machine. After the MII management cycle is finished, the BUSY bit will be cleared automatically.

Media Access Control (MAC)

The function of W90N745 MAC fully meets the requirements defined by the IEEE802.3u specification. The following paragraphs will describe the frame structure and the operation of the transmission and receive.

The transmission data frame sent from the transmit DMA will be encapsulated by the MAC before transmitting onto the MII bus. The sent data will be assembled with the preamble, the start frame delimiter (SFD), the frame check sequence and the padding for enforcing those less than 64 bytes to meet the minimum size frame and CRC sequence. The out going frame format will be as following

10101010 --- 10101010	10101011	d0	d1	d2	-	dn	Padding	CRC31	CRC30	---	CRC0

As mentioned by the above format, the preamble is a consecutive 7-byte long with the pattern “10101010” and the SFD is a one byte 10101011 data. The padding data will be all 0 value if the sent data frame is less than 64 bytes. The padding disable function specified in the bit P of the transmit descriptor is used to control if the MAC needs to pad data at the end of frame data or not when the transmitted data frame is less than 64 bytes. The padding data will not be appended if the padding disable bit is set to be high. The bits CRC0 ... CRC31 are the 32 bits cyclic redundancy check (CRC) sequence. The CRC encoding is defined by the following polynomial specified by the IEEE802.3. This 32 bits CRC appending function will be disabled if the Inhibit CRC of the transmission descriptor is set to high.

The MAC also performs many other transmission functions specified by the IEEE802.3, including the inter-frame spacing function, collision detection, collision enforcement, collision back off and retransmission. The collision back-off timer is a function of the integer slot time, 512 bit times. The number of slot times to delay between the current transmissions attempts to the next attempt is determined by a uniformly distributed random integer algorithm specified by the IEEE802.3. The MAC performs the receive functions specified by the IEEE 802.3 including the address recognition function, the frame check sequence validation, the frame disassembly, framing and collision filtering.

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 W90N745. 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.

6.5.1.1. 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							

Owner [31:30]: 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.

Rx Status [29:16]: 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.

RP [22]: 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.

ALIE [21]: 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.

RXGD [20]: 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.

PTLE [19]: 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.

CRCE [17]: 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.

RXINTR [16]: 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.

RBC [15:0]: 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	

RXBSA [31:2]: 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.

BO [1:0]: 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							

NRXDSA [31:0]: 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.

6.5.1.2. Tx Buffer Descriptor

31	30	16	15	4	3	2	1	0
O	Reserved					I	C	P
Transmit Buffer Starting Address								BO
Tx Status					Transmit Byte Count			
Next Tx Descriptor Starting Address								

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

Owner [31]: 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.

IntEn [2]: 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.

CRCApp [1]: 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.

PadEN [0]: 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	

TXBSA [31:2]: 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.

BO [1:0]: 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							

CCNT [31:28]: Collision Count

The CCNT indicates the how many collision 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.

SQE [26]: 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.

PAU [25]: 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.

TXHA [24]: 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.

LC [23]: 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.

TXABT [22]: 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.

NCS [21]: 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.

EXDEF [20]: 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).

TXCP [19]: 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.

DEF [17]: 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.

TXINTR [16]: 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.

TBC [15:0]: 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							

NTXDSA [31:0]: 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.

6.5.2 EMC Register Mapping

The EMC implements many registers and the registers are separated into three types, the control registers, the status registers and diagnostic 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. And, the diagnostic registers are used for debug only.

EMC Registers

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
CONTROL REGISTERS (44)				
CAMCMR	0xFFF0_3000	R/W	CAM Command Register	0x0000_0000
CAMEN	0xFFF0_3004	R/W	CAM Enable Register	0x0000_0000
CAM0M	0xFFF0_3008	R/W	CAM0 Most Significant Word Register	0x0000_0000
CAM0L	0xFFF0_300C	R/W	CAM0 Least Significant Word Register	0x0000_0000
CAM1M	0xFFF0_3010	R/W	CAM1 Most Significant Word Register	0x0000_0000
CAM1L	0xFFF0_3014	R/W	CAM1 Least Significant Word Register	0x0000_0000
CAM2M	0xFFF0_3018	R/W	CAM2 Most Significant Word Register	0x0000_0000
CAM2L	0xFFF0_301C	R/W	CAM2 Least Significant Word Register	0x0000_0000
CAM3M	0xFFF0_3020	R/W	CAM3 Most Significant Word Register	0x0000_0000
CAM3L	0xFFF0_3024	R/W	CAM3 Least Significant Word Register	0x0000_0000
CAM4M	0xFFF0_3028	R/W	CAM4 Most Significant Word Register	0x0000_0000
CAM4L	0xFFF0_302C	R/W	CAM4 Least Significant Word Register	0x0000_0000
CAM5M	0xFFF0_3030	R/W	CAM5 Most Significant Word Register	0x0000_0000
CAM5L	0xFFF0_3034	R/W	CAM5 Least Significant Word Register	0x0000_0000
CAM6M	0xFFF0_3038	R/W	CAM6 Most Significant Word Register	0x0000_0000
CAM6L	0xFFF0_303C	R/W	CAM6 Least Significant Word Register	0x0000_0000
CAM7M	0xFFF0_3040	R/W	CAM7 Most Significant Word Register	0x0000_0000
CAM7L	0xFFF0_3044	R/W	CAM7 Least Significant Word Register	0x0000_0000
CAM8M	0xFFF0_3048	R/W	CAM8 Most Significant Word Register	0x0000_0000
CAM8L	0xFFF0_304C	R/W	CAM8 Least Significant Word Register	0x0000_0000
CAM9M	0xFFF0_3050	R/W	CAM9 Most Significant Word Register	0x0000_0000
CAM9L	0xFFF0_3054	R/W	CAM9 Least Significant Word Register	0x0000_0000
CAM10M	0xFFF0_3058	R/W	CAM10 Most Significant Word Register	0x0000_0000
CAM10L	0xFFF0_305C	R/W	CAM10 Least Significant Word Register	0x0000_0000
CAM11M	0xFFF0_3060	R/W	CAM11 Most Significant Word Register	0x0000_0000
CAM11L	0xFFF0_3064	R/W	CAM11 Least Significant Word Register	0x0000_0000

Continued.

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
CONTROL REGISTERS (44)				
CAM12M	0xFFF0_3068	R/W	CAM12 Most Significant Word Register	0x0000_0000
CAM12L	0xFFF0_306C	R/W	CAM12 Least Significant Word Register	0x0000_0000
CAM13M	0xFFF0_3070	R/W	CAM13 Most Significant Word Register	0x0000_0000
CAM13L	0xFFF0_3074	R/W	CAM13 Least Significant Word Register	0x0000_0000
CAM14M	0xFFF0_3078	R/W	CAM14 Most Significant Word Register	0x0000_0000
CAM14L	0xFFF0_307C	R/W	CAM14 Least Significant Word Register	0x0000_0000
CAM15M	0xFFF0_3080	R/W	CAM15 Most Significant Word Register	0x0000_0000
CAM15L	0xFFF0_3084	R/W	CAM15 Least Significant Word Register	0x0000_0000
TXDLSA	0xFFF0_3088	R/W	Transmit Descriptor Link List Start Address Register	0xFFFF_FFFC
RXDLSA	0xFFF0_308C	R/W	Receive Descriptor Link List Start Address Register	0xFFFF_FFFC
MCMDR	0xFFF0_3090	R/W	MAC Command Register	0x0000_0000
MIID	0xFFF0_3094	R/W	MII Management Data Register	0x0000_0000
MIIDA	0xFFF0_3098	R/W	MII Management Control and Address Register	0x0090_0000
FFTCR	0xFFF0_309C	R/W	FIFO Threshold Control Register	0x0000_0101
TSDR	0xFFF0_30A0	W	Transmit Start Demand Register	Undefined
RSDR	0xFFF0_30A4	W	Receive Start Demand Register	Undefined
DMARFC	0xFFF0_30A8	R/W	Maximum Receive Frame Control Register	0x0000_0800
MIEN	0xFFF0_30AC	R/W	MAC Interrupt Enable Register	0x0000_0000
Status Registers (11)				
MISTA	0xFFF0_30B0	R/W	MAC Interrupt Status Register	0x0000_0000
MGSTA	0xFFF0_30B4	R/W	MAC General Status Register	0x0000_0000
MPCNT	0xFFF0_30B8	R/W	Missed Packet Count Register	0x0000_7FFF
MRPC	0xFFF0_30BC	R	MAC Receive Pause Count Register	0x0000_0000
MRPCC	0xFFF0_30C0	R	MAC Receive Pause Current Count Register	0x0000_0000
MREPC	0xFFF0_30C4	R	MAC Remote Pause Count Register	0x0000_0000
DMARFS	0xFFF0_30C8	R/W	DMA Receive Frame Status Register	0x0000_0000

Continued.

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
Status Registers (11)				
CTXDSA	0xFFF0_30CC	R	Current Transmit Descriptor Start Address Register	0x0000_0000
CTXBSA	0xFFF0_30D0	R	Current Transmit Buffer Start Address Register	0x0000_0000
CRXDSA	0xFFF0_30D4	R	Current Receive Descriptor Start Address Register	0x0000_0000
CRXBSA	0xFFF0_30D8	R	Current Receive Buffer Start Address Register	0x0000_0000
Diagnostic Registers (7)				
RXFSM	0xFFF0_3200	R	Receive Finite State Machine Register	0x0081_1101
TXFSM	0xFFF0_3204	R	Transmit Finite State Machine Register	0x0101_1101
FSM0	0xFFF0_3208	R	Finite State Machine Register 0	0x0001_0101
FSM1	0xFFF0_320C	R	Finite State Machine Register 1	0x1100_0100
DCR	0xFFF0_3210	R/W	Debug Configuration Register	0x0000_003F
DMMIR	0xFFF0_3214	R	Debug Mode MAC Information Register	0x0000_0000
BISTR	0xFFF0_3300	R/W	BIST Mode Register	0x0000_0000

6.5.2.1. Register Details

CAM Command Register (CAMCMR)

The EMC of W90N745 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	0xFFF0_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			ECMP	CCAM	ABP	AMP	AUP

BITS	DESCRIPTIONS	
[31:5]	Reserved	-
[4]	ECMP	The **ECMP(Enable CAM Compare)** 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	The **CCAM(Complement CAM Compare)** 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.

Continued.

BITS		DESCRIPTIONS
[2]	ABP	The **Accept Broadcast Packet** controls the broadcast packet reception. If ABP is enabled, EMC receives all incoming packet it's 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	The **Accept Multicast Packet** controls the multicast packet reception. If AMP is enabled, EMC receives all incoming packet it's 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	The **Accept Unicast Packet** controls the unicast packet reception. If AUP is enabled, EMC receives all incoming packet it's 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

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	0xFFF0_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	
[31:16]		Reserved
[15:13]	CAM15EN CAM14EN CAM13EN	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.
[12]	CAM12EN	CAM entry 12 is enabled 1'b0: CAM entry 12 disabled. 1'b1: CAM entry 12 enabled.
[11]	CAM11EN	CAM entry 11 is enabled 1'b0: CAM entry 11 disabled. 1'b1: CAM entry 11 enabled.
[10]	CAM10EN	CAM entry 10 is enabled 1'b0: CAM entry 10 disabled. 1'b1: CAM entry 10 enabled.

Continued.

BITS	DESCRIPTIONS	
[9]	CAM9EN	CAM entry 9 is enabled 1'b0: CAM entry 9 disabled. 1'b1: CAM entry 9 enabled.
[8]	CAM8EN	CAM entry 8 is enabled 1'b0: CAM entry 8 disabled. 1'b1: CAM entry 8 enabled.
[7]	CAM7EN	CAM entry 7 is enabled 1'b0: CAM entry 7 disabled. 1'b1: CAM entry 7 enabled.
[6]	CAM6EN	CAM entry 6 is enabled 1'b0: CAM entry 6 disabled. 1'b1: CAM entry 6 enabled.
[5]	CAM5EN	CAM entry 5 is enabled 1'b0: CAM entry 5 disabled. 1'b1: CAM entry 5 enabled.
[4]	CAM4EN	CAM entry 4 is enabled 1'b0: CAM entry 4 disabled. 1'b1: CAM entry 4 enabled.
[3]	CAM3EN	CAM entry 3 is enabled 1'b0: CAM entry 3 disabled. 1'b1: CAM entry 3 enabled.
[2]	CAM2EN	CAM entry 2 is enabled 1'b0: CAM entry 2 disabled. 1'b1: CAM entry 2 enabled.
[1]	CAM1EN	CAM entry 1 is enabled 1'b0: CAM entry 1 disabled. 1'b1: CAM entry 1 enabled.
[0]	CAM0EN	CAM entry 0 is enabled 1'b0: CAM entry 0 disabled. 1'b1: CAM entry 0 enabled.

CAM Entry Registers (CAMxx)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
CAM0M	0xFFF0_3008	R/W	CAM0 Most Significant Word Register	0x0000_0000
CAM0L	0xFFF0_300C	R/W	CAM0 Least Significant Word Register	0x0000_0000
CAM1M	0xFFF0_3010	R/W	CAM1 Most Significant Word Register	0x0000_0000
CAM1L	0xFFF0_3014	R/W	CAM1 Least Significant Word Register	0x0000_0000
CAM2M	0xFFF0_3018	R/W	CAM2 Most Significant Word Register	0x0000_0000
CAM2L	0xFFF0_301C	R/W	CAM2 Least Significant Word Register	0x0000_0000
CAM3M	0xFFF0_3020	R/W	CAM3 Most Significant Word Register	0x0000_0000
CAM3L	0xFFF0_3024	R/W	CAM3 Least Significant Word Register	0x0000_0000
CAM4M	0xFFF0_3028	R/W	CAM4 Most Significant Word Register	0x0000_0000
CAM4L	0xFFF0_302C	R/W	CAM4 Least Significant Word Register	0x0000_0000
CAM5M	0xFFF0_3030	R/W	CAM5 Most Significant Word Register	0x0000_0000
CAM5L	0xFFF0_3034	R/W	CAM5 Least Significant Word Register	0x0000_0000
CAM6M	0xFFF0_3038	R/W	CAM6 Most Significant Word Register	0x0000_0000
CAM6L	0xFFF0_303C	R/W	CAM6 Least Significant Word Register	0x0000_0000
CAM7M	0xFFF0_3040	R/W	CAM7 Most Significant Word Register	0x0000_0000
CAM7L	0xFFF0_3044	R/W	CAM7 Least Significant Word Register	0x0000_0000
CAM8M	0xFFF0_3048	R/W	CAM8 Most Significant Word Register	0x0000_0000
CAM8L	0xFFF0_304C	R/W	CAM8 Least Significant Word Register	0x0000_0000
CAM9M	0xFFF0_3050	R/W	CAM9 Most Significant Word Register	0x0000_0000
CAM9L	0xFFF0_3054	R/W	CAM9 Least Significant Word Register	0x0000_0000
CAM10M	0xFFF0_3058	R/W	CAM10 Most Significant Word Register	0x0000_0000
CAM10L	0xFFF0_305C	R/W	CAM10 Least Significant Word Register	0x0000_0000
CAM11M	0xFFF0_3060	R/W	CAM11 Most Significant Word Register	0x0000_0000
CAM11L	0xFFF0_3064	R/W	CAM11 Least Significant Word Register	0x0000_0000
CAM12M	0xFFF0_3068	R/W	CAM12 Most Significant Word Register	0x0000_0000
CAM12L	0xFFF0_306C	R/W	CAM12 Least Significant Word Register	0x0000_0000
CAM13M	0xFFF0_3070	R/W	CAM13 Most Significant Word Register	0x0000_0000
CAM13L	0xFFF0_3074	R/W	CAM13 Least Significant Word Register	0x0000_0000
CAM14M	0xFFF0_3078	R/W	CAM14 Most Significant Word Register	0x0000_0000
CAM14L	0xFFF0_307C	R/W	CAM14 Least Significant Word Register	0x0000_0000
CAM15M	0xFFF0_3080	R/W	CAM15 Most Significant Word Register	0x0000_0000
CAM15L	0xFFF0_3084	R/W	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	The CAMxM(CAMx Most Significant Word) 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	
[31:16]	CAMxL	The **CAMxL(CAMx Least Significant Word)** 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.
[15:0]	Reserved	-

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	
[31:0]	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.
[15:0]		Reserved

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	0xFFF0_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	The **TXDLSA(Transmit Descriptor Link-List Start Address)** 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	0xFFF0_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]	RXDLSA	The **RXDLSA(Receive Descriptor Link-List Start Address)** 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	0xFFF0_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	
[31:25]	Reserved	-
[24]	SWR	The **SWR (Software Reset)** 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, except for OPMOD bit of MCMDR register. The EMC re-initial is needed after the software reset completed. 1'b0: Software reset completed. 1'b1: Enable software reset.
[23:22]	Reserved	-
[21]	LBK	The **LBK (Internal Loop Back Select)** 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	The **Operation Mode Select** 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	The **Enable MDC Clock Generation** 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	The **Full Duplex Mode Select** 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.

Continued.

BITS	DESCRIPTIONS	
[17]	EnSQE	The **Enable SQE Checking** 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	The **Send PAUSE Frame** 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.
[15:10]	Reserved	-
[9]	NDEF	The **No Defer** 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.

Continued.

BITS	DESCRIPTIONS	
[8]	TXON	The Frame Transmission ON 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.
[7:6]	Reserved	-
[5]	SPCRC	The Strip CRC Checksum 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	The Accept CRC Error Packet 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	The Accept Control Packet 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.

Continued.

BITS	DESCRIPTIONS	
[2]	ARP	The Accept Runt Packet 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	The Accept Long Packet 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	The Frame Reception ON 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	0xFFF0_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	
[31:16]	Reserved	-
[15:0]	MIIData	The MII Management Data 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	0xFFF0_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	
[31:24]	Reserved	-
[23:20]	MDCCR	The MDC Clock Rating 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.
[19]	MDC	The MDC Clock ON Always 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.

Continued.

BITS	DESCRIPTIONS	
[18]	PreSP	The Preamble Suppress 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	The Busy Bit 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	The Write Command 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.
[15:13]		Reserved
[12:8]	PHYAD	The PHY Address keeps the address to differentiate which external PHY is the target of the MII management command.
[7:5]	Reserved	-
[4:0]	PHYRAD	The PHY Register Address keeps the address to indicate which register of external PHY is the target of the MII management command.

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

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.	1. Write data to MIID register
2. Set PHYAD and PHYRAD.	2. Set appropriate MDCCR.
3. Set Write to 1'b0	3. Set PHYAD and PHYRAD.
4. Set bit BUSY to 1'b1 to send a MII management frame out.	4. Set Write to 1'b1
5. Wait BUSY to become 1'b0.	5. Set bit BUSY to 1'b1 to send a MII management frame out.
6. Read data from MIID register.	6. Wait BUSY to become 1'b0.
7. Finish the read command.	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	0xFFF0_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	
[31:22]	Reserved	-
[21:20]	Blength	The **DMA Burst Length** 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
[19:10]	Reserved	-

Continued.

BITS	DESCRIPTIONS	
[9:8]	TxTHD	The **TxFIFO Low Threshold** 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.
[7:2]		Reserved
[1:0]	RxTHD	The **RxFIFO High Threshold** 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	0xFFF0_30A0	W	Transmit Start Demand Register	Undefined

BITS	DESCRIPTIONS	
[31:0]	Reserved	-

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	0xFFF0_30A4	W	Receive Start Demand Register	Undefined

BITS	DESCRIPTIONS	
[31:0]	Reserved	--

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	0xFFF0_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	
[31:16]	Reserved	-
[15:0]	RXMS	The **Maximum Receive Frame Length** 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	0xFFF0_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	
[31:25]	Reserved	-
[24]	EnTxBErr	The **Enable Transmit Bus Error Interrupt** 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	The **Enable Transmit Descriptor Unavailable** Interrupt 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.

Continued

BITS	DESCRIPTIONS	
[22]	EnLC	The **Enable Late Collision Interrupt** 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	The **Enable Transmit Abort Interrupt** 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	The **Enable No Carrier Sense Interrupt** 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	The **Enable Defer Exceed Interrupt** 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	The **Enable Transmit Completion Interrupt** 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.

Continued.

BITS	DESCRIPTIONS	
[17]	EnTXEMP	The **Enable Transmit FIFO Underflow Interrupt** 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	The **EnTXINTR** controls the Tx interrupt generation. If **Enable Transmit Interrupt** 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.
[15]	Reserved	--
[14]	EnCFR	The **Enable Control Frame Receive Interrupt** 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.
[13:12]	Reserved	--
[11]	EnRxBErr	The **Enable Receive Bus Error Interrupt** 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.

Continued.

BITS	DESCRIPTIONS	
[10]	EnRDU	The **Enable Receive Descriptor Unavailable Interrupt** 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	The **Enable DMA Early Notification Interrupt** 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	The **Enable Maximum Frame Length Interrupt** 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	The **Enable More Missed Packet Interrupt** 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	The **Enable Runt Packet Interrupt** 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.

Continued.

BITS	DESCRIPTIONS	
[5]	EnALIE	The **Enable Alignment Error Interrupt** 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	The **Enable Receive Good Interrupt** 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	The **Enable Packet Too Long Interrupt** 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	The **Enable Receive FIFO Overflow Interrupt** 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.

Continued.

BITS	DESCRIPTIONS	
[1]	EnCRCE	The **Enable CRC Error Interrupt** 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	The **Enable Receive Interrupt** 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	0xFFF0_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	
[31:25]	Reserved	-
[24]	TxBErr	The **Transmit Bus Error Interrupt** 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	The **Transmit Descriptor Unavailable Interrupt** 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.

Continued.

BITS	DESCRIPTIONS	
[22]	LC	The Late Collision Interrupt 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	The **Transmit Abort Interrupt** 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	The **No Carrier Sense Interrupt** 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	The Defer Exceed Interrupt 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).

Continued.

BITS	DESCRIPTIONS	
[18]	TXCP	The **Transmit Completion Interrupt** 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	The **Transmit FIFO Underflow Interrupt** 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	The Transmit Interrupt 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 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.
[15]		Reserved

Continued.

BITS	DESCRIPTIONS	
[14]	CFR	The **Control Frame Receive Interrupt** 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.
[13:12]		Reserved
[11]	RxBErr	The **Receive Bus Error Interrupt** 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	The Receive Descriptor Unavailable Interrupt 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	The DMA Early Notification Interrupt 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.

Continued.

BITS	DESCRIPTIONS	
[8]	DFOI	The **Maximum Frame Length Interrupt** 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	The **More Missed Packet Interrupt** 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	The Alignment Error Interrupt 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.

Continued.

BITS	DESCRIPTIONS	
[4]	RXGD	The **Receive Good Interrupt** 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	The **Packet Too Long Interrupt** 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	The **Receive FIFO Overflow Interrupt** 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	The **CRC Error Interrupt** 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.

Continued.

BITS	DESCRIPTIONS	
[0]	RXINTR	The **Receive Interrupt** 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 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	0xFFF0_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	
[31:12]	Reserved	-
[11]	TXHA	The Transmission Halted 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	The Signal Quality Error 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	The Transmission Paused 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	The Deferred Transmission 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	The Collision Count indicates the how many collision 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.
[3]	Reserved	-
[2]	RFFull	The RxFIFO Full 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	The Receive Halted 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	The Control Frame Received 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	0xFFF0_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	
[31:16]	Reserved	-
[15:0]	MPC	The **Miss Packet Count** 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 W90N745 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	0xFFF0_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	
[31:16]	Reserved	-
[15:0]	MRPC	The MAC Receive Pause Count 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 W90N745 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	0xFFF0_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	
[31:16]	Reserved	-
[15:0]	MRPCC	The **MAC Receive Pause Current Count** 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 W90N745 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	0xFFF0_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	
[31:16]	Reserved	
[15:0]	MREPC	The **MAC Remote Pause Count** 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	0xFFF0_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	
[31:16]	Reserved	
[15:0]	RXFLT	The Receive Frame Length/Type 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)

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.

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
CTXDSA	0xFFF0_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

Current Transmit Buffer Start Address Register (CTXBSA)

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.

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
CTXBSA	0xFFF0_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**

Current Receive Descriptor Start Address Register (CRXDSA)

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.

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
CRXDSA	0xFFF0_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**

Current Receive Buffer Start Address Register (CRXBSA)

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.

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
CRXBSA	0xFFF0_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

Receive Finite State Machine Register (RXFSM)

The RXFSM shows the current value of the FSM (Finite State Machine) of RxDMA and RxFIFO controller. The RXFSM is read only and write to it has no effect. The RXFSM is used only for debug.

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
RXFSM	0xFFF0_3200	R	Receive Finite State Machine Register	0x0081_1101

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

BITS	DESCRIPTIONS	
[31:23]	RX_FSM	RxDMA FSM
[22]	Reserved	-
[21:16]	RXBuf_FSM	Receive Buffer FSM
[15:12]	RXFetch_FSM	Receive Descriptor Fetch FSM
[11:8]	RXClose_FSM	Receive Descriptor Close FSM
[7:0]	RFF_FSM	RxFIFO Controller FSM

Transmit Finite State Machine Register (TXFSM)

The TXFSM shows the current value of the FSM (Finite State Machine) of TxDMA and TxFIFO controller. The TXFSM is read only and write to it has no effect. The TXFSM is used only for debug.

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
TXFSM	0xFFF0_3204	R	Transmit Finite State Machine Register	0x0101_1101

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

BITS	DESCRIPTIONS	
[31:24]	TX_FSM	TxDMA FSM
[23:22]	Reserved	-
[21:16]	TXBuf_FSM	Transmit Buffer FSM
[15:12]	TXFetch_FSM	Transmit Descriptor Fetch FSM
[11:8]	TXClose_FSM	Transmit Descriptor Close FSM
[7:5]	Reserved	-
[4:0]	TFF_FSM	TxFIFO Controller FSM

Finite State Machine Register 0 (FSM0)

The FSM0 shows the current value of the FSM (Finite State Machine) of the function module in EMC. The FSM0 is read only and write to it has no effect. The FSM0 is used only for debug.

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
FSM0	0xFFF0_3208	R	Finite State Machine Register 0	0x0001_0101

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

BITS	DESCRIPTIONS	
[31:26]	Reserved	-
[25:16]	TXMAC_FSM	TxMAC FSM
[15:14]	Reserved	-
[13:8]	TXDefer_FSM	Transmit Defer Process FSM
[7:0]	STA_FSM	MII Management I/F FSM

Finite State Machine Register 1 (FSM1)

The FSM1 shows the current value of the FSM (Finite State Machine) of the function module in EMC. The FSM1 is read only and write to it has no effect. The FSM1 is used only for debug.

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
FSM1	0xFFF0_320C	R	Finite State Machine Register 1	0x1100_0100

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

BITS	DESCRIPTIONS	
[31]	Reserved	-
[30:28]	ARB_FSM	Internal Arbiter FSM
[27:24]	TxPause_FSM	Transmit PAUSE Control Frame FSM
[23:14]	Reserved	-
[13:8]	AHB_FSM	[13:8]: AHB Master FSM
[7:0]	RESERVED	-

Debug Configuration Register (DCR)

The DCR is for debug only to multiplex different signal group out. In FPGA emulation, the signals are outputted to probe pins in emulation board. In real chip, the signals are outputted through the GPIO pins.

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
DCR	0xFFF0_3210	R/W	Debug Configuration Register	0x0000_003f

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

BITS	DESCRIPTIONS	
[31:24]	Reserved	-
[23:22]	Enable	The Function Enable outputs two function enable signals to external stimulus circuit. At this stage, only the bit 22 is used for external random collision generator. The random collision generator used only in FPGA emulation.
[21:8]	Reserved	-
[7:6]	Out	The Flag Out provides two output flags to trigger Logic Analyzer for debug. These two bits can be written at any time.
[5:0]	Config	The Configuration controls which group of internal signals can be multiplexed out for debug. Each group includes 16 signals.

CONFIG	SIGNALS	CONFIG	SIGNALS
6'h00	OUT [6], TransDone, GrantLost, Trans_CTR [4:0], LAST, TransCtrExpire, DMode_AHB_CS [5:0]	6'h01	OUT [6], DMode_TxBuf_CS [6:0] DMode_TXFSM_CS [7:0]
6'h02	OUT [6], DMode_RXBuf_CS [5:0], DMode_RXFSM_CS [8:0]	6'h03	OUT [6], TXFIFO_HT, TXFIFO_LT, DMode_TFF_CS [4:0], DMode_RFF_CS [7:0]
6'h04	TxBuf_DRDY, TFF_WPTR [5:0], TX_START, TXSTART, READ, TFF_RPTR [5:0]	6'h05	WRITE, RFF_WPTR [5:0], RXFIFO_HT, RXFIFO_LT, RxBuf_ACK, RFF_RPTR [5:0]

Continued.

CONFIG	SIGNALS	CONFIG	SIGNALS
6'h06	R0_PTLE, RxStart, SFD, WasSFD, RxFrame, WrByte, Rx_OvFlow, 1'b0, R0_RBC [7:0]	**6'h07**	R0_CRCE, RX_DV_In, SynStart, R0_DB, Rx_OvFlow, WRITECTR [2:0], RxByte [7:0]
6'h08	Reserved	**6'h09**	Reserved
6'h0A	OUT [7:6], RegMISTA_Rx_W, RXERR_sync, R0_CRCE, R0_PTLE, R0_RP, RegMISTA_Tx_W, T0_EXDEF, T0_TXABT, T0_CCNT [3:0], 2'b00	**6'h0B**	OUT [7:6], MCMDR_SDPZ_Clr, RegMCMDR_SDPZ_Clr, DMode_Pause_CS [3:0], MacCtlFra, PauseFra, PauseTx, MacCtlFra_sync, PauseFra_sync, PAUSE, Pause_en, FDUP
6'h0C	OUT [7:6], FrameWPtr [1:0], FrameRPtr [1:0], RFF_One, FrameWPtr_Inc, FrameRPtr_Inc, Rounding, NexPktStartPtr [5:0]	**6'h0D**	OUT [7:6], ARB_REQ_Set, ARB_REQ_Clr, DMode_ARB_CS [2:0], TransDone, GrantLost, TransCtrExpire, Trans_CTR [4:0], BURST
6'h0E	R0_CRCE, Rx_OvFlow, R0_MRE, CRCERR, DAMATCH, RxFrame, SFD, RxMIIErr, SynStart, Hi_Lo_Syn, New_DataValid, L_RxFrame, RxStart, DataValid, Hi_Lo, RX_DV_In	**6'h0F**	OUT [6], WRITE, RFF_WPTR [5:0], RxReuse, RxBuf_ACK, RFF_RPTR [5:0]
6'h10	WRITE, RFF_CS [7:1], RFF_WPTR [5:0], RXERR_sync, RxReuse	**6'h11**	OUT [6], TX_CLK, TX_EN, TXD [3:0], RX_CLK, RX_DV, RX_ER, RXD [3:0], CRS, COL
6'h12	OUT [6], TXSTART, TX_START, DMode_TFF_CS [4:0], TXSTART_Set, TXSTART_Clr, TXSTART_Re_Set, FrameWaiting, Deferring, COL, TXCOL, TXCOL_sync	**6'h13**	OUT [6], DMode_TxBuf_CS[6:0], DMode_TFF_CS[4:0], TXFIFO_UF, TXFIFO_HT, TXOK_sync
6'h14	OUT [6], READ, READ_sync, READ_Mask, ReadMask_sync, TFF_RPTR [5:0], DMode_TFF_CS [4:0]	**6'h15**	

Debug Mode MAC Information Register (DMMIR)

The DMMIR keeps the information of MAC module for debug.

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
DMMIR	0xFFF0_3214	R	Debug Mode MAC Information 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
RBC							
7	6	5	4	3	2	1	0
RBC							

BITS	DESCRIPTIONS	
[31:16]	Reserved	-
[15:0]	RBC	Receive Byte Count

BIST Mode Register (BISTR)

The BISTR controls the BIST (Built In Self Test) for embedded SRAM, 256B for RxFIFO and 256B for TxFIFO.

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
BISTR	0xFFF0_3300	R/W	BIST Mode 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				BistFail		Finish	BMEn

BITS	DESCRIPTIONS	
[31:5]	Reserved	-
[3:2]	BistFail	The **BIST Fail** indicates if the BIST test fails or succeeds. If the BistFail is low at the end, the embedded SRAM pass the BIST test, otherwise, it is faulty. The BistFail will be high once the BIST detects the error and remains high during the BIST operation. If BistFail[2] high indicates the embedded SRAM for TxFIFO BIST test failed. If BistFail[3] high indicates the embedded SRAM for RxFIFO BIST test failed. The BistFail is a write clear field. Write 1 to this field clears the content and write 0 has no effect.
[1]	Finish	The **BIST Operation Finish** indicates the end of the BIST operation. When BIST controller finishes all operations, this bit will be high. The Finish is a write clear field. Write 1 to this field clears the content and write 0 has no effect.
[0]	BMEn	The **BIST Mode Enable** is used to enable the BIST operation. If high enables the BIST controller to do embedded SRAM test. This bit is also used to do the reset for BIST circuit. It is necessary to reset the BIST circuit one clock cycle at least in order to initialize the BIST properly. The BMEn can be disabled by write 0.

6.6 GDMA Controller

The W90N745 has a two-channel general DMA controller, 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 nXDREQ. 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 W90N745 GDMA controller can increment source or destination address, decrement them as well, and conduct 8-bit (byte), 16-bit (half-word), or 32-bit (word) data transfers.

The GDMA includes the following features

- AMBA AHB compliant
- Supports 4-data burst mode to boost performance
- Provides support for external GDMA device
- Demand mode speeds up external GDMA operations

6.6.1 GDMA Functional Description

The GDMA directly transfers data between source and destination. The GDMA starts to transfer data after it receives service requests from nXDREQ signal or software. When the entire data have been transferred completely, the GDMA becomes idle. Nevertheless, if another transfer is needed, then the GDMA must be programmed again. There are three transfer modes:

Single Mode

Single mode requires a GDMA request for each data transfer. A GDMA request (nXDREQ or software) causes one byte, one half-word, or one word to transfer if the 4-data burst mode is disabled, or four times of transfer width is the 4-data burst mode is enabled.

Block Mode

The assertion of a single GDMA request causes all of the data to be transferred in a single operation. The GDMA transfer is completed when the current transfer count register reaches zero.

Demand Mode

The GDMA continues transferring data until the GDMA request input nXDREQ becomes inactive.

6.6.2 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
Channel 0				
GDMA_CTL0	0xFFF0_4000	R/W	Channel 0 Control Register	0x0000_0000
GDMA_SRCB0	0xFFF0_4004	R/W	Channel 0 Source Base Address Register	0x0000_0000
GDMA_DSTB0	0xFFF0_4008	R/W	Channel 0 Destination Base Address Register	0x0000_0000
GDMA_TCNT0	0xFFF0_400C	R/W	Channel 0 Transfer Count Register	0x0000_0000
GDMA_CSRC0	0xFFF0_4010	R	Channel 0 Current Source Address Register	0x0000_0000
GDMA_CDST0	0xFFF0_4014	R	Channel 0 Current Destination Address Register	0x0000_0000
GDMA_CTCNT0	0xFFF0_4018	R	Channel 0 Current Transfer Count Register	0x0000_0000
Channel 1				
GDMA_CTL1	0xFFF0_4020	R/W	Channel 1 Control Register	0x0000_0000
GDMA_SRCB1	0xFFF0_4024	R/W	Channel 1 Source Base Address Register	0x0000_0000
GDMA_DSTB1	0xFFF0_4028	R/W	Channel 1 Destination Base Address Register	0x0000_0000
GDMA_TCNT1	0xFFF0_402C	R/W	Channel 1 Transfer Count Register	0x0000_0000
GDMA_CSRC1	0xFFF0_4030	R	Channel 1 Current Source Address Register	0x0000_0000
GDMA_CDST1	0xFFF0_4034	R	Channel 1 Current Destination Address Register	0x0000_0000
GDMA_CTCNT1	0xFFF0_4038	R	Channel 1 Current Transfer Count Register	0x0000_0000

Channel 0/1 Control Register (GDMA_CTL0, GDMA_CTL1)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
GDMA_CTL0	0xFFF0_4000	R/W	Channel 0 Control Register	0x0000_0000
GDMA_CTL1	0xFFF0_4020	R/W	Channel 1 Control Register	0x0000_0000

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

BITS	DESCRIPTIONS	
[31]	RESERVED	-
[30:28]	TC_WIDTH	nRTC/nWTC active width selection, from 1 to 7 HCLK cycles.
[27:26]	REQ_SEL	**External request pin selection, if GDMAMS [3:2]=00, REQ_SEL will be don't care.** If REQ_SEL [27:26]=00, external request don't use. If REQ_SEL [27:26]=01, use nXDREQ. If REQ_SEL [27:26]=10, external request don't use. If REQ_SEL [27:26]=11, external request don't use.
[25]	REQ_ATV	**nXDREQ High/Low active selection** 1'b0 = nXDREQ is **LOW** active. 1'b1 = nXDREQ is **HIGH** active.
[24]	ACK_ATV	**nXDACK High/Low active selection** 1'b0 = nXDACK is **LOW** active. 1'b1 = nXDACK is **HIGH** active.
[23]	RW_TC	**Read/Write terminal count output selection.** 1'b0 = output to nRTC. 1'b1 = output to nWTC.
[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 The address boundary alignment should be depended on TWS [13:12]. 1'b0 = the GDMA_SRCB is on the boundary alignment. 1'b1 = 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 The address boundary alignment should be depended on TWS [13:12]. 1'b0 = the GDMA_DSTB is on the boundary alignment. 1'b1 = the GDMA_DSTB not on the boundary alignment The DABNDERR register bits just can be read only.
[20]	GDMATERR	**GDMA Transfer Error** 1'b0 = No error occurs 1'b1 = Hardware sets this bit on a GDMA transfer failure Transfer error will generate GDMA interrupt

Continued

BITS	DESCRIPTIONS	
[19]	AUTOIEN	**Auto initialization Enable** 1'b0 = Disables auto initialization 1'b1 = 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.
[18]	TC	**Terminal Count** 1'b0 = Channel does not expire 1'b1 = Channel expires; this bit is set only by GDMA hardware, and clear by software to write logic 0. TC [18] is the GDMA interrupt flag. TC [18] or GDMATERR[20] will generate interrupt
[17]	BLOCK	**Bus Lock** 1'b0 = Unlocks the bus during the period of transfer 1'b1 = 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).
[15]	DM	**Demand Mode** 1'b0 = Normal external GDMA mode 1'b1 = When this bit is set to 1, the external GDMA operation is speeded up. When external GDMA device is operating in the demand mode, the GDMA transfers data as long as the external GDMA request signal nXDREQ is active. The amount of data transferred depends on how long the nXDREQ is active. When the nXDREQ is active and GDMA gets the bus in Demand mode, DMA holds the system bus until the nXDREQ signal becomes non-active. Therefore, the period of the active nXDREQ signal should be carefully tuned such that the entire operation does not exceed an acceptable interval (for example, in a DRAM refresh operation).
[14]	Reserved	-

Continued

BITS	DESCRIPTIONS	
[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** 1'b0 = Selects single mode. It requires an external GDMA request for every incurring GDMA operation. 1'b1 = 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.
[10]	Reserved	-
[9]	BME	**Burst Mode Enable** 1'b0 = Disables the 4-data burst mode 1'b1 = Enables the 4-data burst mode FF there are 16 words to be transferred, and BME [9]=1, the GDMA_TCNT should be 0x04; However, if BME [9]=0, the GDMA_TCNT should be 0x10.
[8]	SIEN	**Stop Interrupt Enable** 1'b0 = Do not generate an interrupt when the GDMA operation is stopped 1'b1 = Interrupt is generated when the GDMA operation is stopped
[7]	SAFIX	**Source Address Fixed** 1'b0 = Source address is changed during the GDMA operation 1'b1 = 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** 1'b0 = Destination address is changed during the GDMA operation 1'b1 = 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** 1'b0 = Source address is incremented successively 1'b1 = Source address is decremented successively

Continued

BITS	DESCRIPTIONS	
[4]	DADIR	**Destination Address Direction** 1'b0 = Destination address is incremented successively 1'b1 = Destination address is decremented successively
[3:2]	GDMAMS	**GDMA Mode Select** 00 = Software mode (memory-to-memory) 01 = External nXDREQ mode for external device 10 = Reserved 11 = Reserved
[1]	Reserved	-
[0]	GDMAEN	**GDMA Enable** 1'b0 = Disables the GDMA operation 1'b1 = Enables the GDMA operation; this bit will be clear automatically when the transfer is complete on AUTOIEN [19] register bit is on Disable mode.

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

The GDMA channel starts reading its data from the source address as defined in this source base address register.

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
GDMA_SRCB0	0xFFF0_4004	R/W	Channel 0 Source Base Address Register	0x0000_0000
GDMA_SRCB1	0xFFF0_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

Channel 0/1 Destination Base Address Register (GDMA_DSTB0, DMA_DSTB1)

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

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.

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
GDMA_DSTB0	0xFFF0_4008	R/W	Channel 0 Destination Base Address Register	0x0000_0000
GDMA_DSTB1	0xFFF0_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

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

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
GDMA_TCNT0	0xFFF0_400C	R/W	Channel 0 Transfer Count Register	0x0000_0000
GDMA_TCNT1	0xFFF0_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	
[31:24]	Reserved	-
[23:0]	TFR_CNT	The TFR_CNT represents the required number of GDMA transfers. The maximum transfer count is 16M –1.

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

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
GDMA_CSRC0	0xFFF0_4010	R	Channel 0 Current Source Address Register	0x0000_0000
GDMA_CSRC1	0xFFF0_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	The 32-bit **Current Source Address** 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	0xFFF0_4014	R	Channel 0 Current Destination Address Register	0x0000_0000
GDMA_CDST1	0xFFF0_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	The 32-bit **Current Destination Address** 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)

The Current transfer count register indicates the number of transfer being performed.

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
GDMA_CTCNT0	0xFFF0_4018	R	Channel 0 Current Transfer Count Register	0x0000_0000
GDMA_CTCNT1	0xFFF0_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	
[31:24]	Reserved	-
[23:0]	CURRENT_TFR_CNT	Current Transfer Count register The current transfer count register indicates the number of transfer being performed

6.7 USB Host Controller

The **Universal Serial Bus (USB)** is a low-cost, low-to-mid-speed peripheral interface standard intended for modem, scanners, PDAs, keyboards, mice, and other devices that do not require a high-bandwidth parallel interface. 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 has the following features:

- **Open Host Controller Interface (OHCI)** Revision 1.1 compatible.
- USB Revision 1.1 compatible
- Supports both low-speed (1.5 Mbps) and full-speed (12Mbps) USB devices.
- Handles all the USB protocol.
- Built-in DMA for real-time data transfer
- Multiple low power modes for efficient power management

6.7.1 USB Host Functional Description

6.7.1.1. AHB Interface

The OpenHCI Host Controller is connected to the system by the AHB bus. The design requires both master and slave bus operations. As a master, the Host Controller is responsible for running cycles on the AHB bus to access EDs and TDs as well as transferring data between memory and the local data buffer. As a slave, the Host Controller monitors the cycles on the AHB bus and determines when to respond to these cycles. Configuration and non-real-time control access to the Host Controller operational registers are through the AHB bus slave interface.

6.7.1.2. Host Controller

List Processing

The List Processor manages the data structures from the Host Controller Driver and coordinates all activity within the Host Controller.

Frame Management

Frame Management is responsible for managing the frame specific tasks required by the USB specification and the OpenHCI specification. These tasks are:

1) Management of the OpenHCI frame specific Operational Registers
2) Operation of the Largest Data Packet Counter.
3) Performing frame qualifications on USB Transaction requests to the SIE.
4) Generate SOF token requests to the SIE.

Interrupt Processing

Interrupts are the communication method for HC-initiated communication with the Host Controller Driver. There are several events that may trigger an interrupt from the Host Controller. Each specific event sets a specific bit in the *HcInterruptStatus* register.

Host Controller Bus Master

The Host Controller Bus Master is the central block in the data path. The Host Controller Bus Master coordinates all access to the AHB Interface. There are two sources of bus mastering within Host Controller: the List Processor and the Data Buffer Engine.

Data Buffer

The Data Buffer serves as the data interface between the Bus Master and the SIE. It is a combination of a 64-byte latched based bi-directional asynchronous FIFO and a single Dword AHB Holding Register.

6.7.1.3. USB Interface

The USB interface includes the integrated Root Hub with two external ports, Port 1 and Port 2 as well as the Serial Interface Engine (SIE) and USB clock generator. The interface combines responsibility for executing bus transactions requested by the HC as well as the hub and port management specified by USB.

6.7.2 USB Host Controller Registers Map

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
OpenHCI Registers				
HcRevision	0xFFF0_5000	R	Host Controller Revision Register	0x0000_0010
HcControl	0xFFF0_5004	R/W	Host Controller Control Register	0x0000_0000
HcCommandStatus	0xFFF0_5008	R/W	Host Controller Command Status Register	0x0000_0000
HcInterruptStatus	0xFFF0_500C	R/W	Host Controller Interrupt Status Register	0x0000_0000
HcInterruptEnbale	0xFFF0_5010	R/W	Host Controller Interrupt Enable Register	0x0000_0000
HcInterruptDisbale	0xFFF0_5014	R/W	Host Controller Interrupt Disable Register	0x0000_0000
HcHCCA	0xFFF0_5018	R/W	Host Controller Communication Area Register	0x0000_0000
HcPeriodCurrentED	0xFFF0_501C	R/W	Host Controller Period Current ED Register	0x0000_0000
HcControlHeadED	0xFFF0_5020	R/W	Host Controller Control Head ED Register	0x0000_0000
HcControlCurrentED	0xFFF0_5024	R/W	Host Controller Control Current ED Register	0x0000_0000
HcBulkHeadEd	0xFFF0_5028	R/W	Host Controller Bulk Head ED Register	0x0000_0000
HcBulkCurrentED	0xFFF0_502C	R/W	Host Controller Bulk Current ED Register	0x0000_0000

Continued.

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
OpenHCI Registers				
HcDoneHeadED	0xFFF0_5030	R/W	Host Controller Done Head Register	0x0000_0000
HcFmInterval	0xFFF0_5034	R/W	Host Controller Frame Interval Register	0x0000_2EDF
HcFrameRemaining	0xFFF0_5038	R	Host Controller Frame Remaining Register	0x0000_0000
HcFmNumber	0xFFF0_503C	R	Host Controller Frame Number Register	0x0000_0000
HcPeriodicStart	0xFFF0_5040	R/W	Host Controller Periodic Start Register	0x0000_0000
HcLSThreshold	0xFFF0_5044	R/W	Host Controller Low Speed Threshold Register	0x0000_0628
HcRhDescriptorA	0xFFF0_5048	R/W	Host Controller Root Hub Descriptor A Register	0x0100_0002
HcRhDescriptorB	0xFFF0_504C	R/W	Host Controller Root Hub Descriptor B Register	0x0000_0000
HcRhStatus	0xFFF0_5050	R/W	Host Controller Root Hub Status Register	0x0000_0000
HcRhPortStatus [1]	0xFFF0_5054	R/W	Host Controller Root Hub Port Status [1]	0x0000_0000
HcRhPortStatus [2]	0xFFF0_5058	R/W	Host Controller Root Hub Port Status [2]	0x0000_0000
USB Configuration Registers				
TestModeEnable	0xFFF0_5200	R/W	USB Test Mode Enable Register	0x0XXX_XXXX
OperationalModeEnable	0xFFF0_5204	R/W	USB Operational Mode Enable Register	0x0000_0000

Host Controller Revision Register

REGISTER	OFFSET ADDRESS	R/W	DESCRIPTION	RESET VALUE
HcRevision	0xFFF0_5000	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
Revision							

BITS		DESCRIPTION
[31:8]	Reserved	**Reserved.** Read/Write 0's
[7:0]	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

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
HcControl	0xFFF0_5004	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					RWCE	RWC	IR
7	6	5	4	3	2	1	0
HCFS		BLE	CLE	ISE	PLE	CBR	

BITS	DESCRIPTION	
[31:11]	Reserved	**Reserved.** Read/Write 0's
[10]	RWCE	**RemoteWakeupConnectedEnable** 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]	RWC	**RemoteWakeupConnected** 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]	INR	**InterruptRouting** This bit is used for interrupt routing: 0: Interrupts routed to normal interrupt mechanism (INT). 1: Interrupts routed to SMI.
[7:6]	HCFS	**HostControllerFunctionalState** This field sets the Host Controller state. The Controller may force a state change from USB SUSPEND to USB RESUME after detecting resume signaling from a downstream port. States are: 00: USB RESET 01: USBRESUME 10: USBOPERATIONAL 11: USBSUSPEND
[5]	BLE	**BulkListEnable** When set this bit enables processing of the Bulk list.
[4]	CLE	**Control Listenable** When set this bit enables processing of the Control list.
[3]	ISE	**Isochronous 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]	PLE	**Periodic Listenable** 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]	CBR	**ControlBulkServiceRatio** 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

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
HcCommandStatus	0xFFF0_5008	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							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved				OCR	BLF	CLF	HCR

BITS	DESCRIPTION	
[31:18]	Reserved	**Reserved**
[17:16]	SOC	**ScheduleOverrunCount** This field is increment every time the **SchedulingOverrun** bit in *HcInterruptStatus* is set. The count wraps from '11' to '00.'
[15:4]	Reserved	**Reserved.** Read/Write 0's
[3]	OCR	**OwnershipChangeRequest** When set by software, this bit sets the **OwnershipChange** field in *HcInterruptStatus*. The bit is cleared by software.
[2]	BLF	**BulkListFilled** 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]	CLF	**ControlListFilled** 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]	HCR	**HostControllerReset** 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

All bits are set by hardware and cleared by software.

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
HcInterruptStatus	0xFFF0_500C	R/W	Host Controller Interrupt Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved	OCH	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	FNO	URE	RDT	SOF	WDH	SCO

BITS	DESCRIPTION	
[31]	Reserved	**Reserved**
[30]	OCH	**OwnershipChange** This bit is set when the **OwnershipChangeRequest** bit of *HcCommandStatus* is set.
[29:7]		Reserved
[6]	RHSC	**RootHubStatusChange** This bit is set when the content of *HcRhStatus* or the content of any *HcRhPortStatus* register has changed.
[5]	FNO	**FrameNumberOverflow** Set when bit 15 of **FrameNumber** changes value.
[4]	URE	**UnrecoverableError** This event is not implemented and is hard-coded to '0.' Writes are ignored.
[3]	RDT	**ResumeDetected** Set when Host Controller detects resume signaling on a downstream port.
[2]	SOF	**StartOfFrame** Set when the Frame Management block signals a 'Start of Frame' event.

Continued.

BITS	DESCRIPTION	
[1]	WDH	**WritebackDoneHead** Set after the Host Controller has written *HcDoneHead* to *HccaDoneHead*.
[0]	SCHO	**SchedulingOverrun** Set when the List Processor determines a Schedule Overrun has occurred.

Host Controller Interrupt Enable Register

Writing a '1' to a bit in this register sets the corresponding bit, while writing a '0' leaves the bit unchanged.

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
HcInterruptEnable	0xFFF0_5010	R/W	Host Controller Interrupt Enable Register	0x0000_0000

31	30	29	28	27	26	25	24
MIE	OCE	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	RHCE	FNOE	UREE	RDTE	SOFE	WDHE	SCHOE

BITS	DESCRIPTION	
[31]	MIE	**MasterInterruptEnable** This bit is a global interrupt enable. A write of '1' allows interrupts to be enabled via the specific enable bits listed above.
[30]	OCE	**OwnershipChangeEnable** 0: Ignore 1: Enable interrupt generation due to Ownership Change.
[29:7]	Reserved	**Reserved.** Read/Write 0's
[6]	RHSCE	**RootHubStatusChangeEnable** 0: Ignore 1: Enable interrupt generation due to Root Hub Status Change.

Continued.

BITS		DESCRIPTION
[5]	FNOE	**FrameNumberOverflowEnable** 0: Ignore 1: Enable interrupt generation due to Frame Number Overflow.
[4]	UREE	**UnrecoverableErrorEnable** This event is not implemented. All writes to this bit are ignored.
[3]	RDTE	**ResumeDetectedEnable** 0: Ignore 1: Enable interrupt generation due to Resume Detected.
[2]	SOFE	**StartOfFrameEnable** 0: Ignore 1: Enable interrupt generation due to Start of Frame.
[1]	WDHE	**WritebackDoneHeadEnable** 0: Ignore 1: Enable interrupt generation due to Write-back Done Head.
[0]	SCHOE	**SchedulingOverrunEnable** 0: Ignore 1: Enable interrupt generation due to Scheduling Overrun.

Host Controller Interrupt Disable Register

Writing a '1' to a bit in this register clears the corresponding bit, while writing a '0' to a bit leaves the bit unchanged.

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
HcInterruptEnable	0xFFF0_5014	R/W	Host Controller Interrupt Disable Register	0x0000_0000

31	30	29	28	27	26	25	24
MIE	OCE	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	RHSCE	FNOE	UREE	RDTE	SOFE	WDHE	SCHOE

BITS	DESCRIPTION	
[31]	MIE	**MasterInterruptEnable** Global interrupt disable. A write of '1' disables all interrupts.
[30]	OCE	**OwnershipChangeEnable** 0: Ignore 1: Disable interrupt generation due to Ownership Change.
[29:7]	Reserved	**Reserved. Read/Write 0's**
[6]	RHSCE	**RootHubStatusChangeEnable** 0: Ignore 1: Disable interrupt generation due to Root Hub Status Change.
[5]	FNOE	**FrameNumberOverflowEnable** 0: Ignore 1: Disable interrupt generation due to Frame Number Overflow.
[4]	UREE	**UnrecoverableErrorEnable** This event is not implemented. All writes to this bit will be ignored.
[3]	RDTE	**ResumeDetectedEnable** 0: Ignore 1: Disable interrupt generation due to Resume Detected.
[2]	SOFE	**StartOfFrameEnable** 0: Ignore 1: Disable interrupt generation due to Start of Frame.
[1]	WDHE	**WritebackDoneHeadEnable** 0: Ignore 1: Disable interrupt generation due to Write-back Done Head.
[0]	SCHOE	**SchedulingOverrunEnable** 0: Ignore 1: Disable interrupt generation due to Scheduling Overrun.

Host Controller Communication Area Register

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
HcHCCA	0xFFF0_5018	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	DESCRIPTION	
[31:8]	HCCA	**HCCA** Pointer to HCCA base address.
[7:0]	Reserved	**Reserved**

Host Controller Period Current ED Register

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
HcPeriodCurretED	0xFFF0_501C	R/W	Host Controller Period Current ED Register	0x0000_0000

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

BITS	DESCRIPTION	
[31:4]	PCED	**PeriodCurrentED**. Pointer to the current Periodic List ED.
[3:0]	Reserved	**Reserved.** Read/Write 0's

Host Controller Control Head ED Register

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
HcControlHeadED	0xFFF0_5020	R/W	Host Controller Control Head ED Register	0x0000_0000

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

BITS	DESCRIPTION	
[31:4]	CHED	**ControlHeadED** Pointer to the Control List Head ED.
[3:0]	Reserved	**Reserved**

Host Controller Control Current ED Register

REGISTER	OFFSET ADDRESS	R/W	DESCRIPTION	RESET VALUE
HcControlCurrentED	0xFFF0_5024	R/W	Host Controller Control Current ED Register	0x0000_0000

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

BITS	DESCRIPTION	
[31:4]	CCED	**ControlCurrentED** Pointer to the current Control List ED.
[3:0]	Reserved	**Reserved.** Read/Write 0's

Host Controller Bulk Head ED Register

REGISTER	OFFSET ADDRESS	R/W	DESCRIPTION	RESET VALUE
HcBulkHEADED	0xFFF0_5028	R/W	Host Controller Bulk Head ED Register	0x0000_0000

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

BITS	DESCRIPTION	
[31:4]	BHED	**BulkHeadED**. Pointer to the Bulk List Head ED.
[3:0]	Reserved	**Reserved.** Read/Write 0's

Host Controller Bulk Current ED Register

REGISTER	OFFSET ADDRESS	R/W	DESCRIPTION	RESET VALUE
HcBulkCurrentED	0xFFF0_502C	R/W	Host Controller Bulk Current ED Register	0x0000_0000

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

BITS	DESCRIPTION	
[31:4]	BCED	**BulkCurrentED**. Pointer to the current Bulk List ED.
[3:0]	Reserved	**Reserved.** Read/Write 0's

Host Controller Done Head Register

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
HcDoneHead	0xFFF0_5030	R/W	Host Controller Done Head Register	0x0000_0000

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

BITS	DESCRIPTION	
[31:4]	DOHD	**DoneHead**. Pointer to the current Done List Head ED.
[3:0]	Reserved	**Reserved.** Read/Write 0's

Host Controller Frame Interval Register

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
HcFmInterval	0xFFF0_5034	R/W	Host Controller Frame Interval Register	0x0000_2EDF

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

BITS	DESCRIPTION	
31	FINTVT	**FrameIntervalToggle** This bit is toggled by HCD when it loads a new value into **Frame Interval**.
[30:16]	FSLDP	**FSLargestDataPacket** This field specifies a value that is loaded into the Largest Data Packet Counter at the beginning of each frame.
[15:14]	Reserved	**Reserved**. Read/Write 0's
[13:0]	FINTV	**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

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
HcFmInterval	0xFFF0_5038	R	Host Controller Frame Remaining Register	0x0000_0000

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

BITS		DESCRIPTION
[31]	FRMT	**FrameRemainingToggle** Loaded with **FrameIntervalToggle** when **Frame Remaining** is loaded.
[30:14]	Reserved	**Reserved.** Read/Write 0's
[13:0]	FRM	**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 **Frame Interval**. In addition, the counter loads when the Host Controller transitions into USBOPERATIONAL.

Host Controller Frame Number Register

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
HcFmNumber	0xFFF0_503C	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
FRMN							
7	6	5	4	3	2	1	0
FRMN							

BITS	DESCRIPTION	
[31:16]	Reserved	**Reserved.** Read/Write 0's
[15:0]	FRMN	**FrameNumber** This 16-bit incrementing counter field is incremented coincident with the loading of **FrameRemaining**. The count rolls over from '000Fh' to '0h.'

Host Controller Periodic Start Register

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
HcPeriodicStart	0xFFF0_5040	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
		PERST					
7	6	5	4	3	2	1	0
PERST							

BITS		DESCRIPTION
[31:14]	Reserved	**Reserved**. Read/Write 0's
[13:0]	PERST	**PeriodicStart** This field contains a value used by the List Processor to determine where in a frame the Periodic List processing must begin.

Host Controller Low Speed Threshold Register

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
HcLSThreshold	0xFFF0_5044	R/W	Host Controller Low Speed Threshold Register	0x0000_0628

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				LsThreshold			
7	6	5	4	3	2	1	0
LsTreshold							

BITS		DESCRIPTION
[31:12]	Reserved	Rsvd. Read/Write 0's
[11:0]	LsTreshold	**LSThreshold** This field contains a value used by the Frame Management block to determine whether or not a low speed transaction can be started in the current frame.

Host Controller Root Hub Descriptor A Register

This register is only reset by a power-on reset. It is written during system initialization to configure the Root Hub. This bit should not be written during normal operation.

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
HcRhDescriptorA	0xFFF0_5048	R/W	Host Controller Root Hub Descriptor A Register	0x0100.0002

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

BITS		DESCRIPTION
[31:24]	POTPGT	**PowerOnToPowerGoodTime** 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] is 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.
[23:13]	Reserved	**Reserved.** Read/Write 0's
[12]	NOCP	**NoOverCurrentProtection** Global over-current reporting implemented in HYDRA-2. 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	**OverCurrentProtectionMode** Global over-current reporting implemented in HYDRA-2. This bit should be written 0 and is only valid when NoOverCurrentProtection is cleared. 0 = Global Over-Current 1 = Individual Over-Current
[10]	DEVT	**DeviceType** table of none-4is not a compound device.
[9]	NPSW	**NoPowerSwitching** Global power switching implemented in HYDRA-2. 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]	PSWM	**PowerSwitchingMode** Global power switching mode implemented in HYDRA-2. This bit is only valid when NoPowerSwitching is cleared. This bit should be written '0'. 0 = Global Switching 1 = Individual Switching
[7:0]	NDSP	**NumberDownstreamPorts** table of none-4 supports two downstream ports.

Host Controller Root Hub Descriptor B Register

This register is only reset by a power-on reset. It is written during system initialization to configure the Root Hub. These bits should not be written during normal operation.

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
HcRhDescriptorB	0xFFF0_504C	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
DEVRM							
7	6	5	4	3	2	1	0
DEVRM							

BITS	DESCRIPTION	
[31:16]	PPCM	**PortPowerControlMask** 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]	DEVRM	**DeviceRemoveable** table of none-4 ports default to removable devices. 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

This register is reset by the USBRESET state.

REGISTER	OFFSET ADDRESS	R/W	DESCRIPTION	RESET VALUE
HcRhStstus	0xFFF0_5050	R/W	Host Controller Root Hub Status Register	0x0000_0000

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

BITS	DESCRIPTION	
[31]	CRWE	(Write) **ClearRemoteWakeupEnable** Writing a '1' to this bit clears **DeviceRemoteWakeupEnable**. Writing a '1' has no effect.
[30:18]	Reserved	**Reserved.** Read/Write 0's
[17]	OVIC	**OverCurrentIndicatorChange** 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]	DRWE	**(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.
[14:2]	Reserved	**Reserved.** Read/Write 0's

Continued.

BITS		DESCRIPTION
[1]	OVRCI	**OverCurrentIndicator** 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]	LOPS	(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 [1][2]

This register is reset by the USBRESET state.

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
HcRhPortStatus [1]	0xFFF0_5054	R/W	Host Controller Root Hub Port Status [1]	0x0000_0000
HcRhPortStatus [2]	0xFFF0_5058	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						LSDA	PPS
7	6	5	4	3	2	1	0
Reserved			SPR	CPS	SPS	SPE	DRM

BITS		DESCRIPTION
[31:21]	Reserved	**Reserved.** Read/Write 0's
[20]	PRSC	**PortResetStatusChange** This bit indicates that the port reset signal has completed. 0 = Port reset is not complete. 1 = Port reset is complete.

Continued.

BITS		DESCRIPTION
[19]	POCIC	**PortOverCurrentIndicatorChange** This bit is set when **OverCurrentIndicator** changes. Writing a '1' clears this bit. Writing a '0' has no effect.
[18]	PSSC	**PortSuspendStatusChange** 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	**PortEnableStatusChange** 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	**ConnectStatusChange** This bit indicates a 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'.
[15:10]	Reserved	**Reserved.** Read/Write 0's
[9]	LSDA	(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.
[7:5]	Reserved	**Reserved.** Read/Write 0's

Continued.

BITS		DESCRIPTION
[4]	SPR	(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]	CPS	(Read) **PortOverCurrentIndicator** table of none-2 supports global over-current reporting. 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]	SPS	(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]	SPE	(Read) **PortEnableStatus** 0 = Port disabled. 1 = Port enabled. (Write) **SetPortEnable** Writing a '1' sets **PortEnableStatus**. Writing a '0' has no effect.
[0]	DRM	(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

This register selects which operational mode is enabled. Bits defined as write-only are read as 0's.

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
OperationalModeEnable	0xFFF0_5204	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							SIEPD
7	6	5	4	3	2	1	0
Reserved				OVRCUR	Reserved		DBREG

BITS		BIT DESCRIPTION
[31:9]	Reserved	**Reserved**. Read/write 0
[8]	SIEPD	**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.
[7:4]	Reserved	**Reserved**. Read/write 0
[3]	OVRCURP	**OVRCURP (over current indicator polarity)** When the OVRCURP bit is clear, the OVRCUR **non-inverted** to input into USB host controller. In contrast, when the OVRCURP bit is set, the OVRCUR **inverted** to input into USB host controller.
[2:1]	Reserved	**Reserved**. Read/write 0
[0]	DBREG	**Data Buffer Region 16** When set, the size of the data buffer region is 16 bytes. Otherwise, the size is 32 bytes.

6.8 USB Device Controller

The USB controller interfaces the AHB bus and the USB bus. The USB controller contains both the AHB master interface and AHB slave interface. CPU programs the USB controller through the AHB slave interface. For IN or OUT transfer, the USB controller needs to write data to memory or read data from memory through the AHB master interface. The USB controller also contains the USB transceiver to interface the USB.

6.8.1 USB Endpoints

It consists of four endpoints, designated EP0, EPA, EPB and EPC. Each is intended for a particular use as described below:

EP0: the default endpoint uses control transfer (In/Out) to handle configuration and control functions required by the USB specification. Maximum packed size is 16 bytes.

EPA: designed as a general endpoint. This endpoint could be programmed to be an Interrupt IN endpoint or an Isochronous IN endpoint or a Bulk In endpoint or Bulk OUT endpoint.

EPB: designed as a general endpoint. This endpoint could be programmed to be an Interrupt IN endpoint or an Isochronous IN endpoint or a Bulk In endpoint or Bulk OUT endpoint.

EPC: designed as a general endpoint. This endpoint could be programmed to be an Interrupt IN endpoint or an Isochronous IN endpoint or a Bulk In endpoint or Bulk OUT endpoint.

6.8.2 Standard Device Request

The USB controller has built-in hard-wired state machine to automatically respond to USB standard device request. It also supports to detect the class and vendor requests. For Get Descriptor request and Class or Vendor command, the firmware will control these procedures.

6.8.3 USB Device Register Description

USB Control Register (USB_CTL)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
USB_CTL	0xFFF0_6000	R/W	USB 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
CCMD	VCMD	SIE_RCV	SUS_TST	RWU_EN	SUSP	USB_RST	USB_EN

BITS	DESCRIPTIONS	
[31:8]		Reserved
[7]	CCMD	USB Class Command Decode Control Enable 0: Disable, the H/W circuit doesn't need to decode USB class command. It will return a stall status when it received a USB Class Command. 1: Enable, the H/W circuit decodes USB class command. It will assert an interrupt event when it received a USB Class Command.
[6]	VCMD	USB Vendor Command Decode Enable 0: Disable, the H/W circuit doesn't need to decode USB vendor command. It will return a stall status when it received a USB Vendor Command. 1: Enable, the H/W circuit decodes USB vendor command. It will assert an interrupt event when it received a USB Vendor Command.
[5]	SIE_RCV	USB SIE Differential RCV Source 0: RCV generated by the SIE 1: RCV generated by the USB transceiver
[4]	SUS_TST	USB Suspend Accelerate Test 0: Normal Operation 1: USB Suspend Accelerate Test (Only for Test)
[3]	RWU_EN	USB Remote Wake-up Enable 0: Disable USB Remote Wake-Up Detect 1: Enable USB Remote Wake-Up Detect
[2]	SUSP	USB Suspend Detect Enable 0: Disable USB Suspend Detect 1: Enable USB Suspend Detect
[1]	USB_RST	USB Engine Reset 0: Normal operation 1: Reset USB Engine
[0]	USB_EN	USB Engine Enable 0: disable USB Engine 1: enable USB Engine **Note: set this bit to "0", the device is absent from host. After set this bit to "1", the host will detect a device attached.**

USB Class or Vendor command Register (USB_CVCMD)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
USB_CVCMD	**0xFFF0_6004**	R/W	USB class or vendor 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			CVI_LG				

BITS	DESCRIPTIONS	
[31:5]		Reserved
[4:0]	CVI_LG	Byte Length for Class and Vendor Command and Get Descriptor Return Data Packet

USB Interrupt Enable Register (USB_IE)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
USB_IE	0xFFF0_6008	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
RUM_CLKI	RST_ENDI	USB_CGI	USB_BTI	CVSI	CDII	CDOI	VENI
7	6	5	4	3	2	1	0
CLAI	GSTRI	GCFGI	GDEVI	ERRI	RUMI	SUSI	RSTI

BITS	DESCRIPTIONS	
[31:16]		Reserved
[15]	RUM_CLKI	Interrupt enable for RESUME (for clock is stopped) 0: Disable 1: Enable
[14]	RST_ENDI	Interrupt enable for USB reset end 0: Disable 1: Enable
[13]	USB_CGI	Interrupt Enable for Device Configured 0: Disable 1: Enable **Note: the interrupt occurs when device configured or dis-configured.**
[12]	USB_BTI	Interrupt Enable for USB Bus Transition 0: Disable 1: Enable
[11]	CVSI	Interrupt Enable Control for Status Phase of Class or Vendor Command 0: Disable 1: Enable
[10]	CDII	Interrupt Enable Control for Data-In of Class or Vendor Command 0: Disable 1: Enable
[9]	CDOI	Interrupt Enable Control for Data-Out of Class or Vendor Command 0: Disable 1: Enable
[8]	VENI	Interrupt Enable Control for USB Vendor Command 0: Disable 1: Enable
[7]	CLAI	Interrupt Enable Control for USB Class Command 0: Disable 1: Enable
[6]	GSTRI	Interrupt Enable Control for USB Get_String_Descriptor Command 0: Disable 1: Enable
[5]	GCFGI	Interrupt Enable Control for USB Get_Configuration_Descriptor Command 0: Disable 1: Enable

Continued.

BITS	DESCRIPTIONS	
[4]	GDEVI	Interrupt Enable Control for USB Get_Device_Descriptor Command 0: Disable 1: Enable
[3]	ERRI	Interrupt Enable Control for USB Error Detect 0: Disable 1: Enable
[2]	RUMI	Interrupt Enable Control for USB Resume Detect 0: Disable 1: Enable
[1]	SUSI	Interrupt Enable Control for USB Suspend Detect 0: Disable 1: Enable
[0]	RSTI	Interrupt Enable Control for USB Reset Command Detect 0: Disable 1: Enable

USB Interrupt status Register (USB_IS)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
USB_IS	0xFFF6_000C	R	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
RUM_CLKS	RSTENDS	USB_CGS	USB_BTS	CVSS	CDIS	CDOS	VENS
7	6	5	4	3	2	1	0
CLAS	GSTRS	GCFGS	GDEVS	ERRS	RUMS	SUSS	RSTS

BITS	DESCRIPTIONS	
[31:16]		Reserved
[15]	RUM_CLKS	Interrupt status for RESUME (for clock is stopped) 0: No Interrupt Generated 1: Interrupt Generated
[14]	RSTENDS	Interrupt status for USB reset end 0: No Interrupt Generated 1: Interrupt Generated
[13]	USB_CGS	Interrupt Status for USB Device Configured 0: No Interrupt Generated 1: Interrupt Generated(configured and dis-configured)
[12]	USB_BTS	Interrupt Status for USB Bus Transition 0: No Interrupt Generated 1: Interrupt Generated
[11]	CVSS	Interrupt Status for Status Phase of Class or Vendor Command 0: No Interrupt Generated 1: Interrupt Generated
[10]	CDIS	Interrupt Status for Data-In of Class or Vendor Command 0: No Interrupt Generated 1: Interrupt Generated
[9]	CDOS	Interrupt Status for Data-Out of Class or Vendor Command 0: No Interrupt Generated 1: Interrupt Generated
[8]	VENS	Interrupt Status for USB Vendor Command 0: No Interrupt Generated 1: Interrupt Generated
[7]	CLAS	Interrupt Status for USB Class Command 0: No Interrupt Generated 1: Interrupt Generated

Continued.

BITS	DESCRIPTIONS	
[6]	GSTRS	Interrupt Status for USB Get_String_Descriptor Command 0: No Interrupt Generated 1: Interrupt Generated
[5]	GCFGS	Interrupt Status for USB Get_Configuration_Descriptor Command 0: No Interrupt Generated 1: Interrupt Generated
[4]	GDEVS	Interrupt Status for USB Get_Device_Descriptor Command 0: No Interrupt Generated 1: Interrupt Generated
[3]	ERRS	Interrupt Status for USB Error Detect 0: No Interrupt Generated 1: Interrupt Generated
[2]	RUMS	Interrupt Status for USB Resume Detect 0: No Interrupt Generated 1: Interrupt Generated
[1]	SUSS	Interrupt Status for USB Suspend Detect 0: No Interrupt Generated 1: Interrupt Generated
[0]	RSTS	Interrupt Status for USB Reset Command Detect 0: No Interrupt Generated 1: Interrupt Generated

USB Interrupt Status Clear (USB_IC)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
USB_IC	0xFFF6_0010	R/W	USB interrupt 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
RUM_CLKC	RSTENDC	USB_CGC	USB_BTC	CVSC	CDIC	CDOC	VENC
7	6	5	4	3	2	1	0
CLAC	GSTRC	GCFGC	GDEVC	ERRC	RUMC	SUSC	RSTC

BITS	DESCRIPTIONS	
[31:16]		Reserved
[15]	RUM_CLKC	Interrupt status clear for RESUME (for clock is stopped) 0: NO Operation 1: Clear Interrupt Status
[14]	RSTENDC	Interrupt status clear for USB reset end 0: NO Operation 1: Clear Interrupt Status
[13]	USB_CGC	Interrupt Status Clear for USB Device Configured 0: NO Operation 1: Clear Interrupt Status
[12]	USB_BTC	Interrupt Status Clear for USB Bus Transition 0: NO Operation 1: Clear Interrupt Status
[11]	CVSC	Interrupt Status Clear for Status Phase of Class or Vendor Command 0: NO Operation 1: Clear Interrupt Status
[10]	CDIC	Interrupt Status Clear for Data-In of Class or Vendor Command 0: NO Operation 1: Clear Interrupt Status
[9]	CDOC	Interrupt Status Clear for Data-Out of Class or Vendor Command 0: NO Operation 1: Clear Interrupt Status

Continued.

BITS		DESCRIPTIONS
[8]	VENC	Interrupt Status Clear for USB Vendor Command 0: NO Operation 1: Clear Interrupt Status
[7]	CLAC	Interrupt Status Clear for USB Class Command 0: NO Operation 1: Clear Interrupt Status
[6]	GSTRC	Interrupt Status Clear for USB Get_String_Descriptor Command 0: NO Operation 1: Clear Interrupt Status
[5]	GCFGC	Interrupt Status Clear for USB Get_Configuration_Descriptor Command 0: NO Operation 1: Clear Interrupt Status
[4]	GDEVC	Interrupt Status Clear for USB Get_Device_Descriptor Command 0: NO Operation 1: Clear Interrupt Status
[3]	ERRC	Interrupt Status Clear for USB Error Detect 0: NO Operation 1: Clear Interrupt Status
[2]	RUMC	Interrupt Status Clear for USB Resume Detect 0: NO Operation 1: Clear Interrupt Status
[1]	SUSC	Interrupt Status Clear for USB Suspend Detect 0: NO Operation 1: Clear Interrupt Status
[0]	RSTC	Interrupt Status Clear for USB Reset Command Detect 0: NO Operation 1: Clear Interrupt Status

USB Interface and String Register (USB_IFSTR)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
USB_IFSTR	0xFFF06014	R/W	USB interface and string 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						STR6_EN	STR5_EN
7	6	5	4	3	2	1	0
STR4_EN	STR3_EN	STR2_EN	STR1_EN	INF4_EN	INF3_EN	INF2_EN	INF1_EN

BITS	DESCRIPTIONS	
[31:10]		Reserved
[9]	STR6_EN	USB String Descriptor-6 Control 0: Disable 1: Enable
[8]	STR5_EN	USB String Descriptor-5 Control 0: Disable 1: Enable
[7]	STR4_EN	USB String Descriptor-4 Control 0: Disable 1: Enable
[6]	STR3_EN	USB String Descriptor-3 Control 0: Disable 1: Enable
[5]	STR2_EN	USB String Descriptor-2 Control 0: Disable 1: Enable
[4]	STR1_EN	USB String Descriptor-1 Control 0: Disable 1: Enable

Continued.

BITS	DESCRIPTIONS	
[3]	INF4_EN	USB Interface-4 Control 0: Disable 1: Enable
[2]	INF3_EN	USB Interface-3 Control 0: Disable 1: Enable
[1]	INF2_EN	USB Interface-2 Control 0: Disable 1: Enable
[0]	INF1_EN	USB Interface-1 Control 0: Disable 1: Enable

USB Control transfer-out port 0 (USB_ODATA0)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
USB_ODATA0	0xFFF06018	R	USB control transfer-out port 0 register	0x0000_0000

31	30	29	28	27	26	25	24
ODATA0							
23	22	21	20	19	18	17	16
ODATA0							
15	14	13	12	11	10	9	8
ODATA0							
7	6	5	4	3	2	1	0
ODATA0							

BITS	DESCRIPTIONS	
[31:0]	ODATA0	Control Transfer-out data 0

USB Control transfer-out port 1 (USB_ODATA1)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
USB_ODATA1	0xFFF0601C	R	USB control transfer-out port 1 register	0x0000_0000

31	30	29	28	27	26	25	24
ODATA1							
23	22	21	20	19	18	17	16
ODATA1							
15	14	13	12	11	10	9	8
ODATA1							
7	6	5	4	3	2	1	0
ODATA1							

BITS	DESCRIPTIONS	
[31:0]	ODATA1	Control Transfer-out data 1

USB Control transfer-out port 2 (USB_ODATA2)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
USB_ODATA2	0xFFF06020	R	USB control transfer-out port 2 register	0x0000_0000

31	30	29	28	27	26	25	24
ODATA2							
23	22	21	20	19	18	17	16
ODATA2							
15	14	13	12	11	10	9	8
ODATA2							
7	6	5	4	3	2	1	0
ODATA2							

BITS	DESCRIPTIONS	
[31:0]	ODATA2	Control Transfer-out data 2

USB Control transfer-out port 3 (USB_ODATA3)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
USB_ODATA3	0xFFF06024	R	USB control transfer-out port 3 register	0x0000_0000

31	30	29	28	27	26	25	24
ODATA3							
23	22	21	20	19	18	17	16
ODATA3							
15	14	13	12	11	10	9	8
ODATA3							
7	6	5	4	3	2	1	0
ODATA3							

BITS	DESCRIPTIONS	
[31:0]	ODATA3	Control Transfer-out data 3

USB Control transfer-in data port0 Register (USB_IDATA0)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
USB_IDATA0	0xFFF06028	R/W	USB transfer-in data port0 register	0x0000_0000

31	30	29	28	27	26	25	24
IDATA0							
23	22	21	20	19	18	17	16
IDATA0							
15	14	13	12	11	10	9	8
IDATA0							
7	6	5	4	3	2	1	0
IDATA0							

BITS	DESCRIPTIONS	
[31:6]	IDATA0	Control transfer-in data0

USB Control transfer-in data port 1 Register (USB_IDATA1)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
USB_IDATA1	0xFFF0602C	R/W	USB control transfer-in data port 1	0x0000_0000

31	30	29	28	27	26	25	24
IDATA1							
23	22	21	20	19	18	17	16
IDATA1							
15	14	13	12	11	10	9	8
IDATA1							
7	6	5	4	3	2	1	0
IDATA1							

BITS	DESCRIPTIONS	
[31:6]	IDATA1	Control transfer-in data1

USB Control transfer-in data port 2 Register (USB_IDATA2)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
USB_IDATA2	0xFFF06030	R/W	USB control transfer-in data port 2	0x0000_0000

31	30	29	28	27	26	25	24
IDATA2							
23	22	21	20	19	18	17	16
IDATA2							
15	14	13	12	11	10	9	8
IDATA2							
7	6	5	4	3	2	1	0
IDATA2							

BITS	DESCRIPTIONS	
[31:6]	IDATA2	Control transfer-in data2

USB Control transfer-in data port 3 Register (USB_IDATA3)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
USB_IDATA3	0xFFF06034	R/W	USB control transfer-in data port 3	0x0000_0000

31	30	29	28	27	26	25	24
IDATA3							
23	22	21	20	19	18	17	16
IDATA3							
15	14	13	12	11	10	9	8
IDATA3							
7	6	5	4	3	2	1	0
IDATA3							

BITS	DESCRIPTIONS	
[31:6]	IDATA3	Control transfer-in data3

USB SIE Status Register (USB_SIE)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
USB_SIE	0xFFF06038	R	USB SIE 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						USB_DPS	USB_DMS

BITS	DESCRIPTIONS	
[31:2]		Reserved
[1]	USB_DPS	USB Bus D+ Signal Status 0: USB Bus D+ Signal is low 1: USB Bus D+ Signal is high
[0]	USB_DMS	USB Bus D- Signal Status 0: USB Bus D- Signal is low 1: USB Bus D- Signal is high

USB Engine Register (USB_ENG)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
USB_ENG	0xFFF0603C	R/W	USB Engine 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				SDO_RD	CV_LDA	CV_STL	CV_DAT

BITS	DESCRIPTIONS	
[31:4]		Reserved
[3]	SDO_RD	Setup or Bulk-Out Data Read Control 0: NO Operation 1: Read Setup or Bulk-Out Data from USB Host **NOTE: this bit will auto clear after 32 HCLK**
[2]	CV_LDA	USB Class and Vendor Command Last Data Packet Control 0: NO Operation 1: Last Data Packet for Data Input of Class and Vendor Command **NOTE: this bit will auto clear after 32 HCLK**

Continued.

BITS	DESCRIPTIONS	
[1]	CV_STL	USB Class and Vendor Command Stall Control 0: NO Operation 1: Return Stall for Class and Vendor Command **NOTE: this bit will auto clear after 32 HCLK**
[0]	CV_DAT	USB Class and Vendor Command return data control 0: NO Operation 1: The Data Packet for Data Input of Class and Vendor Command or Get Descriptor command is ready. **NOTE: this bit will auto clear after 32 HCLK**

USB Control Register (USB_CTLS)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
USB_CTLS	0xFFF06040	R	USB control transfer 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
CONF							
7	6	5	4	3	2	1	0
Reserved			CTLRPS				

ITS	DESCRIPTIONS	
[31:16]		Reserved
[15:8]	CONF	USB configured value
[7:5]		Reserved
[4:0]	CTLRPS	Control transfer received packet size

USB Configured Value Register (USB_CONFD)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
USB_CONFD	0xFFF06044	R/W	USB Configured Value 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
CONFD							

BITS	DESCRIPTIONS	
[31:8]		Reserved
[7:0]	CONFD	Software configured value

USB Endpoint A Information Register (EPA_INFO)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
EPA_INFO	0xFFF06048	R/W	USB endpoint A information register	0x0000_0000

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

BITS	DESCRIPTIONS	
[31]		Reserved
[30:29]	EPA_TYPE	Endpoint A type 00: reserved 01: bulk 10: interrupt 11: isochronous
[28]	EPA_DIR	Endpoint A direction 0: OUT 1: IN
[27:26]		Reserved
[25:16]	EPA_MPS	Endpoint A max. packet size
[15:12]	EPA_ALT	Endpoint A alternative setting (READ ONLY)
[11:8]	EPA_INF	Endpoint A interface
[7:4]	EPA_CFG	Endpoint A configuration
[3:0]	EPA_NUM	Endpoint A number

USB Endpoint A Control Register (EPA_CTL)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
EPA_CTL	0xFFF0604C	R/W	USB endpoint A 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	EPA_ZERO	EPA_STL_CLR	EPA_THRE	EPA_STL	EPA_RDY	EPA_RST	EPA_EN

BITS	DESCRIPTIONS	
[31:6]		Reserved
[6]	EPA_ZERO	Send zero length packet to HOST
[5]	EPA_STL_CLR	CLEAR the Endpoint A stall(WRITE ONLY)
[4]	EPA_THRE	Endpoint A threshold (only for ISO) 1: once available space in FIFO over 16 bytes, DMA accesses memory 0: once available space in FIFO over 32 bytes, DMA accesses memory
[3]	EPA_STL	Set the Endpoint A stall
[2]	EPA_RDY	The memory is ready for Endpoint A to access
[1]	EPA_RST	Endpoint A reset
[0]	EPA_EN	Endpoint A enable

USB Endpoint A interrupt enable Register (EPA_IE)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
EPA_IE	0xFFF06050	R/W	USB endpoint A 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		EPA_CF_IE	EPA_BUS_ERR_IE	EPA_DMA_IE	EPA_ALT_IE	EPA_TK_IE	EPA_STL_IE

BITS	DESCRIPTIONS	
[31:6]		Reserved
[5]	EPA_CF_IE	Endpoint A clear feature interrupt enable
[4]	EPA_BUS_ERR_IE	Endpoint A system bus error interrupt enable
[3]	EPA_DMA_IE	Endpoint A DMA transfer complete interrupt enable
[2]	EPA_ALT_IE	Endpoint A alternate setting interrupt enable
[1]	EPA_TK_IE	Endpoint A token input interrupt enable
[0]	EPA_STL_IE	Endpoint A stall interrupt enable

USB Endpoint A Interrupt Clear Register (EPA_IC)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
EPA_IC	0xFFF06054	W	USB endpoint A interrupt 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		EPA_CF_IC	EPA_BUS_ERR_IC	EPA_DMA_IC	EPA_ALT_IC	EPA_TK_IC	EPA_STL_IC

BITS	DESCRIPTIONS	
[31:6]		Reserved
[5]	EPA_CF_INT_IC	Endpoint A clear feature interrupt clear
[4]	EPA_BUS_ERR_IC	Endpoint A system bus error interrupt clear
[3]	EPA_DMA_IC	Endpoint A DMA transfer complete interrupt clear

Continued.

BITS	DESCRIPTIONS	
[2]	EPA_ALT_IC	Endpoint A alternate setting interrupt clear
[1]	EPA_TK_IC	Endpoint A token input interrupt clear
[0]	EPA_STL_IC	Endpoint A stall interrupt clear

USB Endpoint A Interrupt Status Register (EPA_IS)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
EPA_IS	0xFFF06058	R	USB endpoint A 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		EPA_CF_IS	EPA_BUS_ERR_IS	EPA_DMA_IS	EPA_ALT_IS	EPA_TK_IS	EPA_STL_IS

BITS	DESCRIPTIONS	
[31:6]		Reserved
[5]	EPA_CF_IS	Endpoint A clear feature interrupt status
[4]	EPA_BUS_ERR_IS	Endpoint A system bus error interrupt status
[3]	EPA_DMA_IS	Endpoint A DMA transfer complete interrupt status
[2]	EPA_ALT_IS	Endpoint A alternative setting interrupt status
[1]	EPA_TK_IS	Endpoint A token interrupt status
[0]	EPA_STL_IS	Endpoint A stall interrupt status

USB Endpoint A Address Register (EPA_ADDR)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
EPA_ADDR	0xFFF0605C	R/W	USB endpoint A address register	0x0000_0000

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

BITS	DESCRIPTIONS	
[31:0]	EPA_ADDR	Endpoint A transfer address

USB Endpoint A transfer length Register (EPA_LENTH)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
EPA_LENTH	0xFFF06060	R/W	USB endpoint A transfer length register	0x0000_0000

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

BITS	DESCRIPTIONS	
[31:20]		Reserved
[19:0]	EPA_LENTH	Endpoint A transfer length

USB Endpoint B Information Register (EPB_INFO)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
EPB_INFO	0xFFF06064	R/W	USB endpoint B information register	0x0000_0000

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

BITS	DESCRIPTIONS	
[31]		Reserved
[30:29]	EPB_TYPE	Endpoint B type 00: reserved 01: bulk 10: interrupt 11: isochronous
[28]	EPB_DIR	Endpoint B direction 0: OUT 1: IN
[27:26]		Reserved
[25:16]	EPB_MPS	Endpoint B max. packet size

Continued.

BITS	DESCRIPTIONS	
[15:12]	EPB_ALT	Endpoint B alternative setting (READ ONLY)
[11:8]	EPB_INF	Endpoint B interface
[7:4]	EPB_CFG	Endpoint B configuration
[3:0]	EPB_NUM	Endpoint B number

USB Endpoint B Control Register (EPB_CTL)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
EPB_CTL	0xFFF06068	R/W	USB endpoint B 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	EPB_ZERO	EPB_STL_CLR	EPB_THRE	EPB_STL	EPB_RDY	EPB_RST	EPB_EN

BITS	DESCRIPTIONS	
[31:7]		Reserved
[6]	EPB_ZERO	Send zero length packet back to HOST
[5]	EPB_STL_CLR	Clear the Endpoint B stall(WRITE ONLY)
[4]	EPB_THRE	Endpoint B threshold (only for ISO) 1: once available space in FIFO over 16 bytes, DMA accesses memory 0: once available space in FIFO over 32 bytes, DMA accesses memory
[3]	EPB_STL	Set the Endpoint B stall

Continued.

BITS	DESCRIPTIONS	
[2]	EPB_RDY	The memory is ready for Endpoint B to access
[1]	EPB_RST	Endpoint B reset
[0]	EPB_EN	Endpoint B enable

USB Endpoint B interrupt enable Register (EPB_IE)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
EPB_IE	0xFFF0606C	R/W	USB endpoint B 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		EPB_CF_IE	EPB_BUS_ERR_IE	EPB_DMA_IE	EPB_ALT_IE	EPB_TK_IE	EPB_STL_IE

BITS	DESCRIPTIONS	
[31:6]		Reserved
[5]	EPB_CF_IE	Endpoint B clear feature interrupt enable
[4]	EPB_BUS_ERR_IE	Endpoint B system bus error interrupt enable
[3]	EPB_DMA_IE	Endpoint B DMA transfer complete interrupt enable
[2]	EPB_ALT_IE	Endpoint B alternate setting interrupt enable
[1]	EPB_TK_IE	Endpoint B token input interrupt enable
[0]	EPB_STL_IE	Endpoint B stall interrupt enable

USB Endpoint B Interrupt Clear Register (EPB_IC)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
EPB_IC	0xFFF06070	W	USB endpoint B interrupt 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		EPB_CF_IC	EPB_BUS_ERR_IC	EPB_DMA_IC	EPB_ALT_IC	EPB_TK_IC	EPB_STL_IC

BITS	DESCRIPTIONS	
[31:6]		Reserved
[5]	EPB_CF_IC	Endpoint B clear feature interrupt clear
[4]	EPB_BUS_ERR_IC	Endpoint B system bus error interrupt clear
[3]	EPB_DMA_IC	Endpoint B DMA transfer complete interrupt clear
[2]	EPB_ALT_IC	Endpoint B alternate setting interrupt clear
[1]	EPB_TK_IC	Endpoint B token input interrupt clear
[0]	EPB_STL_IC	Endpoint B stall interrupt clear

USB Endpoint B Interrupt Status Register (EPB_IS)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
EPB_IS	0xFFF06074	R	USB endpoint B 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		EPB_CF_IS	EPB_BUS_ERR_IS	EPB_DMA_IS	EPB_ALT_IS	EPB_TK_IS	EPB_STL_IS

BITS	DESCRIPTIONS	
[31:6]		Reserved
[5]	EPB_CF_IS	Endpoint B clear feature interrupt status
[4]	EPB_DMA_IS	Endpoint B system bus error interrupt status
[3]	EPB_DMA_IS	Endpoint B DMA transfer complete interrupt status
[2]	EPB_ALT_IS	Endpoint B alternative setting interrupt status
[1]	EPB_TK_IS	Endpoint B token interrupt status
[0]	EPB_STL_IS	Endpoint B stall interrupt status

USB Endpoint B Address Register (EPB_ADDR)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
EPB_ADDR	0xFFF06078	R/W	USB endpoint B address register	0x0000_0000

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

BITS	DESCRIPTIONS	
[31:0]	EPB_ADDR	Endpoint B transfer address

USB Endpoint B transfer length Register (EPB_LENTH)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
EPB_LENTH	0xFFF0607C	R/W	USB endpoint B transfer length register	0x0000_0000

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

BITS	DESCRIPTIONS	
[31:20]		Reserved
[19:0]	EPB_LENTH	Endpoint B transfer length

USB Endpoint C Information Register (EPC_INFO)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
EPC_INFO	0xFFF06080	R/W	USB endpoint C information register	0x0000_0000

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

BITS	DESCRIPTIONS	
[31]		Reserved
[30:29]	EPC_TYPE	Endpoint C type 00: reserved 01: bulk 10: interrupt 11: isochronous
[28]	EPC_DIR	Endpoint C direction 0: OUT 1: IN
[27:26]		Reserved
[25:16]	EPC_MPS	Endpoint C max. packet size
[15:12]	EPC_ALT	Endpoint C alternative setting (READ ONLY)
[11:8]	EPC_INF	Endpoint C interface
[7:4]	EPC_CFG	Endpoint C configuration
[3:0]	EPC_NUM	Endpoint C number

USB Endpoint C Control Register (EPC_CTL)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
EPC_CTL	0xFFF06084	R/W	USB endpoint C 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	EPC_ZERO	EPC_STL_CLR	EPC_THRE	EPC_STL	EPC_RDY	EPC_RST	EPC_EN

BITS	DESCRIPTIONS	
[31:7]		Reserved
[6]	EPC_ZERO	Send zero length packet back to HOST
[5]	EPC_STL_CLR	Clear the Endpoint C stall(WRITE ONLY)
[4]	EPC_THRE	Endpoint C threshold (only for ISO) 1: once available space in FIFO over 16 bytes, DMA accesses memory 0: once available space in FIFO over 32 bytes, DMA accesses memory
[3]	EPC_STL	Set the Endpoint C stall
[2]	EPC_RDY	The memory is ready for Endpoint C to access
[1]	EPC_RST	Endpoint C reset
[0]	EPC_EN	Endpoint C enable

USB Endpoint C interrupt enable Register (EPC_IE)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
EPC_IE	0xFFF06088	R/W	USB endpoint C 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		EPC_CF_IE	EPC_BUS_ERR_IE	EPC_DMA_IE	EPC_ALT_IE	EPC_TK_IE	EPC_STL_IE

BITS	DESCRIPTIONS	
[31:6]		Reserved
[5]	EPC_CF_IE	Endpoint C clear feature interrupt enable
[4]	EPC_DMA_IE	Endpoint C system bus error interrupt enable
[3]	EPC_DMA_IE	Endpoint C DMA transfer complete interrupt enable
[2]	EPC_ALT_IE	Endpoint C alternate setting interrupt enable
[1]	EPC_TK_IE	Endpoint C token input interrupt enable
[0]	EPC_STL_IE	Endpoint C stall interrupt enable

USB Endpoint C Interrupt Clear Register (EPC_IC)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
EPC_IC	0xFFF0608C	W	USB endpoint C interrupt 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		EPC_CF_IC	EPC_BUS_ERR_IC	EPC_DMA_IC	EPC_ALT_IC	EPC_TK_IC	EPC_STL_IC

BITS	DESCRIPTIONS	
[31:6]		Reserved
[5]	EPC_CF_IC	Endpoint C clear feature interrupt clear
[4]	EPC_DMA_IC	Endpoint C system bus error interrupt clear
[3]	EPC_DMA_IC	Endpoint C DMA transfer complete interrupt clear
[2]	EPC_ALT_IC	Endpoint C alternate setting interrupt clear
[1]	EPC_TK_IC	Endpoint C token input interrupt clear
[0]	EPC_STL_IC	Endpoint C stall interrupt clear

USB Endpoint C Interrupt Status Register (EPC_IS)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
EPC_IS	0xFFF06090	R	USB endpoint C 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		EPC_CF_IS	EPC_BUS_ERR_IS	EPC_DMA_IS	EPC_ALT_IS	EPC_TK_IS	EPC_STL_IS

BITS	DESCRIPTIONS	
[31:6]		Reserved
[5]	EPC_CF_IS	Endpoint C clear feature interrupt status
[4]	EPC_BUS_ERR_IS	Endpoint A system bus error interrupt status
[3]	EPC_DMA_IS	Endpoint A DMA transfer complete interrupt status
[2]	EPC_ALT_IS	Endpoint A alternative setting interrupt status
[1]	EPC_TK_IS	Endpoint A token interrupt status
[0]	EPC_STL_IS	Endpoint A stall status

USB Endpoint C Address Register (EPC_ADDR)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
EPC_ADDR	0xFFF0_6094	R/W	USB endpoint C address register	0x0000_0000

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

BITS	DESCRIPTIONS	
[31:0]	EPC_ADDR	Endpoint C transfer address

USB Endpoint C transfer length Register (EPC_LENTH)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
EPC_LENTH	0xFFF0_6098	R/W	USB endpoint C transfer length register	0x0000_0000

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

BITS	DESCRIPTIONS	
[31:20]		Reserved
[19:0]	EPC_LENTH	Endpoint C transfer length

USB Endpoint A Remain transfer length Register (EPA_XFER)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
EPA_XFER	0xFFF0_609C	R/W	USB endpoint A remain transfer length register	0x0000_0000

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

BITS	DESCRIPTIONS	
[31:20]		Reserved
[19:0]	EPA_XFER	Endpoint A remain transfer length

USB Endpoint A Remain packet length Register (EPA_PKT)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
EPA_PKT	0xFFF0_60A0	R/W	USB endpoint A remain packet length 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						EPA_PKT	
7	6	5	4	3	2	1	0
EPA_PKT							

BITS	Descriptions	
[31:10]		Reserved
[9:0]	EPA_PKT	Endpoint A remain packet length

USB Endpoint B Remain transfer length Register (EPB_XFER)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
EPB_XFER	0xFFF0_60A4	R/W	USB endpoint B remain transfer length register	0x0000_0000

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

BITS	DESCRIPTIONS	
[31:20]		Reserved
[19:0]	EPB_XFER	Endpoint B remain transfer length

USB Endpoint B Remain packet length Register (EPB_PKT)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
EPB_PKT	0xFFF0_60A8	R/W	USB endpoint B remain packet length 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						EPB_PKT	
7	6	5	4	3	2	1	0
EPB_PKT							

BITS	DESCRIPTIONS	
[31:10]		Reserved
[9:0]	EPB_PKT	Endpoint B remain packet length

USB Endpoint C Remain transfer length Register (EPC_XFER)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
EPC_XFER	0xFFF0_60AC	R/W	USB endpoint C remain transfer length register	0x0000_0000

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

BITS	DESCRIPTIONS	
[31:20]		Reserved
[19:0]	EPC_XFER	Endpoint C remain transfer length

USB Endpoint C Remain packet length Register (EPC_PKT)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
EPC_PKT	0xFFF0_60B0	R/W	USB endpoint C remain packet length 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						EPC_PKT	
7	6	5	4	3	2	1	0
EPC_PKT							

BITS	DESCRIPTIONS	
[31:10]		Reserved
[9:0]	EPC_PKT	Endpoint C remain packet length

6.9 Audio Controller

The audio controller consists of I²S/AC-link protocol to interface with external audio CODEC.

One 8-level deep FIFO for read path and write path and each level has 32-bit width (16 bits for right channel and 16 bits for left channel). One DMA controller handles the data movement between FIFO and memory.

The following are the property of the DMA.

- Always 8-beat incrementing burst
- Always bus lock when 8-beat incrementing burst
- When reach middle and end address of destination address, a DMA_IRQ is requested to CPU automatically

An AHB master port and an AHB slave port are offered in audio controller.

6.9.1 I²S Interface

The I²S interface signals are shown as Figure 6.9.1

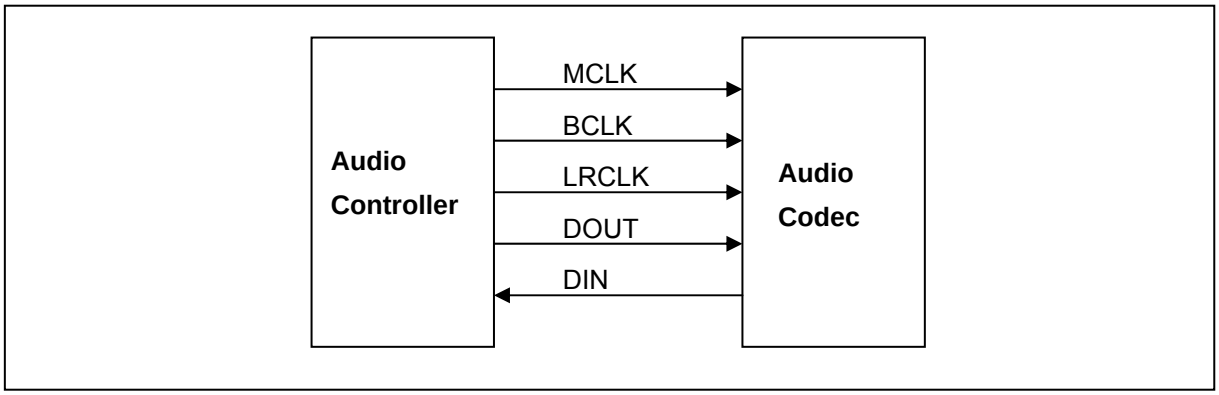


Figure 6.9.1 The interface signal of I²S

The 16 bits I²S and MSB-justified format are support, the timing diagram is shown as Figure 6.9.2

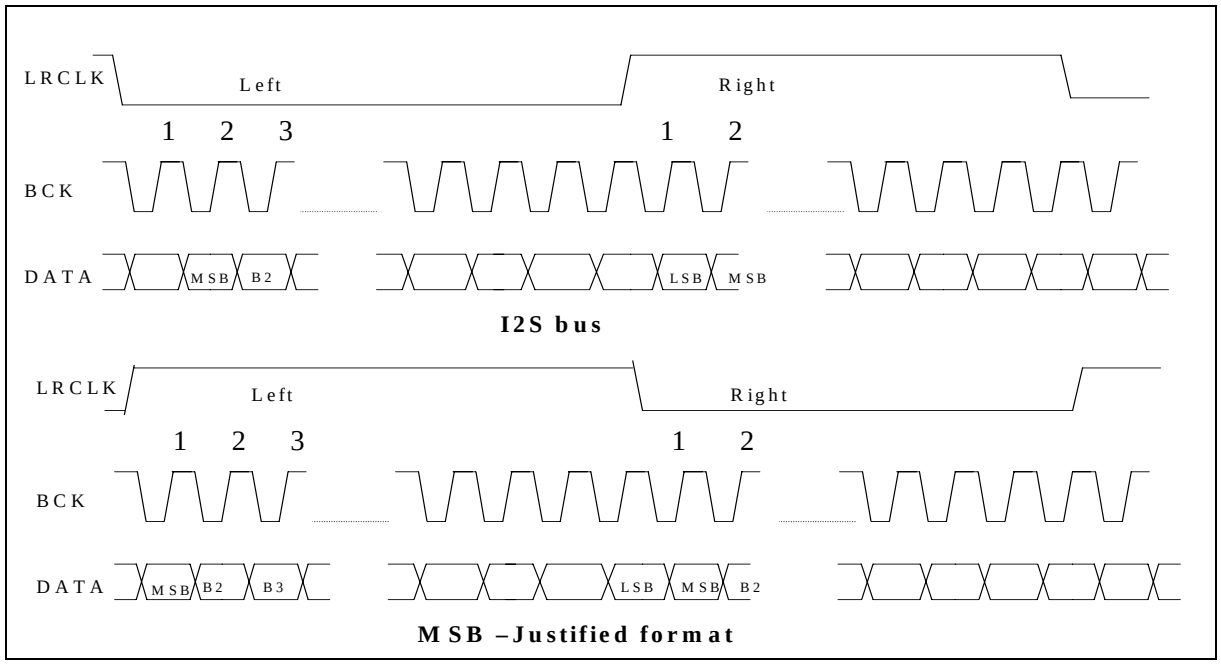


Figure 6.9.2 The format of I²S

The sampling rate, bit shift clock frequency could be set by the control register ACTL_I²SCON.

6.9.2 AC97 Interface

The AC97 interface, called AC-link is supported. For input and output direction, each frame contains a Tag slot and 12 data slots. However, in the 12 data slots, **only 4 slots are used in W90N745**, other 8 slots are not supported, and the control data and audio data are transferred in the 4 valid slots. Each slot contains 20 bits data.

The interface signals are shown as Figure 6.9.3

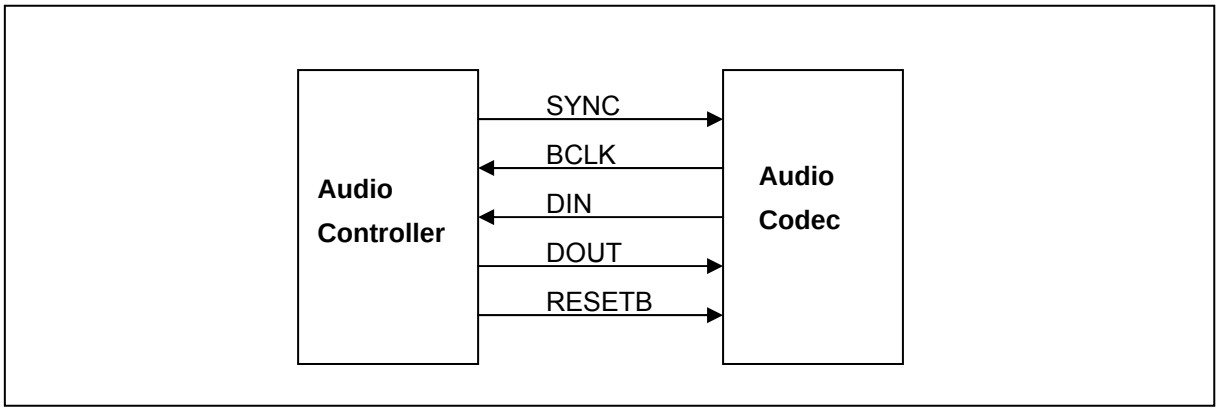


Figure 6.9.3 The interface signal of AC-link

The signal format is shown as Figure 6.9.4

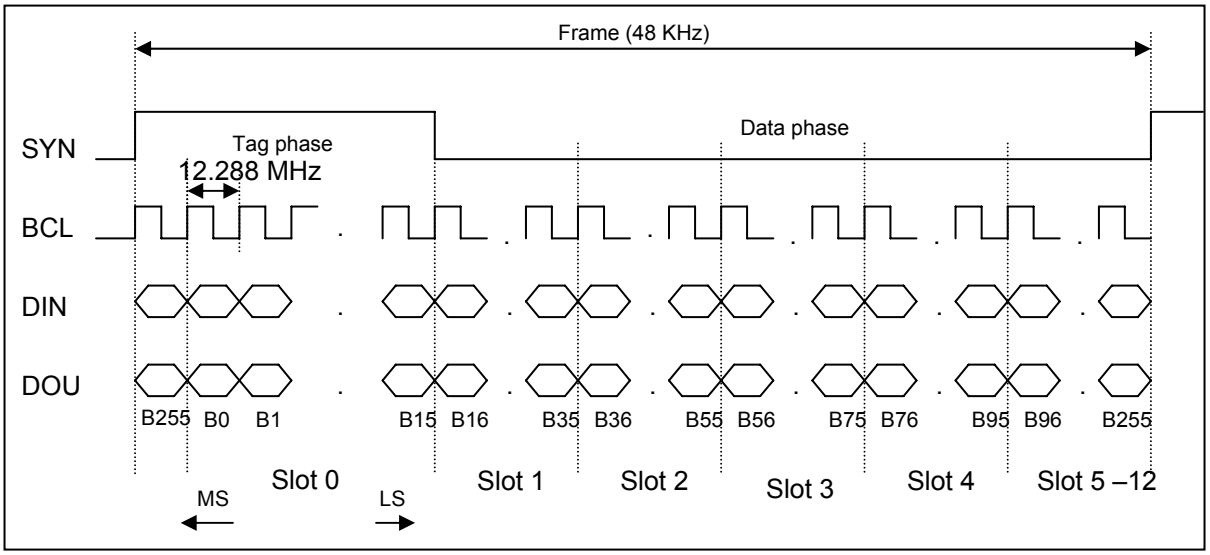


Figure 6.9.4 The signal format of AC-link

The structure of **output frame** is shown as below:

SLOT #	0	1	2	3	4	5	6	7	8	9	10	11	12
CONTENT	Tag	CMD ADDR	CMD DATA	PCM LEFT	PCM RIGHT	**Unused**							
BITS	15-0	19-0	19-0	19-0	19-0	159 - 0							
PHASE	Tag phase	Data phase											

The output frame data format is shown as following:

SLOT #	BIT	DESCRIPTION
Tag (slot 0)	15	Frame validity bit, 1 is valid, 0 is invalid.
	14 - 3	Slot validity, but in W90N745, only bits 6-3 are used, bits 14-7 are unused. Bit 3 is corresponding to slot 1, bit 4 is corresponding to slot 2, etc.. 1 is valid, 0 is invalid. The unused bits 14-7 should be cleared to 0.
	2 - 0	This field should be cleared to 0.
CMD ADDR (slot 1)	19	Read/write control, 1 for read and 0 for write
	18-12	Control register address
	11 - 0	This field should be cleared to 0

Continued.

SLOT #	BIT	DESCRIPTION
CMD DATA (slot 2)	19 - 4	Control register write data. It should be cleared to 0 if current operation is read.
	3 - 0	This field should be cleared to 0
PCM LEFT (slot 3)	19 - 4	PCM playback data for left channel
	3 - 0	This field should be cleared to 0
PCM RIGHT (slot 4)	19 - 4	PCM playback data for right channel
	3 - 0	This field should be cleared to 0

The structure of **input frame** is shown as below:

Slot #	0	1	2	3	4	5	6	7	8	9	10	11	12
Content	Tag	status ADDR	status DATA	PCM LEFT	PCM RIGHT	**Unused**							
Bits	0-15	19-0	19-0	19-0	19-0	159 - 0							

The input frame data format is shown as following:

SLOT #	BIT	DESCRIPTION
Tag (slot 0)	15	Frame validity bit, 1 is valid, 0 is invalid.
	14 - 3	Slot validity, but in W90N745, only bits 6-3 are used, bits 14-7 are unused. Bit 3 is corresponding to slot 1, bit 4 is corresponding to slot 2, etc.. 1 is valid, 0 is invalid. The unused bits 14-7 should be cleared to 0.
	2 - 0	This field should be cleared to 0.
Status ADDR (slot 1)	19	This bit should be cleared to 0
	18-12	Control register address echo which previous frame requested
	11	PCM data for left channel request, it should be always 0 when VRA=0 (VRA: Variable Rate Audio mode).
	10	PCM data for right channel request (Same as Bit 11).
	9 - 0	This field should be cleared to 0
Status DATA (slot 2)	19 - 4	Control register read data which previous frame requested. It should be cleared to 0 if this slot is invalid.
	3 - 0	This field should be cleared to 0
PCM LEFT (slot 3)	19 - 4	PCM record data for left channel
	3 - 0	This field should be cleared to 0
PCM RIGHT (slot 4)	19 - 4	PCM record data for right channel
	3 -0	This field should be cleared to 0

6.9.3 Audio Controller 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
ACTL_CON	0xFFF0_9000	R/W	Audio controller control register	0x0000_0000
ACTL_RESET	0xFFF0_9004	R/W	Sub block reset control register	0x0000_0000
ACTL_RDSTB	0xFFF0_9008	R/W	DMA destination base address register for record	0x0000_0000
ACTL_RDST_LENGTH	0xFFF0_900C	R/W	DMA destination length register for record	0x0000_0000
ACTL_RDSTC	0xFFF0_9010	R	DMA destination current address register for record	0x0000_0000
ACTL_RSR	0xFFF0_9014	R/W	Record status register	0x0000_0000
ACTL_PDSTB	0xFFF0_9018	R/W	DMA destination base address register for play	0x0000_0000
ACTL_PDST_LENGTH	0xFFF0_901C	R/W	DMA destination length register for play	0x0000_0000
ACTL_PDSTC	0xFFF0_9020	R	DMA destination current address register for play	0x0000_0000
ACTL_PSR	0xFFF0_9024	R/W	Play status register	0x0000_0004
ACTL_I²SCON	0xFFF0_9028	R/W	I²S control register	0x0000_0000
ACTL_ACCON	0xFFF0_902C	R/W	AC-link control register	0x0000_0000
ACTL_ACOS0	0xFFF0_9030	R/W	AC-link out slot 0	0x0000_0000
ACTL_ACOS1	0xFFF0_9034	R/W	AC-link out slot 1	0x0000_0080
ACTL_ACOS2	0xFFF0_9038	R/W	AC-link out slot 2	0x0000_0000
ACTL_ACIS0	0xFFF0_903C	R	AC-link in slot 0	0x0000_0000
ACTL_ACIS1	0xFFF0_9040	R	AC-link in slot 1	0x0000_0000
ACTL_ACIS2	0xFFF0_9044	R	AC-link in slot 2	0x0000_0000

Audio controller control registers (ACTL_CON)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
ACTL_CON	0xFFF0_9000	R/W	Audio controller control register	0x0000_0000

The ACTL_CON register control the basic operation of audio controller.

31	30	29	28	27	26	25	24
23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8
Reserved	Reserved	Reserved	R_DMA_IRQ	T_DMA_IRQ	Reserved		I²S_AC_PIN_SEL
7	6	5	4	3	2	1	0
FIFO_TH	Reserved	Reserved			BLOCK_EN[1:0]		Reserved

BITS	DESCRIPTIONS	
[15]	**Reserved**	-
[14]	**Reserved**	-
[13]	**Reserved**	-
[12]	R_DMA_IRQ	When recording, when the DMA destination current address reach the DMA destination end address or middle address, the R_DMA_IRQ bit will be set to 1 automatically, and this bit could be cleared to 0 by CPU. The bit is hardwired to ARM as interrupt request signal with an inverter. The R_DMA_IRQ bit is read/write (**write 1 to clear**)
[11]	T_DMA_IRQ	Transmit DMA interrupt request bit. When DMA current address reach the middle address (((ACTL_DESE – ACTL_DESB)-1)/2 + ACTL_DESB) or reach the end address ACTL_DESB, the bit T_DMA_IRQ will be set to 1, and this bit could be clear to 0 by write “1” by CPU. And the bit is hardwired to ARM as interrupt request signal with an inverter. The T_DMA_IRQ bit is read/write (**write 1 to clear**).
[8]	I²S_AC_PIN_SEL	I²S or AC-link pin selection • If I²S_AC_PIN_SEL = 0, the pins select I²S • If I²S_AC_PIN_SEL = 1, the pins select AC-link The I²S_AC_PIN_SEL bis is read/write
[7]	FIFO_TH	FIFO threshold control bit • If FIFO_TH=0, the FIFO threshold is 8 level • If FIFO_TH=1, the FIFO threshold is 4 level The FIFO_TH bit is read/write
[6]	**Reserved**	
[2:1]	BLOCK_EN[1:0]	Audio interface type selection • If BLOCK_EN[0]=0/1, I²S interface is disable/enable • If BLOCK_EN[1]=0/1, AC-link interface is disable/enable The BLOCK_EN[1:0] bits are read/write
[0]	**Reserved**	

Sub-block reset control register (ACTL_RESET)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
ACTL_RESET	0xFFF0_9004	R/W	Sub block reset control	0x0000_0000

The value in ACTL_RESET register control the reset operation in each sub block.

31	30	29	28	27	26	25	24
23	22	21	20	19	18	17	16
							ACTL_RESET
15	14	13	12	11	10	9	8
RECORD_SINGLE[1:0]		PLAY_SINGLE[1:0]		Reserved			AC_RECORD
7	6	5	4	3	2	1	0
AC_PLAY	I^2S_RECORD	I^2S_PLAY	Reserved			AC_RESET	I^2S_RESET

BITS	DESCRIPTIONS	
[31:17]	**Reserved**	-
[16]	**ACTL_RESET**	**Audio controller reset control bit** 1 = the whole audio controller is reset 0 = the audio controller is normal operation The ACTL_RESET bit is read/write
[15:14]	**RECORD_SINGLE [1:0]**	**record single/dual channel select bits** 2'b11= the record is dual channel 2'b01= the record only select left channel 2'b10= the record only select right channel 2'b00 is reserved Note that, when ADC is selected as record path, it only support left channel record. The PLAY_SINGLE[1:0] bits are read/write

Continued.

BITS	DESCRIPTIONS	
[13:12]	**PLAY_SINGLE [1:0]**	**Playback single/dual channel select bits** PLAY_SINGLE[1:0]=11, the playback is in **stereo** mode PLAY_SINGLE[1:0]=10, the playback is in **mono** mode PLAY_SINGLE[1:0]= 00 & 01 is reserved The PLAY_SINGLE[1:0] bits are read/write
[8]	**AC_RECORD**	**AC link record control bit** AC_RECORD=0, the record path of AC link is disable AC_RECORD=1, the record path of AC link is enable The AC_RECORD bit is read/write
[7]	**AC_PLAY**	**AC link playback control bit** AC_PLAY=0, the playback path of AC link is disable AC_PLAY=1, the playback path of AC link is enable The AC_PLAY bit is read/write
[6]	**I²S_RECORD**	**I²S record control bit** I²S_RECORD=0, the record path of I²S is disable I²S_RECORD=1, the record path of I²S is enable The I²S_RECORD bit is read/write
[5]	**I²S_PLAY**	**I²S playback control bit** I²S_PLAY=0, the playback path of I²S is disable I²S_PLAY=1, the playback path of I²S is enable The I²S_PLAY bit is read/write
[1]	**AC_RESET**	**AC link sub block RESET control bit** AC_RESET=0, release the AC link function block from reset mode AC_RESET=1, force the AC link function block to reset mode The AC_RESET bit is read/write
[0]	**I²S_RESET**	**I²S sub block RESET control bit** I²S_RESET=0, release the I²S function block from reset mode I²S_RESET=1, force the I²S function block to reset mode The I²S_RESET bit is read/write

DMA record destination base address (ACTL_RDSTB)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
ACTL_RDSTB	0xFFF0_9008	R/W	DMA record destination base address	0x0000_0000

The value in ACTL_RDSTB register is the record destination base address of DMA, and only could be changed by CPU.

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

BITS	DESCRIPTIONS	
[31:0]	**AUDIO_RDSTB[31:0]**	32-bit record destination base address The AUDIO_RDSTB[31:0] bits is read/write.

DMA destination end address (ACTL_RDST_LENGTH)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
ACTL_RDST_LENGTH	0xFFF0_900C	R/W	DMA record destination address length	0x0000_0000

The value in ACTL_RDST_LENGTH register is the record destination address length of DMA, and the register could only be changed by CPU.

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

BITS	DESCRIPTIONS	
[31:0]	**AUDIO_RDST_L[31:0]**	32-bit record destination address length The AUDIO_RDST_L[31:0] bits is read/write.

DMA destination current address (ACTL_RDSTC)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
ACTL_RDSTC	0xFFF0_9010	RO	DMA record destination current address	0x0000_0000

The value in ACTL_RDSTC is the DMA record destination current address, this register could only be read by CPU.

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

BITS	DESCRIPTIONS	
[31:0]	**AUDIO_RDSTC[31:0]**	32-bit record destination current address The AUDIO_RDSTC[31:0] bits is read only.

Audio controller record status register (ACTL_RSR)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
ACTL_RSR	0xFFF0_9014	R/W	Audio controller FIFO and DMA status register for record	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					R_FIFO_FULL	R_DMA_END_IRQ	R_DMA_MIDDLE_IRQ

BITS	DESCRIPTIONS	
[31:3]	Reserved	-
[2]	R_FIFO_FULL	Record FIFO full indicator bit R_FIFO_FULL=0, the record FIFO not full R_FIFO_FULL=1, the record FIFO is full The R_FIFO_READY bit is read only
[1]	R_DMA_END_IRQ	DMA end address interrupt request bit for record R_DMA_END_IRQ=0, means record DMA address does not reach the end address R_DMA_END_IRQ=1, means record DMA address reach the end address The R_DMA_END_IRQ bit is readable, and only can be clear by write “1” to this bit
[0]	R_DMA_MIDDLE_IRQ	DMA address interrupt request bit for record R_DMA_MIDDLE_IRQ=0, means record DMA address does not reach the middle address R_DMA_MIDDLE_IRQ=1, means record DMA address reach the middle address The R_DMA_MIDDLE_IRQ bit is readable, and only can be clear by write “1” to this bit

DMA play destination base address (ACTL_PDSTB)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
ACTL_PDSTB	0xFFF0_9018	R/W	DMA play destination base address	0x0000_0000

The value in ACTL_PDSTB register is the play destination base address of DMA, and only could be changed by CPU.

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

BITS	DESCRIPTIONS	
[31:0]	**AUDIO_PDSTB[31:0]**	32-bit play destination base address The AUDIO_PDSTB[31:0] bits is read/write.

DMA destination end address (ACTL_PDST_LENGTH)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
ACTL_PDST_LENGTH	0xFFF0_901C	R/W	DMA play destination address length	0x0000_0000

The value in ACTL_PDST_LENGTH register is the play destination address length of DMA, and the register could only be changed by CPU.

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

BITS	DESCRIPTIONS	
[31:0]	**AUDIO_PDST_L[31:0]**	32-bit play destination address length The AUDIO_PDST_L[31:0] bits is read/write.

DMA destination current address (ACTL_PDSTC)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
ACTL_PDSTC	0xFFF0_9020	RO	DMA play destination current address	0x0000_0000

The value in ACTL_PDSTC is the DMA play destination current address, this register could only be read by CPU.

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

BITS	DESCRIPTIONS	
[31:0]	**AUDIO_PDSTC[31:0]**	32-bit play destination current address The AUDIO_PDSTC[31:0] bits is read/write.

Audio controller playback status register (ACTL_PSR)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
ACTL_PSR	0xFFF0_9024	R/W	Audio controller FIFO and DMA status register for playback	0x0000_0004

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					P_FIFO_EMPTY	P_DMA_END_IRQ	P_DMA_MIDDLE_IRQ

BITS	DESCRIPTIONS	
[31:3]	Reserved	-
[2]	P_FIFO_EMPTY	Playback FIFO empty indicator bit P_FIFO_EMPTY=0, the playback FIFO is not empty P_FIFO_EMPTY=1, the playback FIFO is empty The P_FIFO_EMPTY bit is read only
[1]	P_DMA_END_IRQ	DMA end address interrupt request bit for playback P_DMA_END_IRQ=0, means playback DMA address does not reach the end address P_DMA_END_IRQ=1, means playback DMA address reach the end address The P_DMA_END_IRQ bit is readable, and only can be clear by write “1” to this bit
[0]	P_DMA_MIDDLE_IRQ	DMA address interrupt request bit for playback P_DMA_MIDDLE_IRQ=0, means playback DMA address does not reach the middle address P_DMA_MIDDLE_IRQ=1, means playback DMA address reach the middle address The P_DMA_MIDDLE_IRQ bit is readable, and only can be clear by write “1” to this bit

I²S control register (ACTL_I²SCON)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
ACTL_I²SCON	0xFFF0_9028	R/W	I²S control register	0x0000_0000

The ACTL_I²SCON is the I²S basic operation control register.

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

BITS	DESCRIPTIONS	
[31:20]	Reserved	-
[19:16]	PRS[3:0]	I²S frequency pre-scaler selection bits. (FPLL is the input PLL frequency, MCLK is the output main clock) PSR[3:0]=0000, MCLK=FPLL/1 PSR[3:0]=0001, MCLK=FPLL/2 PSR[3:0]=0010, MCLK=FPLL/3 PSR[3:0]=0011, MCLK=FPLL/4 PSR[3:0]=0100, MCLK=FPLL/5 PSR[3:0]=0101, MCLK=FPLL/6 PSR[3:0]=0110, MCLK=FPLL/7 PSR[3:0]=0111, MCLK=FPLL/8 PSR[3:0]=1000, reserved PSR[3:0]=1001, MCLK=FPLL/10 PSR[3:0]=1010, reserved PSR[3:0]=1011, MCLK=FPLL/12 PSR[3:0]=1100, reserved PSR[3:0]=1101, MCLK=FPLL/14 PSR[3:0]=1110, reserved PSR[3:0]=1111, MCLK=FPLL/16 (when the division factor is 3/5/7, the duty cycle of MCLK is not 50%, the high duration is 0.5*FPLL) The PSR[3:0] bits are read/write

Continued

BITS	DESCRIPTIONS	
[7:6]	BCLK_SEL [1:0]	I²S serial data clock frequency selection bit BCLK_SEL[1:0]=00, 32fs is selected (fs is sampling rate), when FS_SEL=0, the frequency of bit clock is MCLK/8, when FS_SEL=1, the frequency of bit clock is MCLK/12. BCLK_SEL[1:0]=01, 48fs is selected (only when FS_SEL=1, this term could be selection), when FS_SEL=1, the frequency of bit clock is MCLK/8. The BCLK_SEL[1:0] bits are read/write
[5]	FS_SEL	I²S sampling frequency selection bit FS_SEL=0, FMCLK/256 is selected (FMCLK is the frequency of signal MCLK) FS_SEL=1, FMCLK/384 is selected The FS_SEL bit is read/write
[4]	MCLK_SEL	I²S MCLK output selection bit MCLK_SEL=0, I²S MCLK output will follow the PRS[3:0] setting. MCLK_SEL=1, I²S MCLK output will be the same with FPLL. The MCLK_SEL bit is read/write
[3]	FORMAT	I²S format selection bits FORMAT=0, I²S compatible format is selected FORMAT=1, MSB-justified format is selected The FORMAT bit is read/write
[2:0]	Reserved	-

AC-link Control Register (ACTL_ACCON)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
ACTL_ACCON	0xFFF0_902C	R/W	AC-link control register	0x0000_0000

The ACTL_ACCON register is the AC-link basic operation control register.

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		AC_BCLK_PU_EN	AC_R_FINISH	AC_W_FINISH	AC_W_RES	AC_C_RES	Reserved

BITS	DESCRIPTIONS	
[6]	Reserved	-
[5]	AC_BCLK_PU_EN	This bit controls the AC_BCLK pin pull-high resister. AC_BCLK_PU_EN=0, the AC_BCLK pin pull-high resister will be disabled AC_BCLK_PU_EN=1, the AC_BCLK pin pull-high resister will be enabled The AC_BCLK_PU_EN bit is read/write.
[4]	AC_R_FINISH	AC-link read data ready bit. When read data indexed by previous frame is shifted into ACTL_ACIS2, the AC_R_FINISH bit will be set to 1 automatically. After CPU read out the read data, AC_R_FINISH bit will be cleared to 0. AC_R_FINISH=0, read data buffer has been read by CPU AC_R_FINISH=1, read data buffer is ready for CPU read The AC_R_FINISH bit is read only
[3]	AC_W_FINISH	AC-link write frame finish bit. When writing data to register ACTL_ACOS0, the AC_W_FINISH bit will be set to 1 automatically. After AC-link interface shift out the register ACTL_ACOS0, the AC_W_FINISH bit will be cleared to 0. AC_W_FINISH=0, AC-link control data out buffer has been shifted out to codec by CPU and data out buffer is empty. AC_W_FINISH=1, AC-link control data out buffer is ready to be shifted out(After users have wrote data into register ACTL_ACOS0) The AC_W_FINISH bit is read only

Continued.

BITS	DESCRIPTIONS	
[2]	AC_W_RES	AC-link warm reset control bit, when this bit is set to 1, (AC-link begin warn reset procedure, after warn reset procedure finished, this bit will be cleared automatically) the interface signal AC_SYNC is high, when this bit is set to 0, the interface signal AC_SYNC is controlled by AC_BCLK input when this bit is set to 1. Note the AC-link spec. shows it need at least 10 us high duration of AC_SYNC to warn reset AC97. AC_W_RES=0, AC_SYNC pin is controlled by AC_BCLK input pin AC_W_RES=1, AC_SYNC pin is forced to high The AC_W_RES bit is read/write
[1]	AC_C_RES	AC-link cold reset control bit, when this bit is set to 1, the interface signal AC_RESETB is low, when this bit is set to 0, the signal AC_RESETB is high. Note the AC-link spec. shows it need at least 10 us low duration of AC_RESETB to cold reset AC97. AC_C_RES=0, AC_RESETB pin is set to 1 AC_C_RES=1, AC_RESETB pin is set to 0 The AC_C_RES bit is read/write
[0]	Reserved	-

AC-link output slot 0 (ACTL_ACOS0)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
ACTL_ACOS0	0xFFF0_9030	R/W	AC-link out slot 0	0x0000_0000

The ACTL_ACOS0 register store the slot 0 value to be shift out by AC-link. Note that write data to ACTL_ACOS0 register when AC_W_FINISH bit (ACTL_ACCON[3]) is set is invalid. Therefore, **check AC_W_FINISH bit status before write data into ACTL_ACOS0 register.**

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			**VALID_ FRAME**	**SLOT_VALID[3:0]**			

BITS	DESCRIPTIONS	
[31:5]	Reserved	-
[4]	VALID_FRAME	Frame valid indicated bits VALID_FRAME=1, any one of slot is valid VALID_FRAME=0, no any slot is valid The VALID_FRAME bits are read/write
[3:0]	SLOT_VALID [3:0]	Slot valid indicated bits SLOT_VALID[0]= 1/0, indicate Slot 1 valid/invalid SLOT_VALID[1]= 1/0, indicate Slot 2 valid/invalid SLOT_VALID[2]= 1/0, indicate Slot 3 valid/invalid SLOT_VALID[3]= 1/0, indicate Slot 4 valid/invalid The SLOT_VALID[3:0] bits are read/write

The AC-link output slot 1 (ACTL_ACOS1)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
ACTL_ACOS1	0xFFF0_9034	R/W	AC-link out slot 1	0x0000_0080

The ACTL_ACOS1 register store the slot 1 value to be shift out by AC-link.

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**
R_WB	**R_INDEX[6:0]**						

BITS	DESCRIPTIONS	
[31:8]	Reserved	-
[7]	R_WB	Read/Write select bit R_WB=1, a read specified by R_INDEX[6:0] will occur, and the data will appear in next frame R_WB=0, a write specified by R_INDEX[6:0] will occur, and the write data is put at out slot 2 The R_WB bit is read/write
[6:0]	R_INDEX[6:0]	External AC97 CODEC control register index (address) bits The R_INDEX[6:0] bits are read/write

AC-link output slot 2 (ACTL_ACOS2)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
ACTL_ACOS2	0xFFF0_9038	R/W	AC-link out slot 2	0x0000_0000

The ACTL_ACOS2 register store the slot 2 value to be shift out by AC-link.

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
WD[15:8]							
7	6	5	4	3	2	1	0
WD[7:0]							

BITS	DESCRIPTIONS	
[31:0]	Reserved	-
[15:0]	WD[15:0]	AC-link write data The WD[15:0] bits are read/write

AC-link input slot 0 (ACTL_ACIS0)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
ACTL_ACIS0	0xFFF0_903C	R	AC-link in slot 0	0x0000_0000

The ACTL_ACIS0 store the shift in slot 0 data of AC-link.

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			**CODEC_READY**	**SLOT_VALID[3:0]**			

BITS	DESCRIPTIONS	
[31:5]	Reserved	-
[4]	CODEC_READY	External AC97 audio CODEC ready bit CODEC_READY=0, indicate external AC97 audio CODEC is not ready CODEC_READY=1, indicate external AC97 audio CODEC is ready The CODEC_READY bit is read only
[3:0]	SLOT_VALID[3:0]	Slot valid indicated bits SLOT_VALID[0]= 1/0, indicate Slot 1 valid/invalid SLOT_VALID[1]= 1/0, indicate Slot 2 valid/invalid SLOT_VALID[2]= 1/0, indicate Slot 3 valid/invalid SLOT_VALID[3]= 1/0, indicate Slot 4 valid/invalid The SLOT_VALID[3:0] bits are read

AC-link input slot 1 (ACTL_ACIS1)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
ACTL_ACIS1	0xFFF0_9040	R	AC-link in slot 1	0x0000_0000

The ACTL_ACIS1 stores the shift in slot 1 data of AC-link.

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							R_INDEX[6]
7	6	5	4	3	2	1	0
R_INDEX[5:0]						SLOT_REQ[1:0]	

BITS	DESCRIPTIONS	
[31:9]	Reserved	-
[8:2]	R_INDEX[6:0]	Register index. The R_INDEX[6:0] echo the register index (address) when a register read has been requested in the previous frame. The R_INDEX[6:0] bits are read only
[1:0]	SLOT_REQ[1:0]	Slot request. The bits indicate if the external codec need new PCM data that will transfer in next frame. Any bit in SLOT_REQ[1:0] is set to 1, indicate external codec does not need a new sample in the corresponding slot[3:4] of the next frame Any SLOT_REQ[1:0] is clear to 0, indicate external codec need a new sample in the corresponding slot[3:4] of the next frame The SLOT_REQ[1:0] bits are read only

AC-link input slot 2 (ACTL_ACIS2)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
ACTL_ACIS2	0xFFF0_9044	R	AC-link in slot 2	0x0000_0000

The ACTL_ACIS2 stores the shift in slot 2 data of AC-link.

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
RD[15:8]							
7	6	5	4	3	2	1	0
RD[7:0]							

BITS	DESCRIPTIONS	
[31:16]	Reserved	-
[15:0]	RD[15:0]	AC-link read data. The RD[15:0] bits are read only

6.10 Universal Asynchronous Receiver/Transmitter Controller

Asynchronous serial communication block include 4 **UART** blocks and accessory logic. They can be described as follow:

- **UART0**

It is merely a general purpose UART. It does not include any accessory function.

Clock Source	: 15MHz
UART Type	: general UART
FIFO Number	: 16-byte receiving FIFO and 16 byte transmitting FIFO
Modem Function	: N/A
Accessory Function	: N/A

- **UART1**

It is designed for general purpose UART or Bluetooth transceiver. It includes a high speed UART block with 64-byte receiving FIFO and 64-byte transmitting FIFO. It includes 5 clock sources: 15M, 30M, 43.6M, 48M and 60M. Programmer can feel free to choose the clock source and divisor number for suitable baud rate.

Clock Source	: 15MHz from external crystal 30M, 43.6M, 48M, 60M (optional function for Bluetooth HCI transport layer)
UART Type	: high speed UART
FIFO Number	: 64-byte receiving FIFO and 64 byte transmitting FIFO
Modem Function	: CTS and RTS (optional for Bluetooth. If they were enabled, TX & RX in UART2 will be cut off)
Accessory Function	: Bluetooth (optional)
Baud Rate (max)	: 1.875MHz
I/O pin	: TXD1, RXD1, RTS, CTS (optional)

- **UART2**

It is designed for general purpose UART or IrDA SIR. The part of UART includes 16-byte receiving FIFO and 16-byte transmitting FIFO. TXD2/RXD2 of UART2 occupy the same pins with RTS and CTS of UART1. Once the Bluetooth function has been enabled, UART2 should be disabled.

Clock Source	: 15MHz
UART Type	: general UART

FIFO Number : 16-byte receiving FIFO and 16 byte transmitting FIFO
Modem Function : N/A
Accessory Function : IrDA SIR (optional)
I/O Pin : TXD2, RXD2.
I/O Pin Share with : UART1 (Bluetooth function)

- **UART3**

It is also merely a general purpose UART. It does not include any accessory function. It share four I/O pins with AC97/I²S.

Clock Source : 15MHz
UART Type : general UART
FIFO Number : 16-byte receiving FIFO and 16 byte transmitting FIFO
Modem Function : DTR, DSR
Accessory Function : N/A
I/O Pin : TXD3, RXD3, DTR, DSR
I/O Pin Share with : AC97_DATAO, AC97_DATAI, AC97_SYNC, AC97_BITCLK

Table 6.10.1 W90N745 UART features list

BLOCK NUMBER	UART TYPE	CLOCK SOURCE	MODEM FUNCTION SIGNALS	IO PINS	DESIGN TARGET
0	General UART	15M	N/A	TxD0, RXD0	General UART
1	High speed UART	15M, 30M, 43.6M, 48M, 60M	CTS, RTS	TXD1, RXD1, CTS1, RTS1	General UART/ Bluetooth
2	General UART	15M	N/A	TX2, RX2	General UART/IrDA SIR
3	General UART	15M	DTR, DSR	TXD3, RXD3, DRT3, DSR3	General UART

6.10.1 UART0

UART0 is a general UART block. It is same as the UART in W90N740 but without Modem I/O signals. More detail function description, please refer to section 7.10.5 **General UARTcontroller description**

Table 6.10.2 UART0 Register Map

REGISTER	ADDRESS	R/W	OTHER CONDITION	RESET VALUE
UART0_RBR	0xFFF8_0000	R	DLAB=0	Undefined
UART0_THR	0xFFF8_0000	W	DLAB=0	Undefined
UART0_IER	0xFFF8_0004	R/W	DLAB=0	0x0000_0000
UART0_DLL	0xFFF8_0000	R/W	DLAB=1	0x0000_0000
UART0_DLM	0xFFF8_0004	R/W	DLAB=1	0x0000_0000
UART0_IIR	0xFFF8_0008	R		0x8181_8181
UART0_FCR	0xFFF8_0008	W		Undefined
UART0_LCR	0xFFF8_000c	R/W		0x0000_0000
Reserved	0xFFF8_0010			
UART0_LSR	0xFFF8_0014	R		0x6060_6060
Reserved	0xFFF8_0018			
UART0_TOR	0xFFF8_001c	R/W		0x0000_0000

6.10.2 UART1

The UART1 is designed for general purpose UART or Bluetooth HCI transport layer. It is a high speed UART with 64-byte receive FIFO and 64-byte transmit FIFO. To perform 1.875MHz maximum baud rate, UART1 has 5 clock sources, 15M, 30M, 43.6M, 48M, and 60M. The first one is from external 15M crystal clock and the other are divided from system PLL 480MHz output. More detail about high speed UART, please refer to next section 7.10.6 **High Speed UART controller function description.**

The block UART1 offer 4 I/O signals, TX, RX, CTS, and RTS. CTS and RTS are used as flow control for Bluetooth. CTS and RTS share the same I/O pins with TX and RX in block UART2.

Table 6.10.3 UART1 Register Map

REGISTER	ADDRESS	R/W	OTHER CONDITION	RESET VALUE
UART1_RBR	0xFFF8_0100	R	DLAB=0	Undefined
UART1_THR	0xFFF8_0100	W	DLAB=0	Undefined
UART1_IER	0xFFF8_0104	R/W	DLAB=0	0x0000_0000
UART1_DLL	0xFFF8_0100	R/W	DLAB=1	0x0000_0000
UART1_DLM	0xFFF8_0104	R/W	DLAB=1	0x0000_0000
UART1_IIR	0xFFF8_0108	R		0x8181_8181
UART1_FCR	0xFFF8_0108	W		Undefined
UART1_LCR	0xFFF8_010c	R/W		0x0000_0000
UART1_MCR	0xFFF8_0110	R/W		0x0000_0000
UART1_LSR	0xFFF8_0114	R		0x6060_6060
UART1_MSR	0xFFF8_0118	R		0x0000_0000
UART1_TOR	0xFFF8_011c	R/W		0x0000_0000
UART1_UBCR	0xFFF8_0120	R/W		0x0000_0000

UART1 Bluetooth Control Register (UART1_UBCR)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
UART1_UBCR	0xFFF8_0120	R/W	UART 1 Bluetooth 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					UBCR[2:0]		

BITS	DESCRIPTIONS	
[31:3]	**Reserved**	-
[2:0]	**UBCR**	UBCR is a 3 bits register which is used to select clock source to generate suitable baud rate: 000: 15Mhz from external crystal 100: 30Mhz divided from PLL 480Mhz 101: 43.6Mhz divided from PLL 480Mhz 110: 48Mhz divided from PLL 480Mhz 111: 60Mhz divided from PLL 480Mhz

6.10.3 UART2

UART2 contains 2 features: general UART and IrDA SIR decoder/encoder. UART is same as the UART of W90N740 but without modem function. Please read the spec of section 7.10.5 **General UART controller function description.** The IrDA SIR is described as follow:

Table 6.10.4 UART2 Register Map

REGISTER	ADDRESS	R/W	OTHER CONDITION	RESET VALUE
UART2_RBR	0xFFF8_0200	R	DLAB=0	Undefined
UART2_THR	0xFFF8_0200	W	DLAB=0	Undefined
UART2_IER	0xFFF8_0204	R/W	DLAB=0	0x0000_0000
UART2_DLL	0xFFF8_0200	R/W	DLAB=1	0x0000_0000
UART2_DLM	0xFFF8_0204	R/W	DLAB=1	0x0000_0000
UART2_IIR	0xFFF8_0208	R		0x8181_8181
UART2_FCR	0xFFF8_0208	W		Undefined
UART2_LCR	0xFFF8_020c	R/W		0x0000_0000
Reserved	0xFFF8_0210			Undefined
UART2_LSR	0xFFF8_0214	R		0x6060_6060
Reserved	0xFFF8_0218			Undefined
UART2_TOR	0xFFF8_021c	R/W		0x0000_0000
UART2_IRCR	0xFFF8_0220	R/W		0x0000_0040

UART2 IrDA Control Register (UART2_IRCR)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
UART2_IRCR	0xFFF8_0220	R/W	UART 2 IrDA Control 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	INV_RX	INV_TX	Reserved	Reserved	LB	TX_SELECT	IrDA_EN

BITS	DESCRIPTIONS	
[31:7]	**Reserved**	**Reserved**
[6]	**INV_RX**	1: Inverse RX input signal 0: No inversion
[5]	**INV_TX**	1: Inverse TX output signal 0: No inversion
[4:3]	**Reserved**	**Reserved**
[2]	**LB**	**IrDA loop back mode for self test.** 1: enable IrDA loop back mode 0: disable IrDA loop back mode
[1]	**TX_SELECT**	1: enable IrDA transmitter 0: enable IrDA receiver
[0]	**IrDA_EN**	1: enable IrDA block 0: disable IrDA block

6.10.4 UART3

UART3 is a general UART block. It is same as the UART in W90N740 but with some Modem I/O signals.

More detail general UART function description, please refer to next section 7.10.5 General UART controller.

Table 6.10.5 UART3 register map

REGISTER	ADDRESS	R/W	OTHER CONDITION	RESET VALUE
UART3_RBR	0xFFF8_0300	R	DLAB=0	Undefined
UART3_THR	0xFFF8_0300	W	DLAB=0	Undefined
UART3_IER	0xFFF8_0304	R/W	DLAB=0	0x0000_0000
UART3_DLL	0xFFF8_0300	R/W	DLAB=1	0x0000_0000
UART3_DLM	0xFFF8_0304	R/W	DLAB=1	0x0000_0000
UART3_IIR	0xFFF8_0308	R		0x8181_8181
UART3_FCR	0xFFF8_0308	W		Undefined
UART3_LCR	0xFFF8_030c	R/W		0x0000_0000
UART3_MCR	0xFFF8_0310	R/W		0x0000_0000
UART3_LSR	0xFFF8_0314	R		0x6060_6060
UART3_MSR	0xFFF8_0318	R		0x0000_0000
UART3_TOR	0xFFF8_031c	R/W		0x0000_0000

UART3 Modem Control Register (UART3_MCR)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
UART3_MCR	0xFFF8_0310	R/W	UART 3 Modem 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	**Reserved**	**Reserved**	**LBME**	**Reserved**	**Reserved**	**Reserved**	**DTR#**

Note: UART3_MCR is subset of MCR in W90N745. Please refer to section 7.10.5 'General UART Controller'.

UART3 Modem Status Register (UART3_MSR)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
UART3_MSR	0xFFF8_0318	R	UART 3 Modem 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	Reserved	DSR#	Reserved	Reserved	Reserved	DDSR	Reserved

Note: UART3_MSR is subset of MSR in W90N745. Please refer to section 7.10.5 'General UART Controller'.

6.10.5 General UART Controller

The **Universal Asynchronous Receiver/Transmitter (UART)** performs a serial-to-parallel conversion on data characters received from the peripheral such as MODEM, and a parallel-to-serial conversion on data characters received from the CPU. There are five types of interrupts, i.e., **line status interrupt, transmitter FIFO empty interrupt, receiver threshold level reaching interrupt, time out interrupt, and MODEM status interrupt.** One 16-byte transmitter FIFO **(TX_FIFO)** and one 16-byte (plus 3-bit of error data per byte) receiver FIFO **(RX_FIFO)** has been built in to reduce the number of interrupts presented to the CPU. The CPU can completely read the status of the UART at any time during the operation. The reported status information includes the type and condition of the transfer operations being performed by the UART, as well as any error conditions (parity, overrun, framing, or break interrupt) found. The UART includes a programmable baud rate generator that is capable of dividing crystal clock input by divisors to produce the clock that transmitter and receiver needed. The equation is

BaudOut = crystal clock / 16 * [Divisor + 2].

The UART includes the following features:

- Transmitter and receiver are buffered with a 16-byte FIFO each to reduce the number of interrupts presented to the CPU.
- **Subset of MODEM control functions (DSR, DTR, by IP selection)**
- Fully programmable serial-interface characteristics:
  - -- 5-, 6-, 7-, or 8-bit character
  - -- Even, odd, or no-parity bit generation and detection
  - -- 1-, 1&1/2, or 2-stop bit generation
  - -- Baud rate generation

- Line break generation and detection
- False start bit detection
- Full prioritized interrupt system controls
- Loop back mode for internal diagnostic testing

6.10.5.1. 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	RESET VALUE
UART_RBR	0x00	R	Receive Buffer Register (DLAB = 0)	Undefined
UART_THR	0x00	W	Transmit Holding Register (DLAB = 0)	Undefined
UART_IER	0x04	R/W	Interrupt Enable Register (DLAB = 0)	0x0000_0000
UART_DLL	0x00	R/W	Divisor Latch Register (LS) (DLAB = 1)	0x0000_0000
UART_DLM	0x04	R/W	Divisor Latch Register (MS) (DLAB = 1)	0x0000_0000
UART_IIR	0x08	R	Interrupt Identification Register	0x8181_8181
UART_FCR	0x08	W	FIFO Control Register	Undefined
UART_LCR	0x0C	R/W	Line Control Register	0x0000_0000
UART_MCR	0x10	R/W	Modem Control Register (Optional)	0x0000_0000
UART_LSR	0x14	R	Line Status Register	0x6060_6060
UART_MSR	0x18	R	MODEM Status Register (Optional)	0x0000_0000
UART_TOR	0x1C	R/W	Time Out Register	0x0000_0000

Note: Real register address = 0xFFF8_0000+ (UART number – 1) * (0x0100) + offset

Note: All of these registers are implemented 8-bit in UART design and it will be repeated 4 times before send to APB bus. For example, when ARM CPU read register UARTn_BRR, ARM CPU will get UART0_RBR = {RBR[7:0], RBR[7:0], RBR[7:0], RBR[7:0]}.

UART Receive Buffer Register (UART_RBR)

REGISTER	OFFSET	R/W	DESCRIPTION	RESET VALUE
UART_RBR	0x00	R	Receive Buffer Register (DLAB = 0)	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							
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).

UART Transmit Holding Register (UART_THR)

REGISTER	OFFSET	R/W	DESCRIPTION	RESET VALUE
UART_THR	0x00	W	Transmit Holding Register (DLAB = 0)	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							
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).

UART Interrupt Enable Register (UART_IER)

REGISTER	OFFSET	R/W	DESCRIPTION	RESET VALUE
UART_IER	0x04	R/W	Interrupt Enable Register (DLAB = 0)	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			nDBGACK_EN	MSIE	RLSIE	THREIE	RDAIE

BITS	DESCRIPTIONS	
[31:5]	Reserved	-
[4]	nDBGACK_EN	**ICE debug mode acknowledge enable** 0 = When DBGACK is high, the UART receiver time-out clock will be held 1 = No matter what DBGACK is high or not, the UART receiver timer-out clock will not be held
[3]	MSIE	**MODEM Status Interrupt (Irpt_MOS) Enable** 0 = Mask off Irpt_MOS 1 = Enable Irpt_MOS
[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

UART Divider Latch (Low Byte) Register (UART_DLL)

REGISTER	OFFSET	R/W	DESCRIPTION	RESET VALUE
UART_DLL	0x00	R/W	Divisor Latch Register (LS) (DLAB = 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
Baud Rate Divider (Low Byte)							

BITS	DESCRIPTIONS	
[7:0]	Baud Rate Divider (Low Byte)	The low byte of the baud rate divider

UART Divisor Latch (High Byte) Register (UART_DLM)

REGISTER	OFFSET	R/W	DESCRIPTION	RESET VALUE
UART_DLM	0x04	R/W	Divisor Latch Register (MS) (DLAB = 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
Baud Rate Divider (High Byte)							

BITS	DESCRIPTIONS	
[7:0]	**Baud Rate Divider (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

UART Interrupt Identification Register (UART_IIR)

REGISTER	OFFSET	R/W	DESCRIPTION	RESET VALUE
UART_IIR	0x08	R	Interrupt Identification Register	0x8181_8181

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
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 enabling, 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.

Table 6.10.6 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	MODEM Status (Irpt_MOS)	The CTS, DSR, or DCD bits are changing state or the RI bit is changing from high to low.	Reading the MSR **(optional)**

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

UART FIFO Control Register (UART_FCR)

REGISTER	OFFSET	R/W	DESCRIPTION	RESET VALUE
UART_FCR	0x08	W	FIFO Control 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							
7	6	5	4	3	2	1	0
RFITL		RESERVED		DMS	TFR	RFR	FME

<table>
<tr><th>BITS</th><th colspan="2">DESCRIPTIONS</th></tr>
<tr><td>[7:6]</td><td>RFITL</td><td>RX FIFO Interrupt (Irpt_RDA) Trigger Level
<table>
<tr><th>RFITL [7:6]</th><th>Irpt_RDA Trigger Level (Bytes)</th></tr>
<tr><td>00</td><td>01</td></tr>
<tr><td>01</td><td>04</td></tr>
<tr><td>10</td><td>08</td></tr>
<tr><td>11</td><td>14</td></tr>
</table></td></tr>
<tr><td>[3]</td><td>DMS</td><td>DMA Mode Select
The DMA function is not implemented in this version.</td></tr>
<tr><td>[2]</td><td>TFR</td><td>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.</td></tr>
<tr><td>[1]</td><td>RFR</td><td>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.</td></tr>
<tr><td>[0]</td><td>FME</td><td>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.</td></tr>
</table>

UART Line Control Register (UART_LCR)

REGISTER	OFFSET	R/W	DESCRIPTION	RESET VALUE
UART_LCR	0x0C	R/W	Line 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
DLAB	BCB	SPE	EPE	PBE	NSB	WLS	

<table>
<tr><th>BITS</th><th colspan="2">DESCRIPTIONS</th></tr>
<tr><td>[7]</td><td>DLAB</td><td>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}</td></tr>
<tr><td>[6]</td><td>BCB</td><td>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.</td></tr>
<tr><td>[5]</td><td>SPE</td><td>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.</td></tr>
<tr><td>[4]</td><td>EPE</td><td>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.</td></tr>
<tr><td>[3]</td><td>PBE</td><td>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.</td></tr>
<tr><td>[2]</td><td>NSB</td><td>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.</td></tr>
<tr><td>[1:0]</td><td>WLS</td><td>Word Length Select
<table>
<tr><th>WLS[1:0]</th><th>Character length</th></tr>
<tr><td>00</td><td>5 bits</td></tr>
<tr><td>01</td><td>6 bits</td></tr>
<tr><td>10</td><td>7 bits</td></tr>
<tr><td>11</td><td>8 bits</td></tr>
</table></td></tr>
</table>

UART Modem Control Register (UART_MCR)

REGISTER	OFFSET	R/W	DESCRIPTION	RESET VALUE
UART_MCR	0x10	R/W	Modem Control Register **(Optional)**	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			**LBME**	**Reserve**	**Reserve**	**Reserved**	**DTR#**

BITS	DESCRIPTIONS	
[31:5]	**Reserved**	-
[4]	**LBME**	**Loop-back Mode Enable** 0 = Disable 1 = When the loop-back mode is enabled, the following signals are connected internally SOUT connected to SIN and SOUT pin fixed at logic 1 DTR# connected to DSR# and DTR# pin fixed at logic 1
[3:1]	**Reserved**	**-**
[0]	**DTR**	**Complement version of DTR# (Data-Terminal-Ready) signal** Writing 0x00 to MCR, the DTR# bit are set to logic 1's; Writing 0x0f to MCR, the DTR# bit are reset to logic 0's.

UART Line Status Control Register (UART_LSR)

REGISTER	OFFSET	R/W	DESCRIPTION	RESET VALUE
UART_LSR	0x14	R	Line Status Register	0x6060_6060

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
ERR_RX	TE	THRE	BII	FEI	PEI	OEI	RFDR

BITS	DESCRIPTIONS	
[31:8]	Reserved	-
[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**.**

Continued.

BITS	DESCRIPTIONS	
[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)

UART Modem Status Register (UART_MSR)

REGISTER	OFFSET	R/W	DESCRIPTION	RESET VALUE
UART_MSR	0x18	R	MODEM Status Register **(Optional)**	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	Reserved	DSR#	Reserved	Reserved	Reserved	DDSR	Reserved

BITS	DESCRIPTIONS	
[31:6]	Reserved	-
[5]	DSR#	**Complement version of data set ready (DSR#) input** **(This bit is selected by IP)**
[4:2]	Reserved	**-**
[1]	DDSR	**DSR# State Change (This bit is selected by IP)** This bit is set whenever DSR# input has changed state, and it will be reset if the CPU reads the MSR.
[0]	Reserved	**-**

Whenever any of MSR [3:0] is set to logic 1, a Modem Status Interrupt is generated if IER[3]=1. Writing MSR is a null operation (not suggested).

UART Time Out Register (UART_TOR)

REGISTER	OFFSET	R/W	DESCRIPTION	RESET VALUE
UART_TOR	0x1C	R/W	Time Out 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
TOIE	TOIC						

BITS	DESCRIPTIONS	
[31:8]	Reserved	-
[7]	TOIE	**Time Out Interrupt Enable** The feature of receiver time out interrupt is enabled 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.

6.10.6 High speed UART Controller

The **High Speed Universal Asynchronous Receiver/Transmitter (HS_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 types of interrupts, they are, transmitter FIFO empty interrupt, receiver threshold level reaching interrupt**,** line status interrupt (overrun error or parity error or framing error or break interrupt) ,time out interrupt, and Modem status interrupt **.** One 64-byte transmitter FIFO **(TX_FIFO)** and one 64-byte (plus 3-bit of error data per byte) receiver FIFO **(RX_FIFO)** has been built in to reduce the number of interrupts presented to the CPU. The CPU can completely read the status of the UART at any time during the operation. The reported status information includes the type and condition of the transfer operations being performed by the UART, as well as any error conditions (parity, overrun, framing, or break interrupt) found. The UART includes a programmable baud rate generator that is capable of dividing crystal clock input by divisors to produce the clock that transmitter and receiver needed. The equation is

Baud Out = crystal clock / 16 * [Divisor + 2].

The UART includes the following features:

- Transmitter and receiver are buffered with a 64-byte FIFO each to reduce the number of interrupts presented to the CPU.
- Subset of MODEM control function(selected by IP)
- Fully programmable serial-interface characteristics:
  - 5-, 6-, 7-, or 8-bit character
  - Even, odd, or no-parity bit generation and detection
  - 1-, 1&1/2, or 2-stop bit generation
  - Baud rate generation
- False start bit detection
- Full-prioritized interrupt system controls
- Not support Loop back mode

6.10.6.1. High Speed 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	RESET VALUE
HSUART_RBR	0x00	R	Receive Buffer Register (DLAB = 0)	Undefined
HSUART_THR	0x00	W	Transmit Holding Register (DLAB = 0)	Undefined
HSUART_IER	0x04	R/W	Interrupt Enable Register (DLAB = 0)	0x0000_0000
HSUART_DLL	0x00	R/W	Divisor Latch Register (LS)(DLAB = 1)	0x0000_0000
HSUART_DLM	0x04	R/W	Divisor Latch Register (MS)(DLAB = 1)	0x0000_0000

Continued.

REGISTER	OFFSET	R/W	DESCRIPTION	RESET VALUE
HSUART_IIR	0x08	R	Interrupt Identification Register	0x8181_8181
HSUART_FCR	0x08	W	FIFO Control Register	Undefined
HSUART_LCR	0x0C	R/W	Line Control Register	0x0000_0000
HSUART_MCR	0x10	R/W	Modem Control Register **(Optional)**	0x0000_0000
HSUART_LSR	0x14	R	Line Status Register	0x6060_6060
HSUART_MSR	0x18	R	MODEM Status Register **(Optional)**	0x0000_0000
HSUART_TOR	0x1C	R/W	Time Out Register	0x0000_0000

Note: Real register address = 0xFFF8_0000+ (UART number – 1) * (0x0100) + offset

NOTE: All of these registers are implemented 8-bit in UART design and it will be repeated 4 times before send to APB bus. For example, when ARM CPU read register UART1_BRR, ARM CPU will get UART1_RBR = {RBR[7:0], _RBR[7:0], RBR[7:0], RBR[7:0]}.

HSUART Receive Buffer Register (HSUART_RBR)

REGISTER	OFFSET	R/W	DESCRIPTION	RESET VALUE
HSUART_RBR	0x00	R	Receive Buffer Register (DLAB = 0)	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							
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).

HSUART Transmit Holding Register (HSUART_THR)

REGISTER	OFFSET	R/W	DESCRIPTION	RESET VALUE
HSUART_THR	0x00	W	Transmit Holding Register (DLAB = 0)	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							
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).

HSUART Interrupt Enable Register (HSUART_IER)

REGISTER	OFFSET	R/W	DESCRIPTION	RESET VALUE
HSUART_IER	0x04	R/W	Interrupt Enable Register (DLAB = 0)	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			nDBGACK_EN	MSIE	RLSIE	THREIE	RDAIE

BITS	DESCRIPTIONS	
[31:5]	**Reserved**	-
[4]	**nDBGACK_EN**	**ICE debug mode acknowledge enable** 0 = When DBGACK is high, the UART receiver time-out clock will be held 1 = No matter what DBGACK is high or not, the UART receiver timer-out clock will not be held
[3]	**MSIE**	**MODEM Status Interrupt (Irpt_MOS) Enable** 0 = Mask off Irpt_MOS 1 = Enable Irpt_MOS
[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

HSUART Divider Latch (Low Byte) Register (HSUART_DLL)

REGISTER	OFFSET	R/W	DESCRIPTION	RESET VALUE
HSUART_DLL	0x00	R/W	Divisor Latch Register (LS) (DLAB = 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**
Baud Rate Divider (Low Byte)							

BITS	DESCRIPTIONS	
[31:8]	**Reserved**	-
[7:0]	**Baud Rate Divisor (Low Byte)**	The low byte of the baud rate divider

HSUART Divisor Latch (High Byte) Register (HSUART_DLM)

REGISTER	OFFSET	R/W	DESCRIPTION	RESET VALUE
HSUART_DLM	0x04	R/W	Divisor Latch Register (MS) (DLAB = 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**
Baud Rate Divider (High Byte)							

BITS	DESCRIPTIONS	
[31:8]	**Reserved**	
[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]}

HSUART Interrupt Identification Register (HSUART_IIR)

REGISTER	OFFSET	R/W	DESCRIPTION	RESET VALUE
HSUART_IIR	0x08	R	Interrupt Identification Register	0x8181_8181

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**
FMES	**RFTLS**		**DMS**	**IID**			**NIP**

BITS	DESCRIPTIONS	
[31:8]	**Reserved**	**-**
[7]	**FMES**	**FIFO Mode Enable Status** This bit indicates whether the FIFO mode is enabled or not. Since the FIFO mode is always enable, 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	MODEM Status (Irpt_MOS)	The CTS bits are changing state .	Reading the MSR **(optional)**

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

HSUART FIFO Control Register (HSUART_FCR)

REGISTER	OFFSET	R/W	DESCRIPTION	RESET VALUE
HSUART_FCR	0x08	W	FIFO Control 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							
7	6	5	4	3	2	1	0
RFITL				DMS	TFR	RFR	FME

<table>
<tr><th>BITS</th><th colspan="2">DESCRIPTIONS</th></tr>
<tr><td>[31:8]</td><td>Reserved</td><td>-</td></tr>
<tr><td>[7:4]</td><td>RFITL</td><td>RX FIFO Interrupt (Irpt_RDA) Trigger Level
<table>
<tr><th>RFITL</th><th>Irpt_RDA Trigger Level (Bytes)</th></tr>
<tr><td>0000</td><td>01</td></tr>
<tr><td>0001</td><td>04</td></tr>
<tr><td>0010</td><td>08</td></tr>
<tr><td>0011</td><td>14</td></tr>
<tr><td>0100</td><td>30</td></tr>
<tr><td>0101</td><td>46</td></tr>
<tr><td>0110</td><td>62</td></tr>
<tr><td>others</td><td>62</td></tr>
</table></td></tr>
<tr><td>[3]</td><td>DMS</td><td>DMA Mode Select
The DMA function is not implemented in this version.</td></tr>
<tr><td>[2]</td><td>TFR</td><td>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.</td></tr>
<tr><td>[1]</td><td>RFR</td><td>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.</td></tr>
<tr><td>[0]</td><td>FME</td><td>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.</td></tr>
</table>

HSUART Line Control Register (HSUART_LCR)

REGISTER	OFFSET	R/W	DESCRIPTION	RESET VALUE
HSUART_LCR	0x0C	R/W	Line Control Register	0x0000_0000

BITS	DESCRIPTIONS	
[31:8]	**Reserved**	-
[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

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
DLAB	BCB	SPE	EPE	PBE	NSB	WLS	

HSUART Modem Control Register (HSUART_MCR)

REGISTER	OFFSET	R/W	DESCRIPTION	RESET VALUE
HSUART_MCR	0x10	R/W	Modem Control Register **(Optional)**	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			LBME	Reserved		RTS	Reserved

BITS	DESCRIPTIONS	
[31:5]	**Reserved**	-
[4]	**LBME**	**Loop-back Mode Enable** 0 = Disable 1 = When the loop-back mode is enabled, the following signals are connected internally: SOUT connected to SIN and SOUT pin fixed at logic 1 RTS# connected to CTS# and RTS# pin fixed at logic 1

Continued.

BITS	DESCRIPTIONS	
[3:2]	**Reserved**	**-**
[1]	**RTS#**	**Complement version of RTS# (Request-To-Send) signal** Writing 0x00 to MCR, RTS# bit are set to logic 1's; Writing 0x0f to MCR, RTS# bit are reset to logic 0's.
[0]	**Reserved**	**-**

HSUART Line Status Control Register (HSUART_LSR)

REGISTER	OFFSET	R/W	DESCRIPTION	RESET VALUE
HSUART_LSR	0x14	R	Line Status Register	0x6060_6060

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**
ERR_RX	**TE**	**THRE**	**BII**	**FEI**	**PEI**	**OEI**	**RFDR**

BITS	DESCRIPTIONS	
[31:8]	**Reserved**	
[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.

Continued.

BITS	DESCRIPTIONS	
[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).

HSUART Modem Status Register (HSUART_MSR)

REGISTER	OFFSET	R/W	DESCRIPTION	RESET VALUE
HSUART_MSR	0x18	R	MODEM Status Register **(Optional)**	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			**CTS#**	Reserved			**DCTS**

BITS	DESCRIPTIONS	
[31:5]	**Reserved**	**-**
[4]	**CTS#**	**Complement version of clear to send (CTS#) input** **(This bit is selected by IP)**
[3:1]	**Reserved**	**-**
[0]	**DCTS**	**CTS# State Change** **(This bit is selected by IP)** This bit is set whenever CTS# input has changed state, and it will be reset if the CPU reads the MSR.

Whenever any of MSR [0] is set to logic 1, a Modem Status Interrupt is generated if IER[3]=1. Writing MSR is a null operation (not suggested).

HSUART Time Out Register (HSUART_TOR)

REGISTER	OFFSET	R/W	DESCRIPTION	RESET VALUE
HSUART_TOR	0x1C	R/W	Time Out 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
TOIE	TOIC						

BITS	DESCRIPTIONS	
[31:8]	**Reserved**	-
[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.

6.11 Timer/Watchdog Controller

6.11.1 General Timer Controller

The timer module includes two channels, TIMER0 and TIMER1, which allow you to easily implement a counting scheme for use. The timer can perform functions like frequency measurement, event counting, interval measurement, clock 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

- AMBA APB interface compatible
- Two channels with a 8-bit presale counter/24-bit down counter and an interrupt request each
- Independent clock source for each channel
- Maximum uninterrupted time = (1 / 25 MHz) * (256) * (2^24), if TCLK = 25 MHz

6.11.2 Watchdog Timer

6.11.3 Timer Control Registers Map

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

REGISTER	ADDRESS	R/W/C	DESCRIPTION	RESET VALUE
TCSR0	0xFFF8_1000	R/W	Timer Control and Status Register 0	0x0000_0005
TCSR1	0xFFF8_1004	R/W	Timer Control and Status Register 1	0x0000_0005
TICR0	0xFFF8_1008	R/W	Timer Initial Control Register 0	0x0000_0000
TICR1	0xFFF8_100C	R/W	Timer Initial Control Register 1	0x0000_0000
TDR0	0xFFF8_1010	R	Timer Data Register 0	0x0000_0000
TDR1	0xFFF8_1014	R	Timer Data Register 1	0x0000_0000
TISR	0xFFF8_1018	R/W	Timer Interrupt Status Register	0x0000_0000
WTCR	0xFFF8_101C	R/W	Watchdog Timer Control Register	0x0000_0400

Timer Control Register 0/1 (TCSR 0/1)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
TCSR0	0xFFF8_1000	R/W	Timer Control and Status Register 0	0x0000_0005
TCSR1	0xFFF8_1004	R/W	Timer Control and Status Register 1	0x0000_0005

31	30	29	28	27	26	25	24
nDBGACK_EN	CEN	IE	MODE[1:0]		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[7:0]							

BITS	DESCRIPTIONS	
[31]	**nDBGACK_EN**	**ICE debug mode acknowledge enable** 0 = When DBGACK is high, the TIMER counter will be held 1 = No matter DBGACK is high or not, the TIMER counter will not be held
[30]	**CEN**	**Counter Enable** 0 = Stops/Suspends counting 1 = Starts counting
[29]	**IE**	**Interrupt Enable** 0 = **Disable** TIMER Interrupt. 1 = **Enable** TIMER Interrupt. If timer interrupt is enabled, the timer asserts its interrupt signal when the associated counter decrements to zero.
[28:27]	**MODE**	**Timer Operating Mode**

MODE	Timer Operating Mode
00	The timer is operating in the one-shot mode. The associated interrupt signal is generated once (if IE is enabled) and CEN is automatically cleared then.
01	The timer is operating in the periodic mode. The associated interrupt signal is generated periodically (if IE is enabled).
10	The timer is operating in the toggle mode. The interrupt signal is generated periodically (if IE is enabled). And the associated signal (tout) is changing back and forth with 50% duty cycle.
11	Reserved.

Continued

BITS	DESCRIPTIONS	
[26]	**CRST**	**Counter Reset** Set this bit will reset the TIMER counter, and also **force CEN to 0**. 0 = No effect. 1 = Reset Timer's prescale counter, internal 24-bit counter and CEN.
[25]	**CACT**	**Timer is in Active** This bit indicates the counter status of timer. 0 = Timer is **not** active. 1 = Timer is **in** active.
[24:8]	**Reserved**	**Reserved**
[7:0]	**PRESCALE**	**Prescale** Clock input is divided by PRESCALE+1 before it is fed to the counter. If PRESCALE=0, then there is no scaling.

Timer Initial Count Register 0/1 (TICR0/1)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
TICR0	0xFFF8_1008	R/W	Timer Initial Control Register 0	0x0000_0000
TICR1	0xFFF8_100C	R/W	Timer Initial Control Register 1	0x0000_0000

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	
[31:24]	**Reserved**	**Reserved**
[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. **NOTE1:** Never write 0x0 in TIC, or the core will run into unknown state. **NOTE2:** 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 0/1 (TDR0/1)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
TDR0	0xFFF8_10010	R	Timer Data Register 0	0x0000_0000
TDR1	0xFFF8_10014	R	Timer Data Register 1	0x0000_0000

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	
[31:24]	**Reserved**	**Reserved**
[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	DESCRIPTION	RESET VALUE
TISR	0xFFF8_1018	R/W	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						**TIF1**	**TIF0**

BITS	DESCRIPTIONS	
[1]	**TIF1**	**Timer Interrupt Flag 1** This bit indicates the interrupt status of Timer channel 1. 0 = It indicates that the Timer 1 dose not countdown to zero yet. 1 = It indicates that the counter of Timer 1 has decremented to zero. 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.
[0]	**TIF0**	**Timer Interrupt Flag 0** This bit indicates the interrupt status of Timer channel 0. 0 = It indicates that the Timer 0 dose not countdown to zero yet. 1 = It indicates that the counter of Timer 0 has decremented to zero. 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.

Watchdog Timer Control Register (WTCR)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
WTCR	0xFFF8_101C	R/W	Watchdog Timer Control Register	0x0000_0400

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					**WTCLK**	**nDBGACK_EN**	**WTTME**
7	6	5	4	3	2	1	0
WTE	**WTIE**	**WTIS**		**WTIF**	**WTRF**	**WTRE**	**WTR**

BITS	DESCRIPTIONS	
[31:11]	**Reserved**	**Reserved**
[10]	**WTCLK**	**Watchdog Timer Clock** This bit is used for deciding whether the Watchdog timer clock input is divided by 256 or not. Clock source of Watchdog timer is Crystal input. 0 = Using original clock input 1 = The clock input will be divided by 256 **NOTE:** When WTTME = 1, set this bit has no effect on WDT clock (using original clock input).
[9]	**nDBGACK_EN**	**ICE debug mode acknowledge enable** 0 = When DBGACK is high, the Watchdog timer counter will be held 1 = No matter DBGACK is high or not, the Watchdog timer counter will not be held
[8]	**WTTME**	**Watchdog Timer Test Mode Enable** For reasons of efficiency, the 26-bit counter within the Watchdog timer is considered as two independent 13-bit counters in the test mode. They are operated concurrently and separately during the test. This approach can save a lot of time spent in the test. When the 13-bit counter overflows, a Watchdog timer interrupt is generated. 0 = Put the Watchdog timer in normal operating mode 1 = Put the Watchdog timer in test mode
[7]	**WTE**	**Watchdog Timer Enable** 0 = Disable the Watchdog timer (This action will reset the internal counter) 1 = Enable the Watchdog timer

Continued

<table>
<tr><th>BITS</th><th colspan="2">DESCRIPTIONS</th></tr>
<tr><td>[6]</td><td>WTIE</td><td>Watchdog Timer Interrupt Enable
0 = Disable the Watchdog timer interrupt
1 = Enable the Watchdog timer interrupt</td></tr>
<tr><td>[5:4]</td><td>WTIS</td><td>Watchdog Timer Interval Select
These two bits select the interval for the Watchdog timer. No matter which interval is chosen, the reset timeout is always occurred 512 WDT clock cycles later than the interrupt timeout.
<table>
<tr><th>WTIS</th><th>Interrupt Timeout</th><th>Reset Timeout</th><th>Real Time Interval (CLK=15MHz/256)</th></tr>
<tr><td>00</td><td>2^{14} clocks</td><td>2^{14} + 1024 clocks</td><td>0.28 sec.</td></tr>
<tr><td>01</td><td>2^{16} clocks</td><td>2^{16} + 1024 clocks</td><td>1.12 sec.</td></tr>
<tr><td>10</td><td>2^{18} clocks</td><td>2^{18} + 1024 clocks</td><td>4.47 sec.</td></tr>
<tr><td>11</td><td>2^{20} clocks</td><td>2^{20} + 1024 clocks</td><td>17.9 sec.</td></tr>
</table></td></tr>
<tr><td>[3]</td><td>WTIF</td><td>Watchdog Timer Interrupt Flag
If the Watchdog timer interrupt is enabled, then the hardware will set this bit to indicate that the Watchdog timer interrupt has occurred. If the Watchdog timer interrupt is not enabled, then this bit indicates that a timeout period has elapsed.
0 = Watchdog timer interrupt does not occur
1 = Watchdog timer interrupt occurs
NOTE: This bit is read only, but can be cleared by writing 1 to this bit.</td></tr>
<tr><td>[2]</td><td>WTRF</td><td>Watchdog Timer Reset Flag
When the Watchdog timer initiates a reset, the hardware will set this bit. This flag can be read by software to determine the source of reset. Software is responsible to clear it up manually. If WTRE is disabled, then the Watchdog timer has no effect on this bit.
0 = Watchdog timer reset does not occur
1 = Watchdog timer reset occurs
NOTE: This bit is read only, but can be cleared by writing 1 to this bit.</td></tr>
</table>

Continued

BITS	DESCRIPTIONS	
[1]	**WTRE**	**Watchdog Timer Reset Enable** Setting this bit will enable the Watchdog timer reset function. 0 = Disable Watchdog timer reset function 1 = Enable Watchdog timer reset function
[0]	**WTR**	**Watchdog Timer Reset** This bit brings the Watchdog timer into a known state. It helps reset the Watchdog timer before a timeout situation occurring. Failing to set **WTR** before timeout will initiates an interrupt if **WTIE** is set. If the **WTRE** bit is set, Watchdog timer reset will be occurred 512 WDT clock cycles after timeout. This bit is self-clearing. 0 = No operation 1 = Reset the contents of the Watchdog timer

6.12 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 ARM7TDMI processor provides two modes of interrupt, the **Fast Interrupt (FIQ)** mode for critical session and the *Interrupt* **(IRQ)** mode for general purpose. The IRQ exception is occurred when the nIRQ input is asserted. Similarly, the FIQ exception 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 W90N745 incorporates the **advanced interrupt controller (AIC)** that is capable of dealing with the interrupt requests from a total of 32 different sources. Currently, 31 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 differentiates the available 31 interrupt sources into eight priority levels. Interrupt sources within the priority level 0 have the highest priority and the priority level 7 has the lowest. To work this scheme properly, you must specify a certain priority level 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 within the priority level 0 is promoted to the FIQ. Interrupt sources within the priority levels other than 0 can petition for 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 W90N745 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. When the W90N745 is put in the test mode, all interrupt sources must be configured as positive-edge triggered.

The advanced interrupt controller includes the following features:

- AMBA APB bus interface
- External interrupts can be programmed as either edge-triggered or level-sensitive
- External interrupts can be programmed as either low-active or high-active
- Has flags to reflect the status of each interrupt source
- Individual mask for each interrupt source
- Proprietary 8-level interrupt scheme to ease the burden from the interrupt
- 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

6.12.1 Interrupt Sources

Table 6.12.1 W90N745 Interrupt Sources

PRIORITY	NAME	MODE	SOURCE
1 (Highest)	WDT_INT	Positive Level	Watch Dog Timer Interrupt
2	nIRQ0	Programmable	External Interrupt 0
3	nIRQ1	Programmable	External Interrupt 1
4	Reserved	-	-
5	Reserved	-	-
6	AC97_INT	Positive Level	AC97 Interrupt
7	Reserved	-	-
8	Reserved	-	-
9	UART_INT0	Positive Level	UART Interrupt0
10	UART_INT1	Positive Level	UART Interrupt1
11	UART_INT2	Positive Level	UART Interrupt2
12	UART_INT3	Positive Level	UART Interrupt3
13	T_INT0	Positive Level	Timer Interrupt 0
14	T_INT1	Positive Level	Timer Interrupt 1
15	USBH_INT0	Positive Level	USB Host Interrupt 0
16	USBH_INT1	Positive Level	USB Host Interrupt 1
17	EMCTX_INT	Positive Level	EMC TX Interrupt
18	EMCRX_INT	Positive Level	EMC RX Interrupt
19	GDMA_INT0	Positive Level	GDMA Channel Interrupt 0
20	GDMA_INT1	Positive Level	GDMA Channel Interrupt 1
21	Reserved	-	-
22	USBD_INT	Positive Level	USB Device Interrupt
23	Reserved	-	-
24	Reserved	-	-
25	I^2C_INT0	Positive Level	I^2C Interrupt0
26	I^2C_INT1	Positive Level	I^2C Interrupt1
27	SSP_INT	Positive Level	SSP Interrupt
28	PWM _INT	Positive Level	PWM Timer interrupt
29	KPI_INT	Positive Level	Keypad Interrupt
30	PS2_INT	Positive Level	PS2 Interrupt
31	nIRQ2/3_INT	Positive Level	GPIO0 & GPIO30 Interrupt

AIC Functional Description

Hardware Interrupt Vectoring

The hardware interrupt vectoring can be used to shorten the interrupt latency. If not used, priority determination must be carried out by software. When the Interrupt Priority Encoding Register (AIC_IPER) is read, it will return an integer representing the channel that is active and having the highest priority. This integer is equivalent to multiplied by 4 (shifted left two bits to word-align it) such that it may be used directly to index into a branch table to select the appropriate interrupt service routine vector.

Priority Controller

An 8-level priority encoder controls the nIRQ line. Each interrupt source belongs to priority group between of 0 to 7. Group 0 has the highest priority and group 7 the lowest. When more than one unmasked interrupt channels are active at a time, the interrupt with the highest priority is serviced first. If all active interrupts have equal priority, the interrupt with the lowest interrupt source number is serviced first.

The current priority level is defined as the priority level of the interrupt with the highest priority at the time the register AIC_IPER is read. In the case when a higher priority unmasked interrupt occurs while an interrupt already exits, there are two possible outcomes depending on whether the AIC_IPER has been read.

If the processor has already read the AIC_IPER and caused the nIRQ line to be de-asserted, then the nIRQ line is reasserted. When the processor has enabled nested interrupts and reads the AIC_IPER again, it reads the new, higher priority interrupt vector. At the same time, the current priority level is updated to the higher priority.

If the AIC_IPER has not been read after the nIRQ line has been asserted, then the processor will read the new higher priority interrupt vector in the AIC_IPER register and the current priority level is updated.

When the End of Service Command Register (AIC_EOSCR) is written, the current interrupt level is updated with the last stored interrupt level from the stack (if any). Therefore, at the end of a higher priority interrupt, the AIC returns to the previous state corresponding to the preceding lower priority interrupt which had been interrupted.

Interrupt Handling

When the IRQ line is asserted, the interrupt handler must read the AIC_IPER as soon as possible. This can de-assert the nIRQ request to the processor and clears the interrupt if it is programmed to be edge triggered. This allows the AIC to assert the nIRQ line again when a higher priority unmasked interrupt occurs.

The AIC_EOSCR (End of Service Command Register) must be written at the end of the interrupt service routine. This permits pending interrupts to be serviced.

Interrupt Masking

Each interrupt source, including FIQ, can be enabled or disabled individually by using the command registers AIC_MECR and AIC_MDCR. The status of interrupt mask can be read in the read only register AIC_IMR. A disabled interrupt doesn't affect the servicing of other interrupts.

Interrupt Clearing and Setting

All interrupt sources (including FIQ) can be individually set or clear by respectively writing to the registers AIC_SSCR and AIC_SCCR when they are programmed to be edge triggered. This feature of the AIC is useful in auto-testing or software debugging.

Fake Interrupt

When the AIC asserts the nIRQ line, the processor enters interrupt mode and the interrupt handler reads the AIC_IPER, it may happen that AIC de-asserts the nIRQ line after the processor has taken into account the nIRQ assertion and before the read of the AIC_IPER.

This behavior is called a fake interrupt.

The AIC is able to detect these fake interrupts and returns all zero when AIC_IPER is read. The same mechanism of fake interrupt occurs if the processor reads the AIC_IPER (application software or ICE) when there is no interrupt pending. The current priority level is not updated in this situation. Hence, the AIC_EOSCR shouldn't be written.

ICE/Debug Mode

This mode allows reading of the AIC_IPER without performing the associated automatic operations. This is necessary when working with a debug system. When an ICE or debug monitor reads the AIC user interface, the AIC_IPER can be read. This has the following consequences in normal mode:

- If there is no enabled pending interrupt, the fake vector will be returned.
- If an enabled interrupt with a higher priority than the current one is pending, it will be stacked.

In the second case, an End-of-Service command would be necessary to restore the state of the AIC. This operation is generally not performed by the debug system. Therefore, the debug system would become strongly intrusive, and could cause the application to enter an undesired state.

This can be avoided by using ICE/Debug Mode. When this mode is enabled. The AIC performs interrupt stacking only when a write access is performed on the AIC_IPER. Hence, the interrupt service routine must write to the AIC_IPER (any value) just after reading it. When AIC_IPER is written, the new status of AIC, including the value of interrupt source number register (AIC_ISNR), is updated with the value that is kept at previous reading of AIC_IPER The debug system must not write to the AIC_IPER as this would cause undesirable effects.

The following table shows the main steps of an interrupt and the order in which they are performed according to the mode:

ACTION	NORMAL MODE	ICE/DEBUG MODE
Calculate active interrupt	Read AIC_IPER	Read AIC_IPER
Determine and return the vector of the active interrupt	Read AIC_IPER	Read AIC_IPER
Push on internal stack the current priority level	Read AIC_IPER	Write AIC_IPER
Acknowledge the interrupt (Note 1)	Read AIC_IPER	Write AIC_IPER
No effect (Note 2)	Read AIC_IPER	

Notes:

- nIRQ de-assertion and automatic interrupt clearing if the source is programmed as level sensitive.
- Note that software which has been written and debugged using this mode will run correctly in normal mode without modification. However, in normal mode writing to AIC_IPER has no effect and can be removed to optimize the code

6.12.2 AIC Registers Map

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
AIC_SCR1	0xFFF8_2004	R/W	Source Control Register 1	0x0000_0047
AIC_SCR2	0xFFF8_2008	R/W	Source Control Register 2	0x0000_0047
AIC_SCR3	0xFFF8_200C	R/W	Source Control Register 3	0x0000_0047
AIC_SCR4	0xFFF8_2010	R/W	Source Control Register 4	0x0000_0047
AIC_SCR5	0xFFF8_2014	R/W	Source Control Register 5	0x0000_0047
AIC_SCR6	0xFFF8_2018	R/W	Source Control Register 6	0x0000_0047
AIC_SCR7	0xFFF8_201C	R/W	Source Control Register 7	0x0000_0047
AIC_SCR8	0xFFF8_2020	R/W	Source Control Register 8	0x0000_0047
AIC_SCR9	0xFFF8_2024	R/W	Source Control Register 9	0x0000_0047
AIC_SCR10	0xFFF8_2028	R/W	Source Control Register 10	0x0000_0047
AIC_SCR11	0xFFF8_202C	R/W	Source Control Register 11	0x0000_0047
AIC_SCR12	0xFFF8_2030	R/W	Source Control Register 12	0x0000_0047
AIC_SCR13	0xFFF8_2034	R/W	Source Control Register 13	0x0000_0047
AIC_SCR14	0xFFF8_2038	R/W	Source Control Register 14	0x0000_0047
AIC_SCR15	0xFFF8_203C	R/W	Source Control Register 15	0x0000_0047

AIC Registers Map, continued

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
AIC_SCR16	0xFFF8_2040	R/W	Source Control Register 16	0x0000_0047
AIC_SCR17	0xFFF8_2044	R/W	Source Control Register 17	0x0000_0047
AIC_SCR18	0xFFF8_2048	R/W	Source Control Register 18	0x0000_0047
AIC_SCR19	0xFFF8_204C	R/W	Source Control Register 19	0x0000_0047
AIC_SCR20	0xFFF8_2050	R/W	Source Control Register 20	0x0000_0047
AIC_SCR21	0xFFF8_2054	R/W	Source Control Register 21	0x0000_0047
AIC_SCR22	0xFFF8_2058	R/W	Source Control Register 22	0x0000_0047
AIC_SCR23	0xFFF8_205C	R/W	Source Control Register 23	0x0000_0047
AIC_SCR24	0xFFF8_2060	R/W	Source Control Register 24	0x0000_0047
AIC_SCR25	0xFFF8_2064	R/W	Source Control Register 25	0x0000_0047
AIC_SCR26	0xFFF8_2068	R/W	Source Control Register 26	0x0000_0047
AIC_SCR27	0xFFF8_206C	R/W	Source Control Register 27	0x0000_0047
AIC_SCR28	0xFFF8_2070	R/W	Source Control Register 28	0x0000_0047
AIC_SCR29	0xFFF8_2074	R/W	Source Control Register 29	0x0000_0047
AIC_SCR30	0xFFF8_2078	R/W	Source Control Register 30	0x0000_0047
AIC_SCR31	0xFFF8_207C	R/W	Source Control Register 31	0x0000_0047
AIC_IRSR	0xFFF8_2100	R	Interrupt Raw Status Register	0x0000_0000
AIC_IASR	0xFFF8_2104	R	Interrupt Active Status Register	0x0000_0000
AIC_ISR	0xFFF8_2108	R	Interrupt Status Register	0x0000_0000
AIC_IPER	0xFFF8_210C	R	Interrupt Priority Encoding Register	0x0000_0000
AIC_ISNR	0xFFF8_2110	R	Interrupt Source Number Register	0x0000_0000
AIC_IMR	0xFFF8_2114	R	Interrupt Mask Register	0x0000_0000
AIC_OISR	0xFFF8_2118	R	Output Interrupt Status Register	0x0000_0000
AIC_MECR	0xFFF8_2120	W	Mask Enable Command Register	Undefined
AIC_MDCR	0xFFF8_2124	W	Mask Disable Command Register	Undefined
AIC_SSCR	0xFFF8_2128	W	Source Set Command Register	Undefined
AIC_SCCR	0xFFF8_212C	W	Source Clear Command Register	Undefined
AIC_EOSCR	0xFFF8_2130	W	End of Service Command Register	Undefined
AIC_TEST	0xFFF8_2200	W	ICE/Debug mode Register	Undefined

AIC Source Control Registers (AIC_SCR1 ~ AIC_SCR31)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
AIC_SCR1	0xFFF8_2004	R/W	Source Control Register 1	0x0000_0047
AIC_SCR2	0xFFF8_2008	R/W	Source Control Register 2	0x0000_0047
• • •	• • •	• • •	• • •	• • •
AIC_SCR28	0xFFF8_2070	R/W	Source Control Register 28	0x0000_0047
AIC_SCR29	0xFFF8_2074	R/W	Source Control Register 29	0x0000_0047
AIC_SCR30	0xFFF8_2078	R/W	Source Control Register 30	0x0000_0047
AIC_SCR31	0xFFF8_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	
[31:8]	**Reserved**	**Reserved**
[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 other than nIRQ0, nIRQ1 should be configured as level sensitive during normal operation unless in the testing situation.

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

Continued

BITS	DESCRIPTIONS	
[5:3]	**Reserved**	**Reserved**
[2:0]	**PRIORITY**	**Priority Level** Every interrupt source must be assigned a priority level during initiation. 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 that located in the lower channel number has higher priority.

AIC Interrupt Raw Status Register (AIC_IRSR)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
AIC_IRSR	0xFFF8_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]	**IRSx**	This register records the intrinsic state within each interrupt channel. **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
[0]	**Reserved**	**Reserved**

AIC Interrupt Active Status Register (AIC_IASR)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
AIC_IASR	0xFFF8_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]	**IASx**	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. **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
[0]	**Reserved**	**Reserved**

AIC Interrupt Status Register (AIC_ISR)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
AIC_ISR	0xFFF8_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]	**ISx**	This register identifies those interrupt channels whose are both active and enabled. **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)
[0]	**Reserved**	**Reserved**

AIC IRQ Priority Encoding Register (AIC_IPER)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
AIC_IPER	0xFFF8_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
0	VECTOR					0	0

BITS	DESCRIPTIONS	
[6:2]	**Vector**	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. **VECTOR [6:2]: Interrupt Vector** 0 = no interrupt occurs 1 ~ 31 = representing the interrupt channel that is active, enabled, and having the highest priority
[0]	**Reserved**	**Reserved**

AIC Interrupt Source Number Register (AIC_ISNR)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
AIC_ISNR	0xFFF8_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	
[31:5]	**Reserved**	**Reserved**
[4:0]	**IRQID**	The purpose of this register is to record the interrupt channel number that is active, enabled, and has the highest priority. **IRQID [4:0]: IRQ Identification** Stands for the interrupt channel number

AIC Interrupt Mask Register (AIC_IMR)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
AIC_IMR	0xFFF8_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***	**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
[0]	**Reserved**	**Reserved**

AIC Output Interrupt Status Register (AIC_OISR)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
AIC_OISR	0xFFF8_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

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.

BITS	DESCRIPTIONS	
[31:2]	**Reserved**	**Reserved**
[1]	**IRQ**	**IRQ [1]: Interrupt Request** 0 = nIRQ line is inactive. 1 = nIRQ line is active.
[0]	**FIQ**	**FIQ [0]: 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	0xFFF8_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***	**MEC x: Mask Enable Command** 0 = No effect 1 = Enables the corresponding interrupt channel
[0]	**Reserved**	**Reserved**

AIC Mask Disable Command Register (AIC_MDCR)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
AIC_MDCR	0xFFF8_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]	MDCx	**MDC*x*: Mask Disable Command** 0 = No effect 1 = Disables the corresponding interrupt channel
[0]	Reserved	Reserved

AIC Source Set Command Register (AIC_SSCR)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
AIC_SSCR	0xFFF8_2128	W	Source Set Command Register	Undefined

31	30	29	28	27	26	25	24
SSC31	SSC30	SSC29	SSC28	SSC27	SSC26	SSC25	SSC24
23	22	21	20	19	18	17	16
SSC23	SSC22	SSC21	SSC20	SSC19	SSC18	SSC17	SSC16
15	14	13	12	11	10	9	8
SSC15	SSC14	SSC13	SSC12	SSC11	SSC10	SSC9	SSC8
7	6	5	4	3	2	1	0
SSC7	SSC6	SSC5	SSC4	SSC3	SSC2	SSC1	RESERVED

BITS	DESCRIPTIONS	
[31:1]	**SSCx**	When the W90N745 is under debugging or verification, software can activate any interrupt channel by setting the corresponding bit in this register. This feature is useful in hardware verification or software debugging. **SSCx: Source Set Command** 0 = No effect. 1 = Activates the corresponding interrupt channel
[0]	**Reserved**	**Reserved**

AIC Source Clear Command Register (AIC_SCCR)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
AIC_SCCR	0xFFF8_212C	W	Source Clear Command Register	Undefined

31	30	29	28	27	26	25	24
SCC31	**SCC30**	**SCC29**	**SCC28**	**SCC27**	**SCC26**	**SCC25**	**SCC24**
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
SCC23	**SCC22**	**SCC21**	**SCC20**	**SCC19**	**SCC18**	**SCC17**	**SCC16**
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
SCC15	**SCC14**	**SCC13**	**SCC12**	**SCC11**	**SCC10**	**SCC9**	**SCC8**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SCC7	**SCC6**	**SCC5**	**SCC4**	**SCC3**	**SCC2**	**SCC1**	RESERVED

BITS	DESCRIPTIONS	
[31:1]	**SCCx**	When the W90N745 is under debugging or verification, software can deactivate any interrupt channel by setting the corresponding bit in this register. This feature is useful in hardware verification or software debugging. **SCCx: Source Clear Command** 0 = No effect. 1 = Deactivates the corresponding interrupt channels
[0]	**Reserved**	**Reserved**

AIC End of Service Command Register (AIC_EOSCR)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
AIC_EOSCR	0xFFF8_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
---	---	---	---	---	---	---	---

BITS	DESCRIPTIONS	
[31:0]	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.

AIC ICE/Debug Register (AIC_TEST)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
AIC_TEST	0xFFF8_2200	W	ICE/Debug mode 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							
7	6	5	4	3	2	1	0
RESERVED							TEST

BITS	DESCRIPTIONS	
[31:1]	**Reserved**	**Reserved**
[0]	**TEST**	This register indicates whether AIC_IPER will be cleared or not after been read. If bit0 of AIC_TEST has been set, ICE or debug monitor can read AIC_IPER for verification and the AIC_IPER will not be cleared automatically. Write access to the AIC_IPER will perform the interrupt stacking in this mode. **TEST: ICE/Debug mode** 0 = normal mode. 1 = ICE/Debug mode.

6.13 General-Purpose Input/Output

The General-Purpose Input/Output (**GPIO**) module possesses 31 pins and serves multiple function purposes. Each port can be configured by software to meet various system configurations and design requirements. Software must configure each pin before starting the main program. If a pin is not used for multiplexed functions, the pin can be configured as I/O port

Two extended interrupts nIRQ2 (GPIO0 pin) and nIRQ3 (nWAIT pin) are used the same interrupt request (channel #31) of AIC. It can be programmed as low/high sensitive or positive/negative edge triggered. When interrupt #31 assert in AIC, software can poll **XISTATUS** status register to identify which interrupt occur.

These 31 IO pins are divided into 7 groups according to its peripheral interface definition.

- Port0: 5-pin input/output port
- Port1: 2-pin input/output port
- Port2: 10-pin input/output port
- Port3: Reserved
- Port4: 1-pin input/output port
- Port5: 13-pin input/output port need updated
- Port6: Reserved

Table 6.13.1 GPIO multiplexed functions table

PORT0	**Configurable Pin Functions**			
0	GPIO0	AC97_nRESET (I²S_MCLK)	nIRQ2	USBPWREN
1	GPIO1	AC97_DATAI (I²S_DATAI)	PWM0	DTR3
2	GPIO2	AC97_DATAO (I²S_DATAO)	PWM1	DSR3
3	GPIO3	AC97_SYNC (I²S_LRCLK)	PWM2	TXD3
4	GPIO4	AC97_BITCLK (I²S_BITCLK)	PWM3	RXD3
PORT1	**Configuration Pin Functions**			
0	GPIO18	-	nXDACK	-
1	GPIO19	-	nXDREQ	-
PORT2	**Configuration Pin Functions**			
0	GPIO20	PHY_RXERR	KPCOL0	-
1	GPIO21	PHY_CRSDV	KPCOL1	-
2	GPIO22	PHY_RXD[0]	KPCOL2	-
3	GPIO23	PHY_RXD[1]	KPCOL3	-

Table 6.13.1 GPIO multiplexed functions table, continued

4	GPIO24	PHY_REFCLK	KPCOL4	-
5	GPIO25	PHY_TXEN	KPCOL5	-
6	GPIO26	PHY_TXD[0]	KPCOL6	-
7	GPIO27	PHY_TXD[1]	KPCOL7	-
8	GPIO28	PHY_MDIO	KPROW0	-
9	GPIO29	PHY_MDC	KPROW1	-
PORT3	**Configuration Pin Functions**			
RESERVED				
PORT4	**Configuration Pin Functions**			
0	GPIO30	nWAIT	nIRQ3	-
PORT5	**Configuration Pin Functions**			
0	GPIO5	TXD0	-	-
1	GPIO6	RXD0	-	-
2	GPIO7	TXD1	-	-
3	GPIO8	RXD1	-	-
4	GPIO9	TXD2	CTS1	PS2CLK
5	GPIO10	RXD2	RTS1	PS2DATA
6	GPIO11	SCL0	SFRM	TIMER0
7	GPIO12	SDA0	SSPTXD	TIMER1
8	GPIO13	SCL1	SCLK	KPROW3
9	GPIO14	SDA1	SSPRXD	KPROW2
10	GPIO15	nWDOG	USBPWREN	-
11	GPIO16	nIRQ0	-	-
12	GP1O17	nIRQ1	USBOVRCUR	-
PORT6	**Configuration Pin Function**			
RESERVED				

6.13.1 GPIO Register Description

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
GPIO_CFG0	0xFFF8_3000	R/W	GPIO Port0 Configuration Register	0x0000_0000
GPIO_DIR0	0xFFF8_3004	R/W	GPIO Port0 Direction Control Register	0x0000_0000
GPIO_DATAOUT0	0xFFF8_3008	R/W	GPIO Port0 Data Output Register	0x0000_0000
GPIO_DATAIN0	0xFFF8_300C	R	GPIO Port0 Data Input Register	0xXXXX_XXXX
GPIO_CFG1	0xFFF8_3010	R/W	GPIO port1 configuration register	0x0000_0000
GPIO_DIR1	0xFFF8_3014	R/W	GPIO port1 direction control register	0x0000_0000
GPIO_DATAOUT1	0xFFF8_3018	R/W	GPIO port1 data output register	0x0000_0000
GPIO_DATAIN1	0xFFF8_301C	R	GPIO port1 data input register	0xXXXX_XXXX
GPIO_CFG2	0xFFF8_3020	R/W	GPIO Port2 Configuration Register	0x0000_0000
GPIO_DIR2	0xFFF8_3024	R/W	GPIO Port2 Direction Control Register	0x0000_0000
GPIO_DATAOUT2	0xFFF8_3028	R/W	GPIO Port2 Data Output Register	0x0000_0000
GPIO_DATAIN2	0xFFF8_302C	R	GPIO Port2 Data Input Register	0xXXXX_XXXX
GPIO_CFG4	0xFFF8_3040	R/W	GPIO Port4 Configuration Register	0x0015_5555
GPIO_DIR4	0xFFF8_3044	R/W	GPIO Port4 Direction Control Register	0x0000_0000
GPIO_DATAOUT4	0xFFF8_3048	R/W	GPIO Port4 Data Output Register	0x0000_0000
GPIO_DATAIN4	0xFFF8_304C	R	GPIO Port4 Data Input Register	0xXXXX_XXXX
GPIO_CFG5	0xFFF8_3050	R/W	GPIO Port5 Configuration Register	0x0000_0000
GPIO_DIR5	0xFFF8_3054	R/W	GPIO Port5 Direction Control Register	0x0000_0000
GPIO_DATAOUT5	0xFFF8_3058	R/W	GPIO Port5 Data Output Register	0x0000_0000

GPIO Register Description, continued.

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
GPIO_DATAIN5	0xFFF8_305C	R	GPIO Port5 Data Input Register	0xXXXX_XXXX
GPIO_DBNCECON	0xFFF8_3070	R/W	GPIO Input Debounce Control Register	0x0000_0000
GPIO_XICFG	0xFFF8_3074	R/W	Extend Interrupt Configure Register	0xXXXX_XXX0
GPIO_XISTATUS	0xFFF8_3078	R/W	Extend Interrupt Status Register	0xXXXX_XXX0

6.13.2 GPIO Register Description

GPIO Port0 Configuration Register (GPIO_CFG0)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
GPIO_CFG0	0xFFF8_3000	R/W	GPIO port0 configuration 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						PT0CFG4	
7	6	5	4	3	2	1	0
PT0CFG3		PT0CFG2		PT0CFG1		PT0CFG0	

PT0CFG0	11		10		01		00	
	NAME	TYPE	NAME	TYPE	NAME	TYPE	NAME	TYPE
PORT0_0	USB_PWREN	O	nIRQ2	I	AC97RESET or I²SMCLK	O	GPIO0	I/O

PT0CFG1	11		10		01		00	
	NAME	TYPE	NAME	TYPE	NAME	TYPE	NAME	TYPE
PORT0_1	DTR3	O	PWM0	O	AC97DATAI or I²SDATAI	O	GPIO1	I/O

Continued

PT0CFG2	11		10		01		00	
	NAME	TYPE	NAME	TYPE	NAME	TYPE	NAME	TYPE
PORT0_2	DSR3	I	PWM1	O	AC97DATAO or I²SDATAO	O	GPIO2	I/O

PT0CFG3	11		10		01		00	
	NAME	TYPE	NAME	TYPE	NAME	TYPE	NAME	TYPE
PORT0_3	TXD3	O	PWM2	O	AC97SYNC or I²SLRCLK	O	GPIO3	I/O

PT0CFG4	11		10		01		00	
	NAME	TYPE	NAME	TYPE	NAME	TYPE	NAME	TYPE
PORT0_4	RXD3	O	PWM3	O	AC97BITCLK or I²SBITCLK	I O	GPIO4	I/O

GPIO Port0 Direction Register (GPIO_DIR0)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
GPIO_DIR0	0xFFF8_3004	R/W	GPIO port0 in/out direction control and pull-up enable register	0x0000_0000

31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED				PUPEN0[3:0]			
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
RESERVED			OMDEN0[4:0]				

BITS	DESCRIPTION	
[31:20]	RESERVED	-
[19:16]	**PUPEN0**	**GPIO3 -GPIO0 port pin internal pull-up resister enable** There are 4 bits for this register, the corresponding bit is set to "1" will enable pull-up resister on IO pin. 1 = enable 0 = disable After power on the pull-up resisters are disabled. **NOTE:** GPIO4 is used as AC97 BITCLK input, an IO pad with Schmitt trigger input buffer PDB04SDGZ is implemented for this pin. Due to TSMC IO library without pull-up register, an external pull-up resister is necessary.
[15:5]	RESERVED	
[4:0]	**OMDEN0**	**GPIO4 ~GPIO0 output mode enable** 1 = output mode 0 = input mode **NOTE:** Output mode enable bits are valid only when bit PT0CFG4-0 is configured as general purpose I/O mode. Each port pin can be enabled individually by setting the corresponding control bit.

GPIO Port0 Data Output Register (GPIO_DATAOUT0)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
GPIO_DATAOUT0	0xFFF8_3008	R/W	GPIO port0 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			DATAOUT0				

BITS	DESCRIPTION	
[31:5]	RESERVED	-
[4:0]	**DATAOUT0**	**PORT0 data output value** Writing data to this register will reflect the data value on the corresponding port0 pin when it is configured as general purpose output pin. And writing data to reserved bits is not effective.

GPIO Port0 Data Input Register (GPIO_DATAIN0)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
GPIO_DATAIN0	0xFFF8_300C	R/W	GPIO port0 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			DATAIN0				

BITS	DESCRIPTION	
[31:5]	RESERVED	-
[4:0]	**DATAIN0**	**PORT0 data input value** The DATAIN0 indicates the status of each GPIO0~GPIO4 port pin regardless of its operation mode. The reserved bits will be read as "0".

GPIO Port1 Configuration Register (GPIO_CFG1)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
GPIO_CFG1	0xFFF8_3010	R/W	GPIO port1 configuration 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				PT1CFG1		PT1CFG0	

*In the following pin definition, mark with shading is default function.

PT1CFG0	11		10		01		00	
	NAME	TYPE	NAME	TYPE	NAME	TYPE	NAME	TYPE
PORT1_0	-		-	-	nXDACK	O	GPIO18	I/O

PT1CFG1	11		10		01		00	
	NAME	TYPE	NAME	TYPE	NAME	TYPE	NAME	TYPE
PORT1_1	-		-	-	nXDREQ	I	GPIO19	I/O

GPIO Port1 Direction Register (GPIO_DIR1)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
GPIO_DIR1	0xFFF8_3014	R/W	GPIO port0 in/out direction control and pull-up enable register	0x0000_0000

31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED						PUPEN1[1:0]	
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
RESERVED						OMDEN1[1:0]	

BITS	DESCRIPTION	
[31:18]	RESERVED	-
[17:16]	**PUPEN1**	**GPIO19 ~ GPIO18 port pins internal pull-up resister enable** This is a 2-bit registers, set corresponding bit to “1” will enable pull up resister in IO pin. 1 = enable 0 = disable After power on the resisters are disabled.
[15:2]	RESERVED	-

Continued.

[1:0]	**OMDEN1**	**GPIO19 ~ GPIO18 output mode enable** 1 = enable 0 = disable **NOTE:** Output mode enable bits are valid only when bit PT1CFG1-0 is configured as general purpose I/O mode. Each port pin can be enabled individually by setting the corresponding control bit.

GPIO Port1 Data Output Register (GPIO_DATAOUT1)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
GPIO_DATAOUT1	0xFFF8_3018	R/W	GPIO port1 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						DATAOUT1[1:0]	

BITS	DESCRIPTION	
[31:2]	RESERVED	-
[1:0]	**DATAOUT1**	**PORT1 data output value** Writing data to this register will reflect the data value on the corresponding port1 pin when it is configured as general purpose output pin. And writing data to reserved bits is not effective.

GPIO Port1 Data Input Register (GPIO_DATAIN1)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
GPIO_DATAIN1	0xFFF8_301C	R/W	GPIO port1 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						DATAIN1[1:0]	

BITS	DESCRIPTION	
[31:2]	RESERVED	-
[1:0]	**DATAIN1**	**Port1 input data register** The DATAIN1 indicates the status of each GPIO19~GPIO18 pin regardless of its operation mode. The reserved bits are read as 0s.

GPIO Port2 Configuration Register (GPIO_CFG2)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
GPIO_CFG2	0xFFF8_3020	R/W	GPIO port2 configuration register	0x0000_0000

31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED				PT2CFG9		PT2CFG8	
15	14	13	12	11	10	9	8
PT2CFG7		PT2CFG6		PT2CFG5		PT2CFG4	
7	6	5	4	3	2	1	0
PT2CFG3		PT2CFG2		PT2CFG1		PT2CFG0	

*In the following pin definition, mark with shading is default function.

PT2CFG0	11		10		01		00	
	Name	Type	Name	Type	Name	Type	Name	Type
PORT2_0	RESERVED		KPCOL0	I	PHY_RXERR	I	GPIO20	I/O

PT2CFG1	11		10		01		00	
	Name	Type	Name	Type	Name	Type	Name	Type
PORT2_1	RESERVED		KPCOL1	I	PHY_CRSDV	I	GPIO21	I/O

PT2CFG2	11		10		01		00	
	Name	Type	Name	Type	Name	Type	Name	Type
PORT2_2	RESERVED		KPCOL2	I	PHY_RXD[0]	I	GPIO22	I/O

PT2CFG3	11		10		01		00	
	Name	Type	Name	Type	Name	Type	Name	Type
PORT2_3	RESERVED		KPCOL3	I	PHY_RXD[1]	I	GPIO23	I/O

PT2CFG4	11		10		01		00	
	Name	Type	Name	Type	Name	Type	Name	Type
PORT2_4	RESERVED		KPCOL4	I	PHY_REFCLK	I	GPIO24	I/O

PT2CFG5	11		10		01		00	
	Name	Type	Name	Type	Name	Type	Name	Type
PORT2_5	RESERVED		KPCOL5	I	PHY_TXEN	O	GPIO25	I/O

PT2CFG6	11		10		01		00	
	Name	Type	Name	Type	Name	Type	Name	Type
PORT2_6	RESERVED		KPCOL6	I	PHY_TXD[0]	O	GPIO26	I/O

PT2CFG7	11		10		01		00	
	Name	Type	Name	Type	Name	Type	Name	Type
PORT2_7	RESERVED		KPCOL7	I	PHY_TXD[1]	O	GPIO27	I/O

PT2CFG8	11		10		01		00	
	Name	Type	Name	Type	Name	Type	Name	Type
PORT2_8	RESERVED		KPROW0	O	PHY_MDIO	I/O	GPIO28	I/O

PT2CFG9	11		10		01		00	
	Name	Type	Name	Type	Name	Type	Name	Type
PORT2_9	RESERVED		KPROW1	O	PHY_MDC	O	GPIO29	I/O

GPIO Port2 Direction Register (GPIO_DIR2)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
GPIO_DIR2	0xFFF8_3024	R/W	GPIO port2 in/out direction control and pull-up enable register	0x0000_0000

31	30	29	28	27	26	25	24
RESERVED						PUPEN2[9:8]	
23	22	21	20	19	18	17	16
PUPEN2[7:0]							
15	14	13	12	11	10	9	8
RESERVED						OMDEN2[9:8]	
7	6	5	4	3	2	1	0
OMDEN2[7:0]							

BITS	DESCRIPTION	
[31:26]	RESERVED	-
[25:16]	**PUPEN2**	**GPIO29 ~ GPIO20 port pin internal pull-up resister enable** There are 10 bits for this register, the corresponding bit is set to "1" will enable pull-up resister on IO pin. 1 = enable 0 = disable After power on, the registers are disabled.
[15:10]	RESERVED	-
[9:0]	**OMDEN2**	**GPIO19 ~ GPIO20 output mode enable** 1 = output mode 0 = input mode **NOTE:** Output mode enable bits are valid only when bit PT2CFG9-0 is configured as general purpose I/O mode. Each port pin can be enabled individually by setting the corresponding control bit.

PGPIO Port2 Data Output Register (GPIO_DATAOUT2)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
GPIO_DATAOUT2	0xFFF8_3028	R/W	GPIO port2 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						DATAOUT2[9:8]	
7	6	5	4	3	2	1	0
DATAOUT2[7:0]							

BITS	DESCRIPTION	
[31:10]	RESERVED	-
[9:0]	**DATAOUT2**	**PORT2 data output value** Writing data to this register will reflect the data value on the corresponding port2 pin when it is configured as general purpose output pin. And writing data to reserved bits is not effective.

GPIO Port2 Data Input Register (GPIO_DATAIN2)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
GPIO_DATAIN2	0xFFF8_302C	R/W	GPIO port2 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						DATAIN2[9:8]	
7	6	5	4	3	2	1	0
DATAIN2[7:0]							

BITS	DESCRIPTION	
[31:10]	RESERVED	-
[9:0]	**DATAIN2**	**Port2 input data register** The DATAIN2 indicates the status of each GPIO18~GPIO27 pin regardless of its operation mode. The reserved bits will be read as 0s.

GPIO Port4 Configuration Register (GPIO_CFG4)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
GPIO_CFG4	0xFFF8_3040	R/W	GPIO port4 configuration register	0x0015_5555

31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED		PT4CFG10		RESERVED			
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
RESERVED							

*In the following pin definition, mark with shading is default function.

PT4CFG10	11		10		01		00	
	NAME	TYPE	NAME	TYPE	NAME	TYPE	NAME	TYPE
PORT4_10	RESERVED		nIRQ3	I	nWAIT	I	GPIO28	I/O

GPIO Port4 Direction Register (GPIO_DIR4)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
GPIO_DIR4	0xFFF8_3044	R/W	GPIO port4 in/out direction control and pull-up enable register	0x0000_0000

31	30	29	28	27	26	25	24
RESERVED					PUPEN4[10]	RESERVED	
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED					OMDEN4[10]	RESERVED	
7	6	5	4	3	2	1	0
RESERVED							

BITS	DESCRIPTION	
[31:27]	RESERVED	-
[26]	**PUPEN4**	**GPIO28 pin internal pull-up resister enable** There is 1 bit for this register, the bit is set to “1” will enable pull-up resister on IO pin. 1 = enable 0 = disable After power on the pull-up resister is disabled
[25:11]	RESERVED	-
[10]	**OMDEN4**	**GPIO28 output mode enable** 1 = enable 0 = disable **NOTE:** Output mode enable bits are valid only when bit PT4CFG10 is configured as general purpose I/O mode. Each port pin can be enabled individually by setting the corresponding control bit.
[9:0]	RESERVED	-

GPIO Port4 Data Output Register (GPIO_DATAOUT4)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
GPIO_DATAOUT4	0xFFF8_3048	R/W	GPIO port4 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					DATAOUT4[10]	RESERVED	
7	6	5	4	3	2	1	0
RESERVED							

BITS	DESCRIPTION	
[31:11]	RESERVED	-
[10]	**DATAOUT4**	**PORT4 data output value** Writing data to this register will reflect the data value on the corresponding port4 pin when it is configured as general purpose output pin. And writing data to reserved bits is not effective.
[9:0]	RESERVED	-

GPIO Port4 Data Input Register (GPIO_DATAIN4)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
GPIO_DATAIN4	0xFFF8_304C	R/W	GPIO port4 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					DATAIN4[10]	RESERVED	
7	6	5	4	3	2	1	0
RESERVED							

BITS	DESCRIPTION	
[31:11]	RESERVED	-
[10:0]	**DATAIN4**	**Port4 input data register** The DATAIN4 indicates the status of GPIO28 pin regardless of its operation mode. The reserved bits will be read as 0s
[9:0]	RESERVED	-

GPIO Port5 Configuration Register (GPIO_CFG5)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
GPIO_CFG5	0xFFF8_3050	R/W	GPIO port5 configuration register	0x0000_0000

31	30	29	28	27	26	25	24
RESERVED						PT5CFG12	
23	22	21	20	19	18	17	16
PT5CFG11		PT5CFG10		PT5CFG9		PT5CFG8	
15	14	13	12	11	10	9	8
PT5CFG7		PT5CFG6		PT5CFG5		PT5CFG4	
7	6	5	4	3	2	1	0
PT5CFG3		PT5CFG2		PT5CFG1		PT5CFG0	

*In the following pin definition, mark with shading is default function.

PT5CFG0	11		10		01		00	
	Name	Type	Name	Type	Name	Type	Name	Type
PORT5_0	RESERVED		RESERVED		TXD0	O	GPIO5	I/O

PT5CFG1	11		10		01		00	
	Name	Type	Name	Type	Name	Type	Name	Type
PORT5_1	RESERVED		RESERVED		RXD0	I	GPIO6	I/O

PT5CFG2	11		10		01		00	
	Name	Type	Name	Type	Name	Type	Name	Type
PORT5_2	RESERVED		RESERVED		TXD1	O	GPIO7	I/O

PT5CFG3	11		10		01		00	
	Name	Type	Name	Type	Name	Type	Name	Type
PORT5_3	RESERVED		RESERVED		RXD1	I	GPIO8	I/O

PT5CFG4	11		10		01		00	
	Name	Type	Name	Type	Name	Type	Name	Type
PORT5_4	PS2CLK	O	CTS1	I	TXD2	IO	GPIO9	I/O

Continued

PT5CFG5	11		10		01		00	
	Name	Type	Name	Type	Name	Type	Name	Type
PORT5_5	PS2DATA	I/O	RTS1	IO	RXD2	I	GPIO10	I/O

PT5CFG6	11		10		01		00	
	Name	Type	Name	Type	Name	Type	Name	Type
PORT5_6	TIMER0	O	SFRM	O	SCL0	I/O	GPIO11	I/O

PT5CFG7	11		10		01		00	
	Name	Type	Name	Type	Name	Type	Name	Type
PORT5_7	TIMER1	O	SSPTXD	O	SDA0	I/O	GPIO12	I/O

PT5CFG8	11		10		01		00	
	Name	Type	Name	Type	Name	Type	Name	Type
PORT5_8	KPROW2	O	SSPSCLK	O	SCL1	I/O	GPIO13	I/O

PT5CFG9	11		10		01		00	
	Name	Type	Name	Type	Name	Type	Name	Type
PORT5_9	KPROW3	O	SSPRXD	I/O	SDA1	I/O	GPIO14	I/O

PT5CFG10	11		10		01		00	
	Name	Type	Name	Type	Name	Type	Name	Type
PORT5_10	RESERVED		USBPWREN	O	nWDOG	O	GPIO15	I/O

PT5CFG11	11		10		01		00	
	Name	Type	Name	Type	Name	Type	Name	Type
PORT5_11	RESERVED		RESERVED		nIRQ0	I	GPIO16	I/O

PT5CFG12	11		10		01		00	
	Name	Type	Name	Type	Name	Type	Name	Type
PORT5_12	RESERVED		USBOVCUR	I	nIRQ1	I	GPIO17	I/O

GPIO Port5 Direction Register (GPIO_DIR5)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
GPIO_DIR5	0xFFF8_3054	R/W	GPIO port5 in/out direction control and pull-up enable register	0x0000_0000

31	30	29	28	27	26	25	24
RESERVED			PUPEN5[12:8]				
23	22	21	20	19	18	17	16
PUPEN5[7:0]							
15	14	13	12	11	10	9	8
RESERVED			OMDEN5[12:8]				
7	6	5	4	3	2	1	0
OMDEN5[7:0]							

BITS	DESCRIPTION	
[31:29]	RESERVED	-
[28:16]	**PUPEN5**	**GPIO17 ~ GPIO5 port pin internal pull-up resister enable** There are 13 bits for this register, the corresponding bit is set to "1" will enable pull-up resister on IO pin. 1 = enable 0 = disable After power on the pull-up resisters are disable.
[15:13]	RESERVED	
[12:0]	**OUTEN5**	**GPIO17 ~ GPIO5 output mode enable** 1 = output mode 0 = input mode **NOTE:** Output mode enable bits are valid only when bit PT5CFG12-0 is configured as general purpose I/O mode. Each port pin can be enabled individually by setting the corresponding control bit.

GPIO Port5 Data Output Register (GPIO_DATAOUT5)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
GPIO_DATAOUT5	0xFFF8_3058	R/W	GPIO port5 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			DATAOUT5[12:8]				
7	6	5	4	3	2	1	0
DATAOUT5[7:0]							

BITS	DESCRIPTION	
[31:13]	RESERVED	-
[12:0]	**DATAOUT5**	**PORT5 data output value** Writing data to this register will reflect the data value on the corresponding port5 pin when it is configured as general purpose output pin. And writing data to reserved bits is not effective.

GPIO Port5 Data Input Register (GPIO_DATAIN5)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
GPIO_DATAIN5	0xFFF8_305C	R/W	GPIO port5 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			DATAIN5[12:8]				
7	6	5	4	3	2	1	0
DATAIN5[7:0]							

BITS	DESCRIPTION	
[31:13]	RESERVED	-
[12:0]	**DATAIN5**	**Port5 input data register** The DATAIN5 indicates the status of each GPIO17~GPIO5 pin regardless of its operation mode. The reserved bits will be read as 0s.

GPIO Debounce Control Register (GPIO_DBNCECON)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
GPIO_DBNCECON	0xFFF8_3070	R/W	GPIO debounce control register	0xXXXX_XX00

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	DBCLKSEL			RESERVED		DBEN1	DBEN0

BITS	DESCRIPTION	
[31:7]	RESERVED	-
[6:4]	**DBCLKSEL**	**Debounce 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 / $2^{DBCLKSEL}$**
[3:2]	RESERVED	-
[1]	**DBEN1**	**Debounce circuit enable for GPIO17 (nIRQ1)** 1 = enable 0 = disable
[0]	**DBEN0**	**Debounce circuit enable for GPIO16 (nIRQ0)** 1 = enable 0 = disable

GPIO Interrupt Configuration Register (GPIO_XICFG)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
GPIO_XICFG	0xFFF8_3074	R/W	Extend interrupt configure register	0xXXXX_XX00

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
EnIRQ3	DBE3	ISTYPE3		EnIRQ2	DBE2	ISTYPE2	

BITS	DESCRIPTION	
[31:8]	RESERVED	-
[7]	**EnIRQ3**	**Enable nIRQ3** Setting this bit 1 to enable nIRQ3. 1 = Enable nIRQ3 0 = Disable nIRQ3 The AIC interrupt channel 31 is reserved for nIRQ3 and nIRQ2 (wired-OR), if this bit is set and nIRQ3 occur, then it will send an interrupt request signal into AIC module.
[6]	**DBE3**	**Debounce circuit enable for nIRQ3** (alternative function of nWAIT pin) The nIRQ3 shares the same debounce circuit with nIRQ[3:0], software can configure debounce sampling time in GPIO_DEBNCE control register. DBE3 function is the same as DBE0 in GPIO_DBENCE register. 1 = Enable debounce 0 = Disable debounce
[5:4]	**ISTYPE3**	**nIRQ3 source type** ISTYPE3 / Interrupt Source Type 2'b00 – LOW level sensitive 2'b01 – HIGH level sensitive 2'b10 – Negative edge triggered 2'b11 – Positive edge triggered

Continued

BITS	DESCRIPTION	
[3]	**EnIRQ2**	**Enable nIRQ2** Setting this bit 1 to enable nIRQ2 1 = Enable nIRQ2 0 = Disable nIRQ2 The AIC interrupt channel 31 is reserved for nIRQ3 and nIRQ2 (wire-OR), if this bit is set and nIRQ2 occur, then it will send an interrupt request signal into AIC module.
[2]	**DBE2**	**Debounce circuit enable for nIRQ2** (alternative function of GPIO0 pin) 1 = Enable debounce 0 = Disable debounce The nIRQ2 shares the same debounce circuit with nIRQ[1:0], software can configure debounce sampling time in GPIO_DEBNCE control register. DBE2 function is the same as DBE0 in GPIO_DBENCE register.
[1:0]	**ISTYPE2**	**nIRQ2 source type** **ISTYPE2** / **Interrupt Source Type** 2'b00 / LOW level sensitive 2'b01 / HIGH level sensitive 2'b10 / Negative edge triggered 2'b11 / Positive edge triggered

GPIO Interrupt Status Register (GPIO_XISTATUS)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
GPIO_XISTATUS	0xFFF8_3078	R/W	Extend interrupt status register	0xXXXX_XX00

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						nIRQ3	nIRQ2

BITS	DESCRIPTION	
[31:2]	RESERVED	-
[1]	**nIRQ3**	**Interrupt 3 status** When interrupt input is detected with ISTYPE3 triggered condition, this flag will be set. It must be cleared by software. 1 = interrupt nIRQ3 is detected. 0 = No interrupt
[0]	**nIRQ2**	**Interrupt 2 status** When interrupt input is detected with ISTYPE2 triggered condition, this flag will be set. It must be cleared by software. 1 = interrupt nIRQ2 is detected. 0 = no interrupt

6.14 I²C Interface

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 made up to 100 kbit/s in Standard-mode, up to 400 kbit/s in the Fast-mode, or up to 3.4 Mbit/s in the High-speed mode. Only 100kbps and 400kbps modes are supported directly. For High-speed mode special IOs are needed. If these IOs are available and used, then High-speed mode is also supported.

Data is transferred between a Master and a Slave synchronously to SCL on the SDA line on a **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:

- AMBA APB interface compatible
- Compatible with Philips 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

6.14.1 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

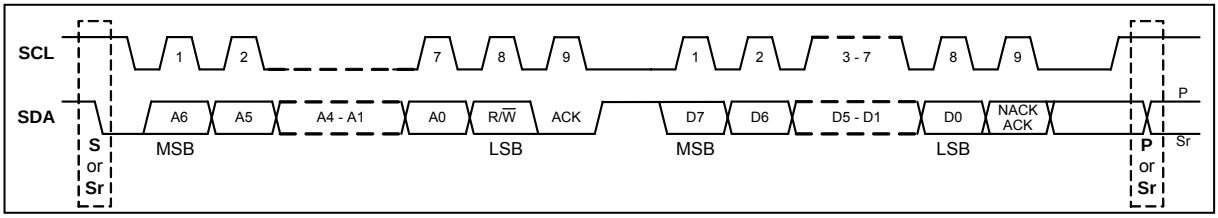
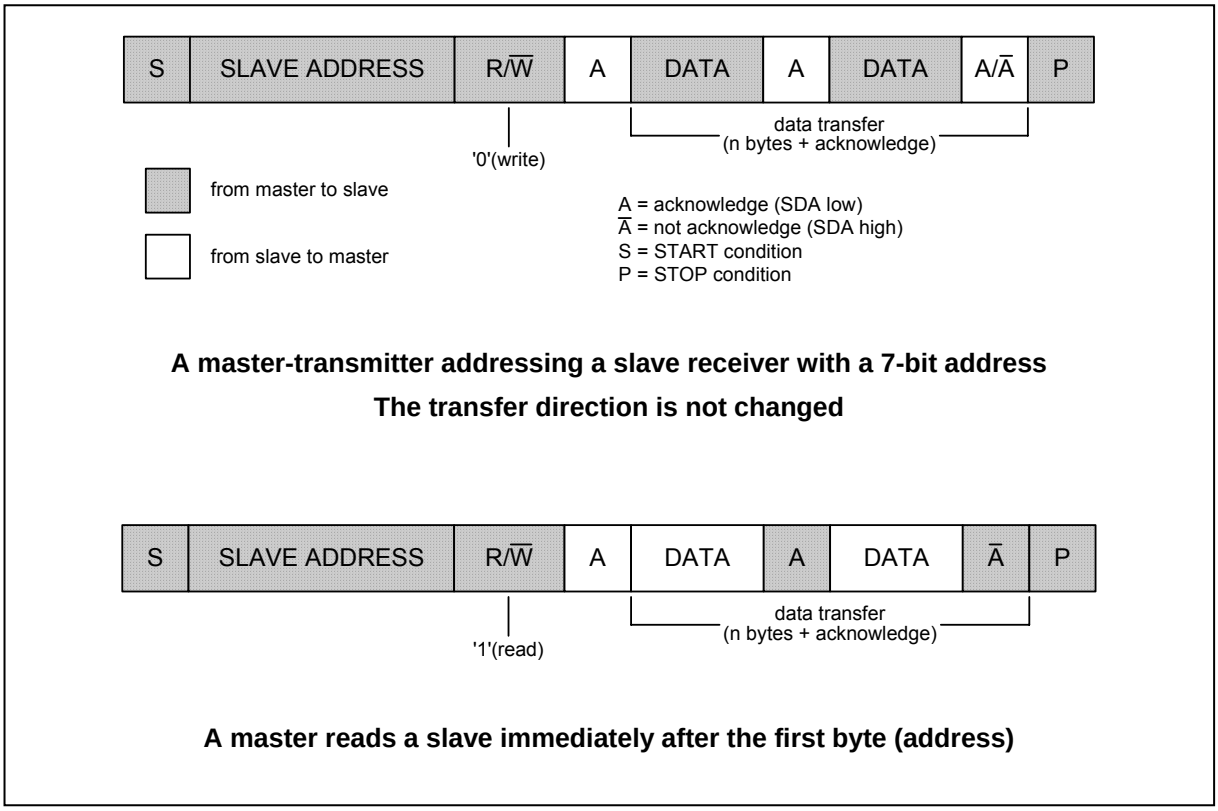


Figure 6.14.1 Data transfer on the I²C-bus



START or Repeated START signal

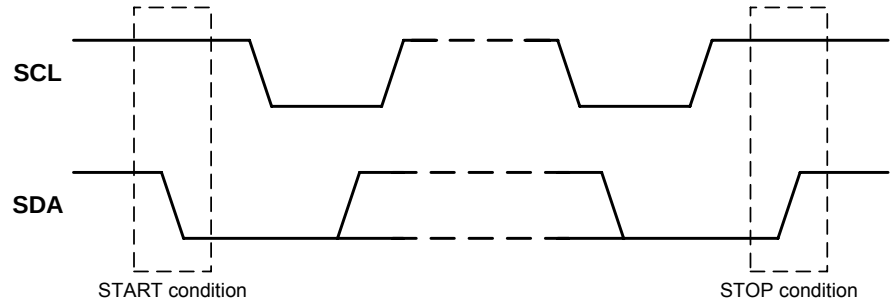
When the bus is free/idle, meaning no master device is engaging the bus (both SCL and SDA lines are high), a master can initiate a transfer by sending a START signal. A START signal, usually referred to as the **S-bit**, is defined as a **HIGH to LOW** transition on the SDA line while SCL is **HIGH**. 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**.

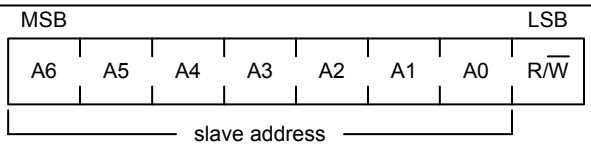


START and STOP conditions

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.



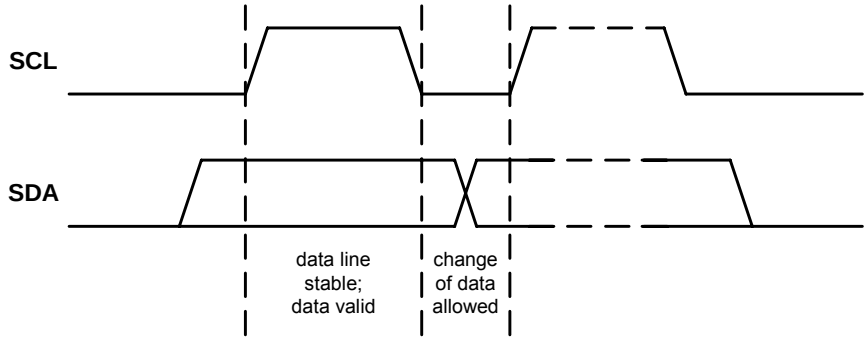
The first byte after the START procedure

Data Transfer

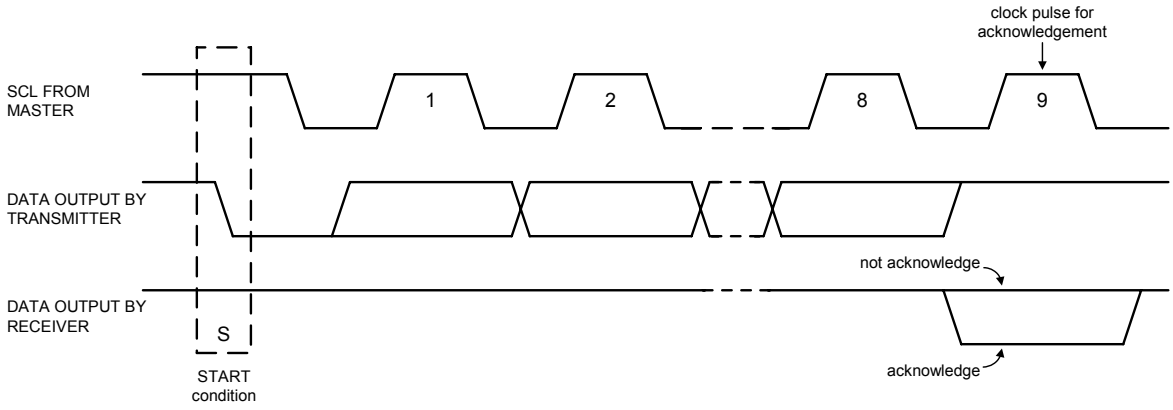
Once successful slave addressing has been achieved, the data transfer can proceed on a byte-by-byte basis in the direction specified by the RW bit sent by the master. Each transferred byte is followed by an acknowledge bit on the 9th SCL clock cycle. If the slave signals a **Not Acknowledge (NACK)**, 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 I^2C_TIP flag, indicating that a **Transfer is In Progress**. When the transfer is done the I^2C_TIP flag is cleared, the IF flag set if 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 I^2C_TIP flag is cleared.



Bit transfer on the I^2C-bus



Acknowledge on the I^2C-bus

6.14.2 I²C Serial Interface Control Registers Map

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

NOTE1: The reset value of I²C_WR0/1 is 0x3F only when SCR, SDR and SER are connected to pull high resistor.

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
I²C Interface 0				
I²C_CSR0	0xFFF8_6000	R/W	I²C0 Control and Status Register	0x0000_0000
I²C_DIVIDER0	0xFFF8_6004	R/W	I²C0 Clock Prescale Register	0x0000_0000
I²C_CMDR0	0xFFF8_6008	R/W	I²C0 Command Register	0x0000_0000
I²C_SWR0	0xFFF8_600C	R/W	I²C0 Software Mode Control Register	0x0000_003F
I²C_RxR0	0xFFF8_6010	R	I²C0 Data Receive Register	0x0000_0000
I²C_TxR0	0xFFF8_6014	R/W	I²C0 Data Transmit Register	0x0000_0000
I²C Interface 1				
I²C_CSR1	0xFFF8_6100	R/W	I²C1 Control and Status Register	0x0000_0000
I²C_DIVIDER1	0xFFF8_6104	R/W	I²C1 Clock Prescale Register	0x0000_0000
I²C_CMDR1	0xFFF8_6108	R/W	I²C1 Command Register	0x0000_0000
I²C_SWR1	0xFFF8_610C	R/W	I²C1 Software Mode Control Register	0x0000_003F
I²C_RxR1	0xFFF8_6110	R	I²C1 Data Receive Register	0x0000_0000
I²C_TxR1	0xFFF8_6114	R/W	I²C1 Data Transmit Register	0x0000_0000

I²C Control and Status Register 0/1 (I²C_CSR0/1)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
I²C_CSR0	0xFFF8_6000	R/W	I²C Control and Status Register 0	0x0000_0000
I²C_CSR1	0xFFF8_6100	R/W	I²C Control and Status 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				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	
[31:12]	**Reserved**	**Reserved**
[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.
[3]	**Reserved**	**Reserved**
[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** I²C Interrupt. 1 = **Enable** I²C Interrupt.
[0]	**I²C_EN**	**I²C Core Enable** 0 = **Disable** I²C core, serial bus outputs are controlled by SDW/SCW. 1 = **Enable** I²C core, serial bus outputs are controlled by I²C core.

I²C Prescale Register 0/1 (I²C_DIVIDER 0 /1)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
I²C_DIVIDER0	0xFFF8_6004	R/W	I²C Clock Prescale Register 0	0x0000_0000
I²C_DIVIDER1	0xFFF8_6104	R/W	I²C Clock Prescale 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
DIVIDER[15:8]							
7	6	5	4	3	2	1	0
DIVIDER[7:0]							

BITS	DESCRIPTIONS	
[15:0]	**DIVIDER**	**Clock Prescale Register** It is used to prescale the SCL clock line. Due to the structure of the I²C interface, the core uses a 5*SCL clock internally. The prescale register must be programmed to this 5*SCL frequency (minus 1). Change the value of the prescale register only when the "I²C_EN" bit is cleared. Example: pclk = 32MHz, desired SCL = 100KHz $prescale = \frac{32\ MHz}{5 * 100\ KHz} - 1 = 63\ (dec) = 3F\ (hex)$

I²C Command Register 0/1 (I²C_CMDR 0/1)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
I²C_CMDR0	0xFFF8_6008	R/W	I²C Command Register 0	0x0000_0000
I²C_CMDR1	0xFFF8_6108	R/W	I²C Command 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
Reserved			START	STOP	READ	WRITE	ACK

NOTE: Software can write this register only when I²C_EN = 1.

BITS	DESCRIPTIONS	
[31:5]	**Reserved**	**Reserved**
[4]	**START**	**Generate Start Condition** Generate (repeated) start condition on I²C bus.
[3]	**STOP**	**Generate Stop Condition** Generate stop condition on I²C 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 I²C 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.

I²C Software Mode Register 0/1(I²C_SWR 0/1)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
I²C_SWR0	0xFFF8_600C	R/W	I²C Software Mode Control Register 0	0x0000_003F
I²C_SWR1	0xFFF8_610C	R/W	I²C Software Mode Control Register 1	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		Reserved	SDR	SCR	Reserved	SDW	SCW

Note: This register is used as software mode of I²C. 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	
[31:6]	**Reserved**	**Reserved**
[5]	**Reserved**	**Reserved**
[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]	**Reserved**	**Reserved**
[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**.

I²C Data Receive Register 0/1 (I²C_RxR 0/1)

REGISTER	OFFSET	R/W	DESCRIPTION	RESET VALUE
I²C_RXR0	0xFFF8_6010	R	I²C Data Receive Register 0	0x0000_0000
I²C_RXR1	0xFFF8_6110	R	I²C Data Receive 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
Rx[7:0]							

BITS	DESCRIPTIONS	
[31:8]	**Reserved**	**Reserved**
[7:0]	**Rx**	**Data Receive Register** The last byte received via I²C bus will put on this register. The I²C core only used 8-bit receive buffer.

I²C Data Transmit Register 0/1 (I²C_TxR 0/1)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
I²C_TXR0	0xFFF8_6014	R/W	I²C Data Transmit Register	0x0000_0000
I²C_TXR1	0xFFF8_6114	R/W	I²C 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 I²C core used 32-bit transmit buffer and provide multi-byte transmit function. Set CSR[Tx_NUM] to a value that you want to transmit. I²C 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

6.15 Universal Serial Interface

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. 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:

- AMBA APB interface compatible
- Support USI (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 or falling edge of serial clock independently
- 1 slave/device select lines
- Fully static synchronous design with one clock domain

6.15.1 USI Timing Diagram

The timing diagram of USI is shown as following.

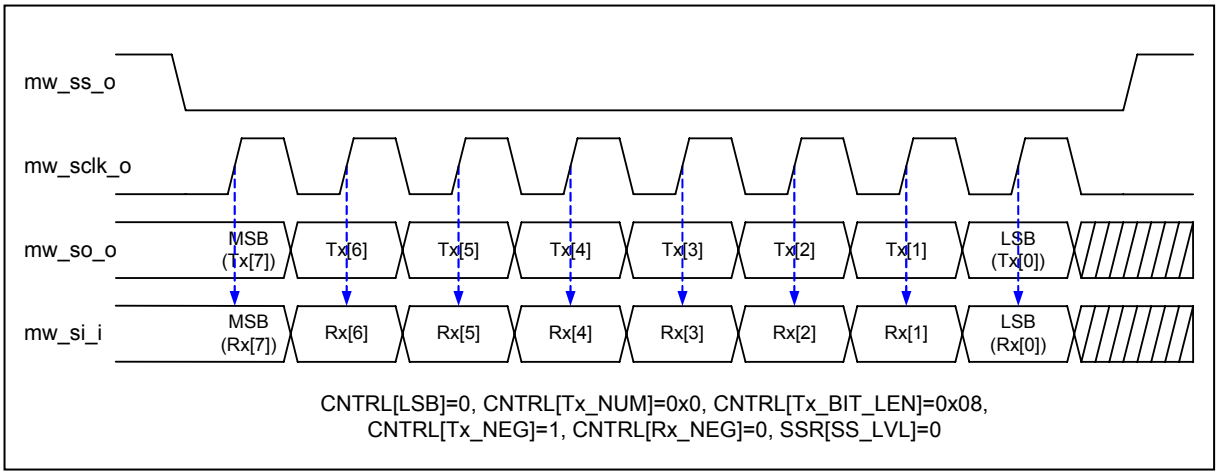


Figure 6.15.1 USI Timing

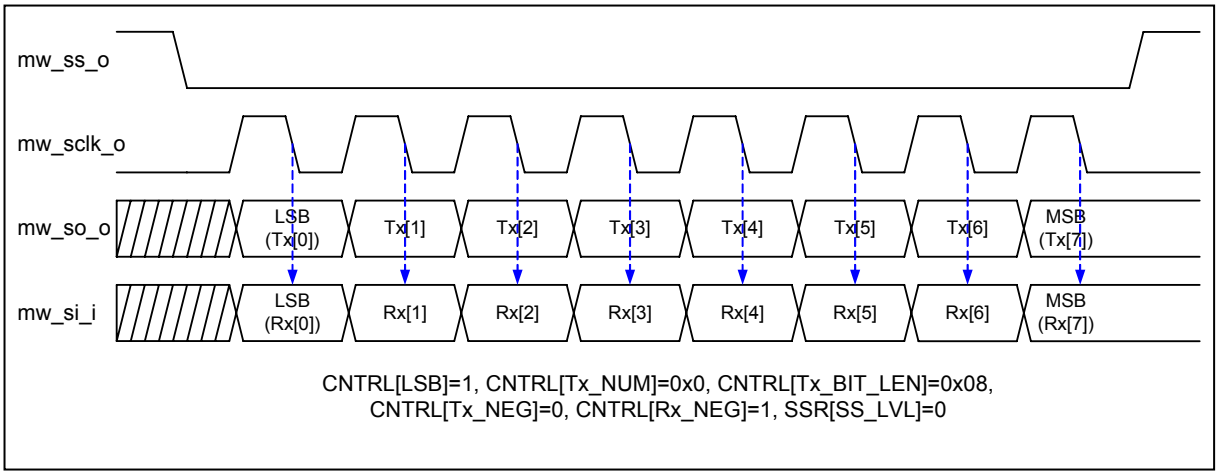


Figure 6.15.2 Alternate Phase SCLK Clock Timing

6.15.2 USI Registers Map

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

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
USI_CNTRL	0xFFF8_6200	R/W	Control and Status Register	0x0000_0004
USI_DIVIDER	0xFFF8_6204	R/W	Clock Divider Register	0x0000_0000
USI_SSR	0xFFF8_6208	R/W	Slave Select Register	0x0000_0000
Reserved	0xFFF8_620C	N/A	Reserved	N/A
USI_Rx0	0xFFF8_6210	R	Data Receive Register 0	0x0000_0000
USI_Rx1	0xFFF8_6214	R	Data Receive Register 1	0x0000_0000
USI_Rx2	0xFFF8_6218	R	Data Receive Register 2	0x0000_0000
USI_Rx3	0xFFF8_621C	R	Data Receive Register 3	0x0000_0000
USI_Tx0	0xFFF8_6210	W	Data Transmit Register 0	0x0000_0000
USI_Tx1	0xFFF8_6214	W	Data Transmit Register 1	0x0000_0000
USI_Tx2	0xFFF8_6218	W	Data Transmit Register 2	0x0000_0000
USI_Tx3	0xFFF8_621C	W	Data Transmit Register 3	0x0000_0000

NOTE 1: When software programs CNTRL, the GO_BUSY bit should be written last.

USI_Control and Status Register (USI_CNTRL)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
USI_CNTRL	0xFFF8_6200	R/W	USI Control and Status Register	0x0000_0004

31	30	29	28	27	26	25	24
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	
[31:18]	Reserved	Reserved
[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 = 0x1 … 3 SCLK clock cycle …… SLEEP = 0xe … 16 SCLK clock cycle SLEEP = 0xf … 17 SCLK clock cycle

Continued

BITS	DESCRIPTIONS	
[11]	**Reserved**	**Reserved**
[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(Microwire/SPI) master core has no effect.

USI Divider Register (USI_DIVIDER)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
USI_Divider	0xFFF8_6204	R/W	USI 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.

USI Slave Select Register (USI_SSR)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
USI_SSR	0xFFF8_6208	R/W	USI 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 slave select of the selected device must be set to its active level before starting any read or write transfer.

USI Data Receive Register 0/1/2/3 (USI_Rx0/1/2/3)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
USI_RX0	0xFFF8_6210	R	USI Data Receive Register 0	0x0000_0000
USI_RX1	0xFFF8_6214	R	USI Data Receive Register 1	0x0000_0000
USI_RX2	0xFFF8_6218	R	USI Data Receive Register 2	0x0000_0000
USI_RX3	0xFFF8_621C	R	USI 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 0/1/2/3 (Tx0/1/2/3)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
USI_TX0	0xFFF8_6210	W	USI Data Transmit Register 0	0x0000_0000
USI_TX1	0xFFF8_6214	W	USI Data Transmit Register 1	0x0000_0000
USI_TX2	0xFFF8_6218	W	USI Data Transmit Register 2	0x0000_0000
USI_TX3	0xFFF8_621C	W	USI 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 means 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.

6.16 PWM

The W90N745 have 4 channels PWM timers. They can be divided into two groups. Each group has 1 Prescaler, 1 clock divider, 2 clock selectors, 2 16-bit counters, 2 16-bit comparators, 1 Dead-Zone generator. They are all driven by PCLK (80 MHz). Each channel can be used as a timer and issue interrupt independently.

Two channels PWM timers in one group share the same prescaler. Clock divider provides each channel with 5 clock sources (1, 1/2, 1/4, 1/8, 1/16). Each channel receives its own clock signal from clock divider which receives clock from 8-bit prescaler. The 16-bit counter in each channel receive clock signal from clock selector and can be used to handle one PWM period. The 16-bit comparator compares number in counter with threshold number in register loaded previously to generate PWM duty cycle.

The clock signal from clock divider is called PWM clock. Dead-Zone generator utilize PWM clock as clock source. Once Dead-Zone generator is enabled, output of two PWM timer in one group is blocked. Two output pin are all used as Dead-Zone generator output signal to control off-chip power device.

To prevent PWM driving output pin with unsteady waveform, 16-bit counter and 16-bit comparator are implemented with double buffering feature. User can feel free to write data to counter buffer register and comparator buffer register without generating glitch.

When 16-bit down counter reaches zero, the interrupt request is generated to inform CPU that time is up. When counter reaches zero, if counter is set as toggle mode, it is reloaded automatically and start to generate next cycle. User can set counter as one-shot mode instead of toggle mode. If counter is set as one-shot mode, counter will stop and generate one interrupt request when it reaches zero.

The value of comparator is used for pulse width modulation. The counter control logic changes the output level when down-counter value matches the value of compare register.

The PWM timer features are shown as below:

- Two 8-bit prescalers and two clock dividers
- Four clock selectors
- Four 16-bit counters and four 16-bit comparators
- Two Dead-Zone generator

6.16.1 PWM Double Buffering and Reload Automatically

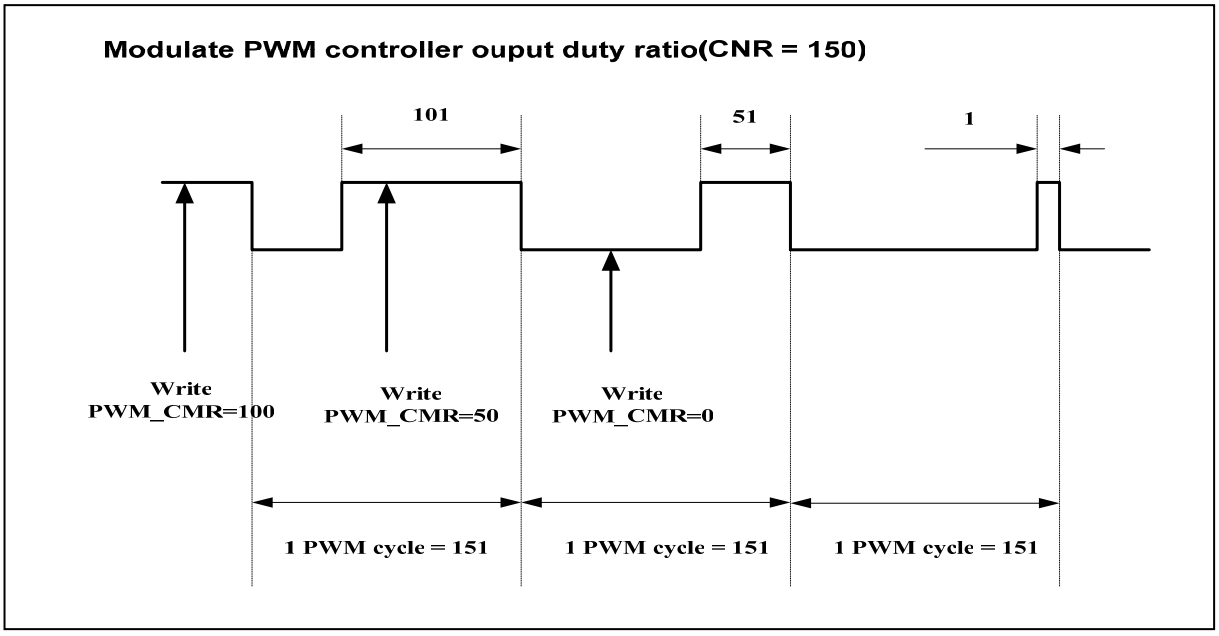
W90N745 PWM Timers have a double buffering function, enabling the reload value changed for next timer operation without stopping current timer operation. Although new timer value is set, current timer operation still operate successfully.

The counter value can be written into PWM_CNR0, PWM_CNR1, PWM_CNR2, PWM_CNR3 and current counter value can be read from PWM_PDR0, PWM_PDR1, PWM_PDR2, PWM_PDR3.

The auto-reload operation copies from PWM_CNR0, PWM_CNR1, PWM_CNR2, PWM_CNR3 to down-counter when down-counter reaches zero. If PWM_CNR0~3 are set as zero, counter will be halt when counter count to zero. If auto-reload bit is set as zero, counter will be stopped immediately.

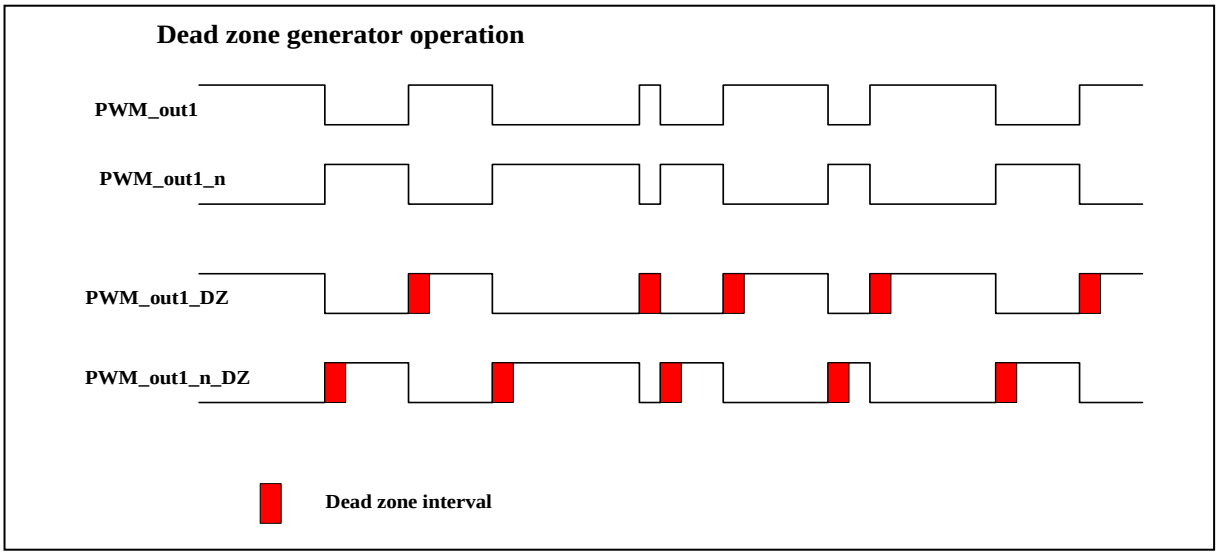
6.16.2 Modulate Duty Ratio

The double buffering function allows CMR written at any point in current cycle. The loaded value will take effect from next cycle.



6.16.3 Dead Zone Generator

W90N745 PWM is implemented with Dead Zone generator. They are built for power device protection. This function enables generation of a programmable time gap at the rising of PWM output waveform. User can program PWM_PPR [31:24] and PWM_PPR [23:16] to determine the Dead Zone interval.



6.16.4 PWM Timer Start Procedure

1. Setup clock selector (PWM_CSR)
2. Setup prescaler & dead zone interval (PWM_PPR)
3. Setup inverter on/off, dead zone generator on/off, toggle mode /one-shot mode, and PWM timer off. (PWM_PCR)
4. Setup comparator register (PWM_CMR)
5. Setup counter register (PWM_CNR)
6. Setup interrupt enable register (PWM_PIER)
7. Enable PWM timer (PWM_PCR)

6.16.5 PWM Timer Stop Procedure

Method 1 : Set 16-bit down counter(PWM_CNR) as 0, and monitor PWM_PDR. When PWM_PDR reaches to 0, disable PWM timer (PWM_PCR). (Recommended)

Method 2 : Set 16-bit down counter(PWM_CNR) as 0. When interrupt request happen, disable PWM timer (PWM_PCR). (Recommended)

Method 3 : Disable PWM timer directly (PWM_PCR). (Not recommended)

6.16.6 PWM Register Map

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
PWM_PPR	0xFFF8_7000	R/W	PWM Prescaler Register	0x0000_0000
PWM_CSR	0xFFF8_7004	R/W	PWM Clock Select Register	0x0000_0000
PWM_PCR	0xFFF8_7008	R/W	PWM Control Register	0x0000_0000
PWM_CNR0	0xFFF8_700C	R/W	PWM Counter Register 0	0x0000_0000
PWM_CMR0	0xFFF8_7010	R/W	PWM Comparator Register 0	0x0000_0000
PWM_PDR0	0xFFF8_7014	R	PWM Data Register 0	0x0000_0000
PWM_CNR1	0xFFF8_7018	R/W	PWM Counter Register 1	0x0000_0000
PWM_CMR1	0xFFF8_701C	R/W	PWM Comparator Register 1	0x0000_0000
PWM_PDR1	0xFFF8_7020	R	PWM Data Register 1	0x0000_0000
PWM_CNR2	0xFFF8_7024	R/W	PWM Counter Register 2	0x0000_0000
PWM_CMR2	0xFFF8_7028	R/W	PWM Comparator Register 2	0x0000_0000
PWM_PDR2	0xFFF8_702C	R	PWM Data Register 2	0x0000_0000
PWM_CNR3	0xFFF8_7030	R/W	PWM Counter Register 3	0x0000_0000
PWM_CMR3	0xFFF8_7034	R/W	PWM Comparator Register 3	0x0000_0000
PWM_PDR3	0xFFF8_7038	R	PWM Data Register 3	0x0000_0000
PWM_PIER	0xFFF8_703C	R/W	PWM Interrupt Enable Register	0x0000_0000
PWM_PIIR	0xFFF8_7040	R/C	PWM Interrupt Indication Register	0x0000_0000

PWM Prescaler Register (PWM_PPR)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
PWM_PPR	0xFFF8_7000	R/W	PWM Prescaler Register	0x0000_0000

31	30	29	28	27	26	25	24
DZI1							
23	22	21	20	19	18	17	16
DZI0							
15	14	13	12	11	10	9	8
CP1							
7	6	5	4	3	2	1	0
CP0							

BITS	DESCRIPTIONS	
[31:24]	**DZI1**	**DZI1:** Dead zone interval register 1, these 8-bit determine dead zone length. The 1 unit time of dead zone length is received from clock selector 2.
[23:16]	**DZI0**	**DZI0:** Dead zone interval register 0, these 8-bit determine dead zone length. The 1 unit time of dead zone length is received from clock selector 0.
[15:8]	**CP1**	**CP1 : Clock prescaler 1 for PWM Timer channel 2 & 3** Clock input is divided by (CP1 + 1) before it is fed to the counter. 2 & 3 If CP1=0, then the prescaler 1 output clock will be stopped.
[7:0]	**CP0**	**CP0 : Clock prescaler 0 for PWM Timer channel 0 & 1** Clock input is divided by (CP0 + 1) before it is fed to the counter. 0 & 1 If CP0=0, then the prescaler 0 output clock will be stopped.

PWM Clock Select Register (PWM_CSR)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
PWM_CSR	0xFFF8_7004	R/W	PWM Clock 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	CSR3			Reserved	CSR2		
7	6	5	4	3	2	1	0
Reserved	CSR1			Reserved	CSR0		

BITS	DESCRIPTIONS	
[14:12]	**CSR3**	Select clock input for channel 3
[10:8]	**CSR2**	Select clock input for channel 2.
[6:4]	**CSR1**	Select clock input for channel 1
[2:0]	**CSR0**	Select clock input for channel 0

CSR3	INPUT CLOCK DIVIDED BY
000	2
001	4
010	8
011	16
100	1

PWM Control Register (PWM_PCR)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
PWM_PCR	0xFFF8_7008	R/W	PWM Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved				PCR19	PCR18	PCR17	PCR16
15	14	13	12	11	10	9	8
PCR15	PCR14	PCR13	PCR12	PCR11	PCR10	PCR09	PCR08
7	6	5	4	3	2	1	0
PCR07	PCR06	PCR05	PCR04	PCR03	PCR02	PCR01	PCR00

BITS	DESCRIPTIONS	
[19]	**PCR 19**	**Channel 3 toggle/one shot mode** 1 = toggle mode 0 = one shot mode
[18]	**PCR 18**	**Channel 3 Inverter on/off** 1 = inverter on 0 = inverter off
[17]	**PCR 17**	Reserved
[16]	**PCR 16**	**Channel 3 enable/disable** 1 = enable 0 = disable
[15]	**PCR 15**	**Channel 2 toggle/one shot mode** **1 = toggle mode** **0 = one shot mode**
[14]	**PCR 14**	**Channel 2 Inverter on/off** **1 = inverter on** **0 = inverter off**
[13]	**PCR 13**	**Reserved**

Continued

BITS	DESCRIPTIONS	
[12]	**PCR 12**	**Channel 2 enable/disable** 1 = enable 0 = disable
[11]	**PCR 11**	**Channel 1 toggle/one shot mode** 1 = toggle mode 0 = one shot mode
[10]	**PCR 10**	**Channel 1 Inverter on/off** 1 = inverter on 0 = inverter off
[09]	**PCR 09**	**Reserved**
[08]	**PCR 08**	**Channel 1 enable/disable** 1 = enable 0 = disable
[07]	**PCR 07**	**Reserved**
[06]	**PCR 06**	**Reserved**
[05]	**PCR 05**	**Dead-Zone generator 1 enable/disable** 1 = enable dead-zone generator 0 = disable dead-zone generator
[04]	**PCR 04**	**Dead-Zone generator 0 enable/disable** 1 = enable dead-zone generator 0 = disable dead-zone generator
[03]	**PCR 03**	**Channel 0 toggle/one shot mode** **1 = toggle mode** **0 = one shot mode**
[02]	**PCR 02**	**Channel 0 Inverter on/off** **1 = inverter on** **0 = inverter off**
[01]	**PCR 01**	**Reserved**
[00]	**PCR 00**	**Channel 0 enable/disable** 1 = enable 0 = disable

PWM Counter Register 0/1/2/3 (PWM_CNR0/1/2/3)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
PWM_CNR0	0xFFF8_700C	R/W	PWM Counter Register 0	0x0000_0000
PWM_CNR1	0xFFF8_7018	R/W	PWM Counter Register 1	0x0000_0000
PWM_CNR2	0xFFF8_7024	R/W	PWM Counter Register 2	0x0000_0000
PWM_CNR3	0xFFF8_7030	R/W	PWM Counter Register 3	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**
CNRx[15:8]							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CNRx[7:0]							

BITS	DESCRIPTIONS	
[31:16]	**Reserved**	**-**
[15:0]	**CNRx**	**CNR:** PWM counter/timer buffer. Inserted data range: 65535~0. Unit: 1 PWM clock cycle **Note 1:** One PWM counter countdown interval = CNR + 1.If CNR is loaded as zero, PWM counter will be stopped. **Note 2:** Programmer can feel free to write data to CNR at any time, and it will be reloaded when PWM counter reaches zero.

PWM Comparator Register 0/1/2/3 (PWM_CMR0/1/2/3)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
PWM_CMR0	0xFFF8_7010	R/W	PWM Comparator Register 0	0x0000_0000
PWM_CMR1	0xFFF8_701C	R/W	PWM Comparator Register 1	0x0000_0000
PWM_CMR2	0xFFF8_7028	R/W	PWM Comparator Register 2	0x0000_0000
PWM_CMR3	0xFFF8_7034	R/W	PWM Comparator Register 3	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**
CMRx[15:8]							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CMRx[7:0]							

BITS	DESCRIPTIONS	
[31:16]	**Reserved**	-
[15:0]	**CMRx**	**CMR:** PWM comparator register Inserted data range: 65535~0. CMR is used to determine PWM output duty ratio. **Note 1: PWM duty** = CMR + 1.If CMR is loaded as zero, PWM duty = 1 **Note 2:** Programmer can feel free to write data to CMR at any time, and it will be reloaded when PWM counter reaches zero.

PWM Data Register 0/1/2/3 (PWM_PDR 0/1/2/3)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
PWM_PDR0	0xFFF8_7014	R	PWM Data Register 0	0x0000_0000
PWM_PDR1	0xFFF8_7020	R	PWM Data Register 1	0x0000_0000
PWM_PDR2	0xFFF8_702C	R	PWM Data Register 2	0x0000_0000
PWM_PDR3	0xFFF8_7038	R	PWM Data Register 3	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**
PDRx[15:8]							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PDRx[7:0]							

BITS	DESCRIPTIONS	
[31:16]	**Reserved**	-
[15:0]	**PDRx**	**PDR:** PWM Data register. User can monitor PDR to get current value in 16-bit down counter.

PWM Interrupt Enable Register (PWM_PIER)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
PWM_PIER	0xFFF8_703C	R/W	PWM 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				PIER3	PIER2	PIER1	PIER0

BITS	DESCRIPTIONS	
[31:4]	**Reserved**	-
[3]	**PIER3**	**Enable/Disable PWM counter channel 3 interrupt request** 1 = enable 0 = disable
[2]	**PIER2**	**Enable/Disable PWM counter channel 2 interrupt request** 1 = enable 0 = disable
[1]	**PIER1**	**Enable/Disable PWM counter channel 1 interrupt request** 1 = enable 0 = disable
[0]	**PIER0**	**Enable/Disable PWM counter channel 0 interrupt request** 1 = enable 0 = disable

PWM Interrupt Indication Register (PWM_PIIR)

REGISTER	ADDRESS	R/W/C	DESCRIPTION	RESET VALUE
PWM_PIIR	0xFFF8_7040	R/C	PWM Interrupt Indication 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				**PIIR3**	**PIIR2**	**PIIR1**	**PIIR0**

BITS	DESCRIPTIONS	
[3]	**PIIR3**	PWM counter channel 3 interrupt flag
[2]	**PIIR2**	PWM counter channel 2 interrupt flag
[1]	**PIIR1**	PWM counter channel 1 interrupt flag
[0]	**PIIR0**	PWM counter channel 0 interrupt flag
Note: User can clear each interrupt flag by writing a zero to corresponding bit in PIIR		

6.17 Keypad Interface

W90N745 Keypad Interface (**KPI**) is an APB slave with 4-row scan output and 8-column scan input. KPI scans an array up to 16x8 with an external 4 to 16 decoder. It can also be programmed to scan 8x8 or 4x8 key array. If the 4x8 array is selected then external decoder is not necessary because the scan signals are dived by W90N745 itself. For minimum pin counts application, an auxiliary priority encoder (TTL 74148) can be used to encode 8 columns input to 3 binary code and one indicator flag. Total 8 pins are required to implement 16x8 key scan.

Any 1 or 2 keys in the array that pressed are debounced and encoded. The keypad controller scan key matrix from ROW0 COL 0 → 1 → 2 …. → 7, ROW1 COL 0 → 1 → 2 … → 7 till to ROW 16 (or ROW 8 or ROW 4) COL 0 → 0 → 1 …. → 7. If more than 2 keys are pressed, only the keys or apparent keys in the array with the lowest address will be decoded.

KPI also supports 2-keys scan interrupt and specified 3-keys interrupt or chip reset. If the 3 pressed keys matches with the 3 keys defined in **KPI3KCONF**, it will generate an interrupt or chip reset to nWDOG reset output depend on the **ENRST** setting. The interrupt is generated whenever the scanner detects a key is pressed. The interrupt conditions are 1 key, 2 keys and 3keys.

W90N745 provides one keypad connecting interface. The interface is in Ethernet RMII PHY interface and I^2C interface 2 SDA1, SCL1 (GPIO18-27). Software should set KPSEL bit to “0” in KPICONF register to select MAC PHY interface for connecting keypad matrix.

The keypad interface has the following features:

- maximum 16x8 array
- programmable debounce time
- low-power wakeup mode
- programmable three-key reset

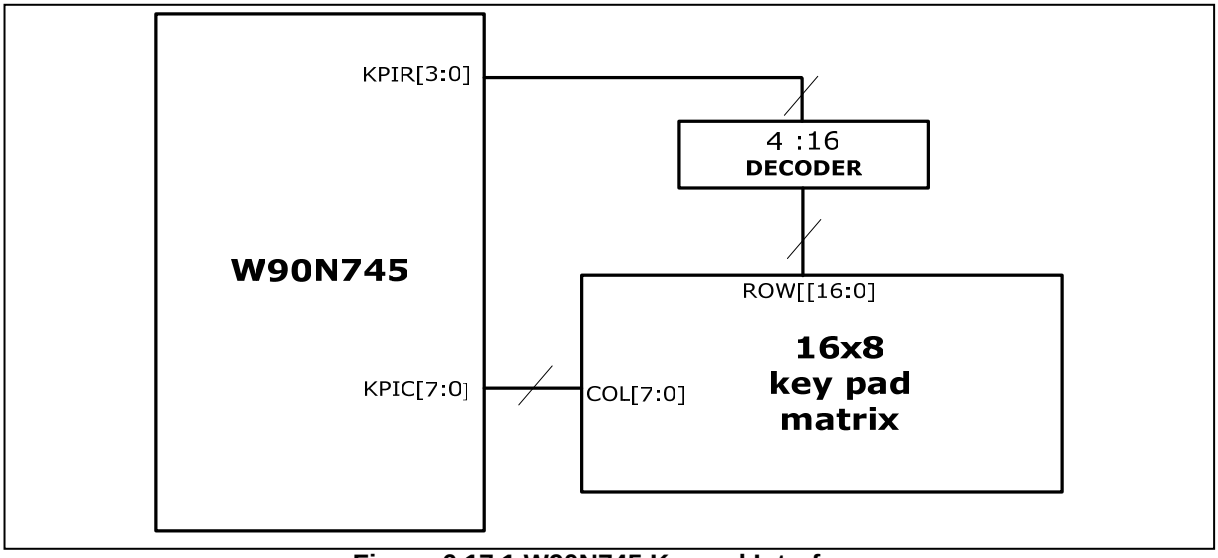


Figure 6.17.1 W90N745 Keypad Interface

6.17.1 Keypad Interface Register Map

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
KPICONF	0xFFF8_8000	R/W	Keypad Controller Configuration Register	0x0000_0000
KPI3KCONF	0xFFF8_8004	R/W	Keypad Controller 3-keys Configuration Register	0x0000_0000
KPILPCONF	0xFFF8_8008	R/W	Keypad Controller Low Power Configuration Register	0x0000_0000
KPISTATUS	0xFFF8_800C	R/O	Keypad Controller Status Register	0x0000_0000

6.17.2 Register Description

Keypad Controller Configuration Register (KPI_CONF)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
KPICONF	0xFFF8_8000	R/O	key pad configuration register	0x0000_0000

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

BITS	DESCRIPTION	
[31:22]	RESERVED	-
[21]	ENCODE	**Enable Encode Function** If an auxiliary 8 to 3 encoder is used to minimize keypad interface pin counts, user can connect encoder data to KPI_COL[2:0] and indicator flag (low active) to KPI_COL[3]. 1 = enable encoder function 0 = default. (8 column inputs)
[20]	ODEN	**Open Drain Enable** If there are more than one key are pressed in the same column, then “short-circuit” will appear between active scan and inactive scan row. Software can set this bit HIGH to enable scan output KPI_ROW[3:0] pins work as “open-drain” to avoid the “short-circuit”. 1 = Open drain 0 = push-pull driver
[19]	KPSEL	**Key pad select** Software should set this bit to “0” to select MAC PHY interface for connecting keypad matrix. 0 = pin #53~55, #57~60, #62~65 #19 and #20 are used as keypad interface

Continued

<table>
<tr><th>BITS</th><th colspan="2">DESCRIPTION</th></tr>
<tr><td>[18]</td><td>ENKP</td><td>Key pad scan enable
Setting this bit high enable the key scan function.
1 = enable key pad scan
0 = disable key pad scan</td></tr>
<tr><td>[17:16]</td><td>KSIZE</td><td>Key array size
<table><tr><th>KSIZE</th><th>Key array size</th></tr><tr><td>2’b00</td><td>4x8, 3x8, 2x8, 1x8</td></tr><tr><td>2’b01</td><td>8x8, 7x8, 6x8, 5x8</td></tr><tr><td>2’b1x</td><td>16x8, 15x8, 14x8, 13x8, 12x8, 11x8, 10x8, 9x8</td></tr></table></td></tr>
<tr><td>[15:8]</td><td>DBTC</td><td>Debounce terminal count
Debounce counter counts the number of consecutive scans that decoded the same keys. When de-bounce counter counter is equal to terminal count it will generate a key scan interrupt.</td></tr>
<tr><td>[7:0]</td><td>PRESCALE</td><td>Row scan cycle pre-scale value
This value is used to prescale row scan cycle. The prescale counter is clocked by 0.9375MHz clock.
Key array scan time = 1.067us x PRESCALE x16 ROWS
The following example is the scan time for PRESCALE = 0xFA
Tscan_time = 1.067us x 250 x16 = 4.268ms
If debounce terminal count = 0x05, key detection interrupt is fired in approximately 21.34ms. The array scan time can range from 17.07us to 1.118 sec.</td></tr>
</table>

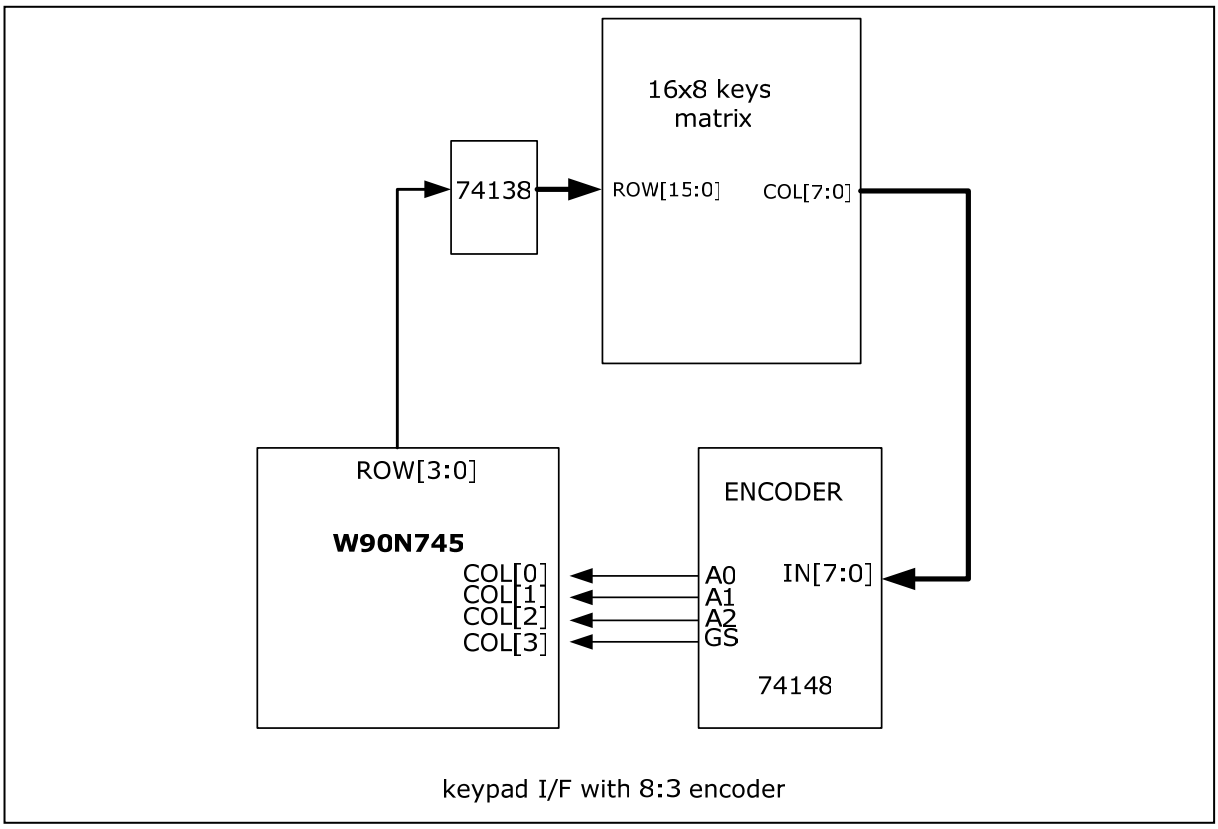


Figure 6.17.2 Keypad Interface with row decoder and column encoder

Keypad Controller 3-keys Configuration Register (KPI3KCONF)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
KPI3KCONF	0xFFF8_8004	W/R	three-key configuration register	0x0000_0000

31	30	29	28	27	26	25	24
RESERVED						EN3KY	ENRST
23	22	21	20	19	18	17	16
RESERVED	K32R				K32C		
15	14	13	12	11	10	9	8
RESERVED	K31R				K31C		
7	6	5	4	3	2	1	0
RESERVED	K30R				K30C		

<table>
<tr><th>BITS</th><th colspan="2">DESCRIPTION</th></tr>
<tr><td>[31:26]</td><td>RESERVED</td><td>-</td></tr>
<tr><td>[25]</td><td>EN3KY</td><td>Enable three-keys detection
Setting this bit enables hardware to detect 3 keys specified by software</td></tr>
<tr><td>[24]</td><td>ENRST</td><td>Enable three-key reset
Setting this bit enable hardware reset when three-key is detected.
<table><tr><th>EN3KY</th><th>ENRST</th><th>Function</th></tr><tr><td>0</td><td>X</td><td>three-key function is disable</td></tr><tr><td>1</td><td>0</td><td>generate three-key interrupt</td></tr><tr><td>1</td><td>1</td><td>hardware reset by three-key-reset</td></tr></table></td></tr>
<tr><td>[23]</td><td>RESERVED</td><td>-</td></tr>
<tr><td>[22:19]</td><td>K32R</td><td>The #2 key row address
The #2 means the row address and the column address is the highest of the specified 3-kyes.</td></tr>
<tr><td>[18:16]</td><td>K32C</td><td>The #2 key column address</td></tr>
<tr><td>[15]</td><td>RESERVED</td><td>-</td></tr>
<tr><td>[14:11]</td><td>K31R</td><td>The #1 key row address
The #1 means the row address and the column address is the 2nd of the specified 3-kyes.</td></tr>
<tr><td>[10:8]</td><td>K31C</td><td>The #1 key column address</td></tr>
<tr><td>[7]</td><td>RESERVED</td><td>-</td></tr>
<tr><td>[6:3]</td><td>K30R</td><td>The #0 key row address
The #0 means the row address and the column address is the lowest of the specified 3-kyes.</td></tr>
<tr><td>[2:0]</td><td>K30C</td><td>The #0 key column address</td></tr>
</table>

Application Note: Due to hardware scan from {row[0], col[0]}, {row[0], col[1]}, …, to {row[15], col[7]} the {K30R,K30C} should be filled the lowest address of the three-keys. For example, if {2,0} {4,6}, {1,3} keys are defined as three-keys. Software should set {K30R, K30C} = {1, 3}, {K31R, K31C} = {2, 0} and **{K32R, K32C} = {4, 6}.**

KeyPad Interface Low Power Mode Configuration Register (KPILPCONF)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
KPILPCOF	0xFFF8_8008	W/R	Low power configuration register	0x0000_0000

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

BITS	DESCRIPTION	
[31:17]	RESERVED	-
[16]	WAKE	**Lower power wakeup enable** Setting this bit enables low power wakeup 1 = wakeup enable 0 = not enable
[15:8]	LPWCEN	**Low power wakeup column enable** Specify columns for low power wakeup. For example, if user wants to use keys in row N and column 0, 2, 5 to wake up W90N745, then the LPWCEN should be fill 8'b00100101.
[7:4]	RESERVED	-
[3:0]	LPWR	**Low power wakeup row address** Define the row address keys used to wakeup. For 16x8 or 8x8 (with 4:16 or 3:8 decoder) keypad key configuration, LPWR means "Hex" code but for 4x8 (without decoder), LPWR means "binary" code. For example, if user wants to use all keys on row 3 of 16x8 keypad to wakeup W90N745, then 0x3 should be fill into this register but for 4x8 keypad it should be filled as 4'b1000.

Key Pad Interface Status Register (KPISTATUS)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
KPISTATUS	0xFFF8_800C	R/O	key pad status register	0x0000_0000

31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED		INT	3KRST	PDWAKE	3KEY	2KEY	1KEY
15	14	13	12	11	10	9	8
RESERVED	KEY1R				KEY1C		
7	6	5	4	3	2	1	0
RESERVED	KEY0R				KEY0C		

BITS	DESCRIPTION	
[31:22]	RESERVED	-
[21]	INT	**Key interrupt** This bit indicates the key scan interrupt is active and that one or two keys have changed status. The interrupt also occur when the three specified keys are detected if ENRST bit in KPI3KFCON is cleared. It will be cleared by hardware automatically when software read KPISTATUS register.
[20]	3KRST	**3-Keys reset flag** This bit is a record flag for software reference, it will be set after 3-keys reset occur. 1 = 3 keys reset 0 = not reset. This bit is cleared while it is read.
[19]	PDWAKE	**Power Down Wakeup flag** This flag indicates the chip is wakeup from power down by keypad 1 =wakeup up by keypad 0 = not wakeup
[18]	3KEY	**Specified three-key is detected.** This flag indicates specified-three-keys was detected. Software can read this bit to know the keypad interrupt is 3 key or not.

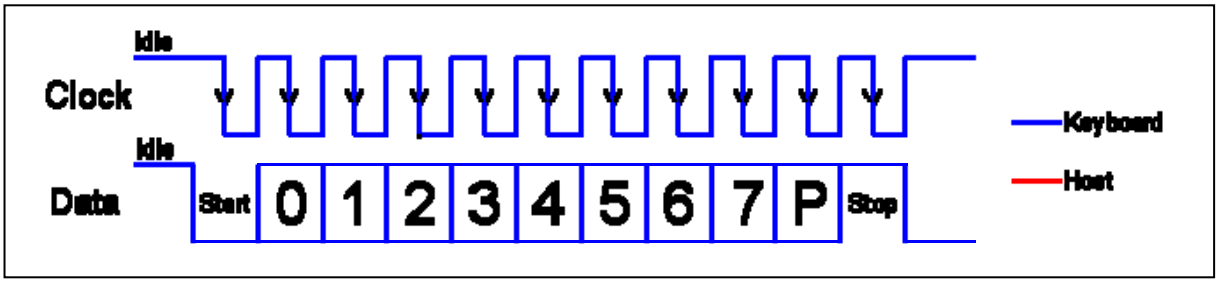
Continued

BITS	DESCRIPTION	
[17]	2KEY	**Double-key press** This bit indicates that 2 keys have been detected. Software can read {KEY1R, KEY1C} and {KEY0R, KEY0C} to know which two keys are pressed.
[16]	1KEY	**Single-key press** This bit indicates that 1 key has been detected. Software can read {KEY0R, KEY0C} to know which key is pressed.
[15]	RESERVED	-
[14:11]	KEY1R	**KEY1 row address** This value indicates key1 row address. The keypad controller scan keypad matrix from row 0, column0 →1 →2 …. → 7 and then row1 column 0 → 1 → 2 →7 so the lowest key address will be stored in {KEY0R, KEY0C}. This register stores the 2nd address, if more than one key is pressed.
[10:8]	KEY1C	**KEY1 column address** This value indicates key1 column address..
[7]	RESERVED	-
[6:3]	KEY0R	**KEY1 row address** This value indicates key0 row address. This value indicates key0 row address. This value indicates key1 row address. The keypad controller scan keypad matrix from row 0, column0 →1 →2 …. → 7 and then row1 col 0 → 1 → 2 → …→ 7 still to row16 (or 8, or 4) column 0 → 1 → 2 ….. → 7 so the lowest key address will be stored in {KEY0R, KEY0C}.
[2:0]	KEY0C	**KEY1 column address** This value indicates key0 row address.

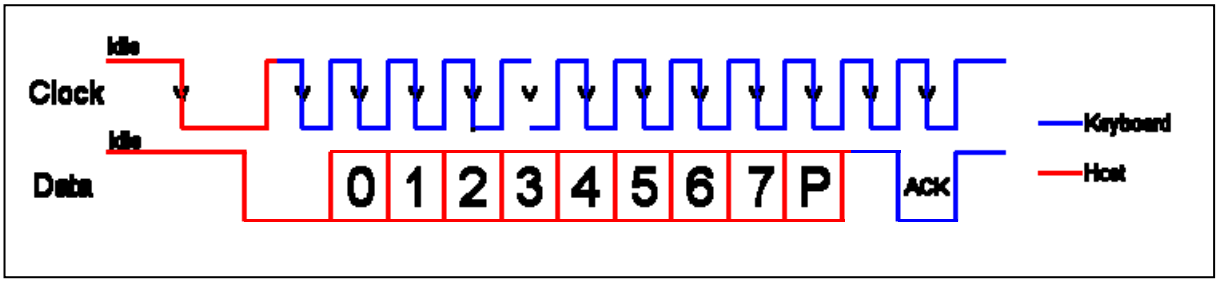
6.18 PS2 Host Interface Controller

W90N745 PS2 host controller interface is an APB slave consisted of PS2 protocol. It is used to connect to your IBM keyboard or other device through PS2 interface. For example, the IBM keyboard will sends scan codes to the host controller, and the scan codes will tell your Keyboard Bios what keys you have pressed or released. Besides Scan codes, commands can also be sent to the keyboard from host. The most common commands would be the setting/resetting of the status indicators (i.e. the Num lock, Caps Lock & Scroll Lock LEDs).

The PS2 interface implements a bi-directional protocol. The keyboard can send data to the Host and the Host can send data to the Keyboard using two PS2 Clock and PS2 Data lines. Both the PS2 Clock and Data lines are Open Collector bi-directional I/O lines. The Host has the ultimate priority over direction. The keyboard is free to send data to the host when both the PS2 Data and PS2 Clock lines are high (Idle). If the host takes the PS2 Clock line low, the keyboard will buffer any data until the PS2 Clock is released, ie goes high. The transmission of data in the forward direction, ie Keyboard to Host is done with a frame of 11 bits. The first bit is a Start Bit (Logic 0) followed by 8 data bits (LSB First), one Parity Bit (Odd Parity) and a Stop Bit (Logic 1). Each bit should be read on the falling edge of the clock. The Keyboard will generate the clock. The frequency of the clock signal typically ranges from 20 to 30 KHz.



The Host to Keyboard Protocol is initiated by taking the PS2 data line low. It is common to take the PS2 Clock line low for more than 60us and then the KBD data line is taken low, while the KBD clock line is released. After that, the keyboard will start generating a clock signal on its PS2 clock line. After the first falling edge has been detected, host will load the first data bit on the PS2 Data line. This bit will be read into the keyboard on the next falling edge, after which host place the next bit of data. This process is repeated for the 8 data bits. It will follow an Odd Parity Bit after the data byte.



6.18.1 PS2 Host Controller Interface Register Map

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
PS2CMD	0xFFF8_9000	R/W	PS2 Host Controller Command Register	0x0000_0000
PS2STS	0xFFF8_9004	R/W	PS2 Host Controller Status Register	0x0000_0000
PS2SCANCODE	0xFFF8_9008	RO	PS2 Host Controller RX Scan Code Register	0x0000_0000
PS2ASCII	0xFFF8_900C	RO	PS2 Host Controller RX ASCII Code Register	0x0000_0000

6.18.2 Register Description

PS2 Host Controller Command Register (PS2_CMD)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
PS2CMD	0xFFF8_9000	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						TRAP_SHIFT	EnCMD
7	6	5	4	3	2	1	0
PS2CMD							

BITS	DESCRIPTIONS	
[31:10]	RESERVED	-
[9]	**TRAP_SHIFT**	**Trap Shift Key Output to Scan Code Register** If the shift key scan code (0x12 0r 0x59) is received by host, software can indicate host whether to update to scan code register or not. No ASCII or SCAN codes will be reported for the shift keys if this bit is set. In this condition, host will only report the shift keys at the RX_shift_key bit of Status register and no interrupt will occur for the shift keys. This is useful for those who wish to use the ASCII data stream and don't want to "manually" filter out the shift key codes. This bit is clear by default.
[8]	**EnCMD**	**Enable write PS2 Host Controller Commands** **This bit enables the write function of Host controller comr device. Set this bit will start the write process of PS2CMD con hardware will automatically clear this bit while write pro finished.**
[7:0]	**PS2CMD**	**PS2 Host Controller Commands** **This command filed is sent by the Host to the Keyboard. Tl common command would be the setting/resetting of the Indicators (i.e. the Num lock, Caps Lock & Scroll Lock LEDs).**

PS2 Host Controller Status Register (PS2_STS)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
PS2STS	0xFFF8_9004	R/W	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		TX_err	TX_IRQ	RESERVED			RX_IRQ

BITS	DESCRIPTIONS	
[31:6]	RESERVED	-
[5]	**TX_err**	This Transmit Error Status bit indicates software that device doesn't response ACK after Host wrote a command to it. This bit is valid when TX_IRQ is asserted. It will automatically reset after software starts next command writing process. This bit is read only.
[4]	**TX_IRQ**	This Transmit Complete Interrupt bit indicates software that the process of Host controller writing command to device is finished. Software needs to write one to this bit to clear this interrupt.
[3:1]		Reserved
[0]	**RX_IRQ**	This Receive Interrupt bit indicates software that Host controller receives one byte data from device. This data is stored at PS2_SCANCODE register. Software needs to write one to this bit to clear this interrupt after reading receiving data in RX_SCAN_CODE register. Note that the reception of the Extend (0xE0) and Release (0xF0) scan code will not cause an interrupt by host. The case of the shift key codes will be determined by the TRAP_SHIFT bit of PS2_CMD register.

PS2 Host Controller RX Scan Code Register (PS2_SCANCODE)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
PS2SCANCODE	0xFFFF_9008	R/W	PS2 Host RX Scan Code 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					RX_shift_key	RX_release	RX_extend
7	6	5	4	3	2	1	0
RX_SCAN_CODE							

BITS	DESCRIPTIONS	
[31:11]	RESERVED	-
[10]	RX_shift_key	This Receive Shift Key bit indicates that left or right shift key on the keyboard is hold. This bit is read only and will clear by host when the release shift key codes are received.
[9]	RX_release	Receive Released Byte When one key has been released, the keyboard will send F0 (hex) to inform Host controller. This bit indicates software that Host controller receives release byte (F0). This bit is read only and will update when host has received next data byte.
[8]	RX_extend	Receive Extend Byte A handful of the keys on keyboard are extended keys and thus require two more scan code. These keys are preceded by an E0 (hex). This bit indicates software that Host controller receives extended byte (E0). This bit is read only and will update when host has received next data byte.
[7:0]	RX_SCAN_CODE	PS2 Host Controller Received Data Field This field stores the original data content transmitted from device. This filed is valid when RX_IRQ is asserted. Note that host will not report “Extend” or “Release” scan code to this field and not generate interrupt if they are received by host, i.e. 0xE0 and 0xF0. The case of the shift key codes will be determined by the TRAP_SHIFT bit of PS2_CMD register.

PS2 Host Controller RX ASCII Code Register (PS2_ASCII)

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
PS2ASCII	0xFFF8_900C	R/W	PS2 Host RX ASCII Code 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_ASCII_CODE							

BITS	DESCRIPTIONS	
[31:8]	RESERVED	-
[7:0]	RX_ASCII_CODE	PS2 Host Controller Received Data Filed This field stores the ASCII data content transmitted from device. Therefore, this part translates the scan code into an ASCII value. It will be read as 0x2E when there is no ASCII code mapped to the scan code stored in RX_SCAN_CODE register. This filed is valid when RX_IRQ is asserted.

7. ELECTRICAL SPECIFICATIONS

7.1 Absolute Maximum Ratings

Ambient temperature	-40 °C ~ +85°C
Storage temperature	-40 °C ~ +125°C
Voltage on any pin	-0.5V ~ 6V
Power supply voltage (Core logic)	-0.5V ~ 1.92V
Power supply voltage (IO Buffer)	-0.5V ~ 3.6V
Injection current (latch-up testing)	100mA
Crystal Frequency	4MHz ~ 30MHz

7.2 DC Specifications

7.2.1 Digital DC Characteristics

(Normal test conditions: VDD33/USBVDD = 3.3V+/- 0.3V, VDD18/DVDD18/AVDD18 = 1.8V+/- 0.18V

TA = -40 °C ~ +85 °C unless otherwise specified)

SYMBOL	PARAMETER	CONDITION	MIN.	MAX.	UNIT
VDD33/ USBVDD	Power Supply		3.00	3.60	V
VDD18/ DVDD18/ AVDD18	Power Supply		1.62	1.98	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		1.47	1.5	V
VT-	Schmitt trigger negative-going threshold		0.89	0.95	V
V_{OL}	Output Low Voltage	Depend on driving	-	0.4	V
V_{OH}	Output High Voltage	Depend on driving	2.4	-	V

Continued.

SYMBOL	PARAMETER	CONDITION	MIN.	MAX.	UNIT
I_{CC1}	1.8V Supply Current	F_{CPU} = 80MHz	-	150	mA
ICC2	3.3V Supply Current	FCPU = 80MHz	-	60	mA
I_{CCRTC}	RTC 1.8V Supply Current	FRTC = 32.768KHZ	-	7	uA
I_{IH}	Input High Current	V_{IN} = 2.4 V	-1	1	μA
I_{IL}	Input Low Current	V_{IN} = 0.4 V	-1	1	μA
I_{IHP}	Input High Current (pull-up)	V_{IN} = 2.4 V	-15	-10	μA
I_{ILP}	Input Low Current (pull-up)	V_{IN} = 0.4 V	-55	-25	μA
I_{IHD}	Input High Current (pull-down)	V_{IN} = 2.4 V	25	60	μA
I_{ILD}	Input Low Current (pull-down)	V_{IN} = 0.4 V	5	10	μA

Table 7.2.1 TSMC IO DC Characteristics

PARAMETER		MIN.	TYP.	MAX.
V_{IL}	Input Low Voltage	-0.3V		0.8V
V_{IH}	Input High Voltage	2V		5.5V
V_T	Threshold point	1.46V	1.59V	1.75V
V_{T+}	Schmitt trig low to high threshold point	1.47V	1.50V	1.50V
V_{T-}	Schmitt trig, high to low threshold point	0.90V	0.94V	0.96V
I_I	Input leakage current @V_I= 3.3V or 0V			+/- 10uA
I_{oz}	Tri-state output leakage current @Vo =3.3V or 0V			+/- 10UA
R_{PU}	Pull-up resister	44KΩ	66KΩ	110KΩ
R_{PD}	Pull-down resister	25KΩ	50KΩ	110KΩ
V_{OL}	Output low voltage @$_{IOL}$(min)			0.4V
V_{OH}	Output high voltage @I_{OH} (min)	2.4V		

Continued.

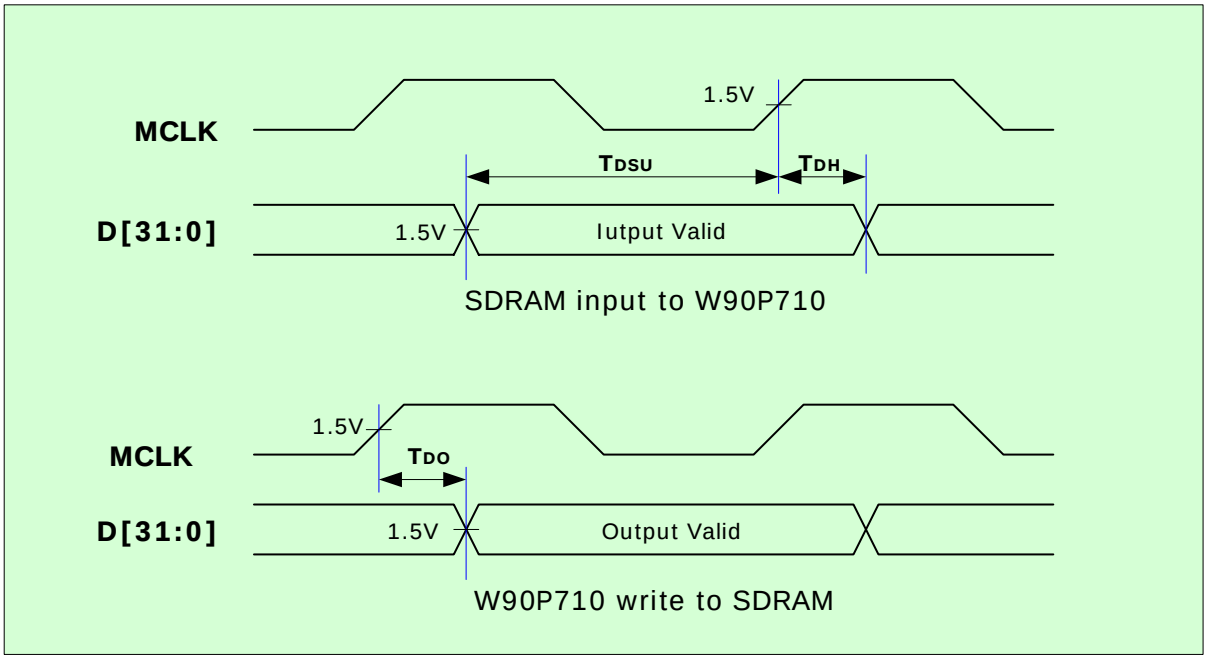
PARAMETER		MIN.	TYP.	MAX.
I_{OL}	Low level output current @V_{OL} = 0.4V 4mA	4.9mA	7.4mA	9.8mA
	Low level output current @V_{OL} = 0.4V 8mA	9.7mA	14.9mA	19.5mA
	Low level output current @V_{OL} = 0.4V 12mA	14.6mA	22.3mA	29.3mA
I_{OH}	High level output current @V_{OH} = 2.4V 4mA	6.3mA	12.8mA	21.2mA
	High level output current @V_{OH} = 2.4V 8mA	12.7mA	25.6mA	42.4mA
	High level output current @V_{OH} = 2.4V 12mA	19.0mA	38.4mA	63.6mA
NOTE: The values in this table are copied from TSMC 1P5M IO library tpz937g_240b silicon report. This table is just for reference. More precision DC vaule should refer to Alpha-Test result.				

7.2.2 USB Transceiver DC Characteristics

SYMBOL	PARAMETER	CONDITIONS	MIN.	MAX.	UNIT
V_{DI}	Differential Input Sensitivity	|DP – DM|	0.2		V
V_{CM}	Differential Common Mode Range	Includes V_{DI} range	0.8	2.5	V
V_{SE}	Single Ended Receiver Threshold		0.8	2.0	V
V_{OL}	Static Output Low Voltage	RL of 1.5 KΩ to 3.6 V		0.3	V
V_{OH}	Static Output High Voltage	RL of 15 KΩ to VSS	2.8	3.6	V
V_{CRS}	Output Signal Crossover Voltage		1.3	2.0	V
Z_{DRV}	Driver Output Resistance	Steady state drive	28	43	Ω
C_{IN}	Pin Capacitance			20	pF

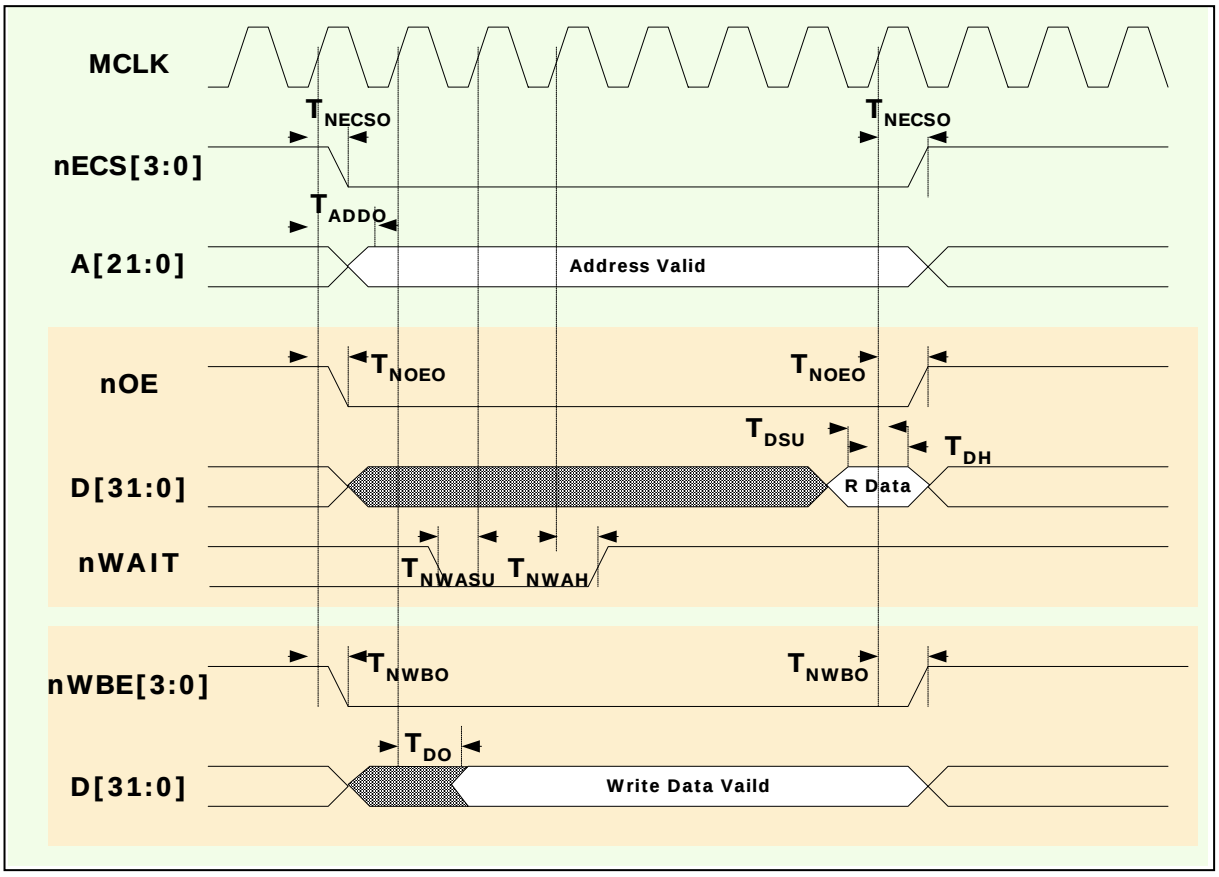
7.3 AC Specifications

7.3.1 EBI/SDRAM Interface AC Characteristics



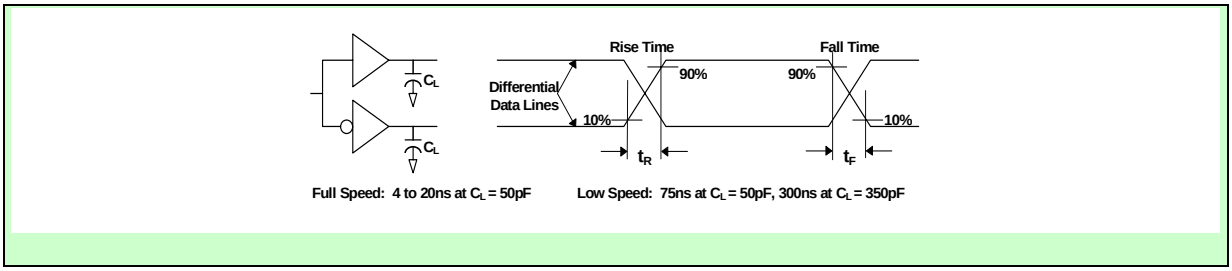
SYMBOL	PARAMETER	MIN.	MAX.	UNIT
T_{DSU}	D [31:0] Setup Time	2		ns
T_{DH}	D [31:0] Hold Time	2		ns
T_{DO}	D [31:0], A [24:0], nSCS [1:0], SDQM [3:0], CKE, nSWE, nSRAS, nSCAS	2	7	ns

7.3.2 EBI/(ROM/SRAM/External I/O) AC Characteristics



SYMBOL	DESCRIPTION	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	7		ns
T_{DSU}	Read Data Setup Time	0		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

7.3.3 USB Transceiver AC Characteristics



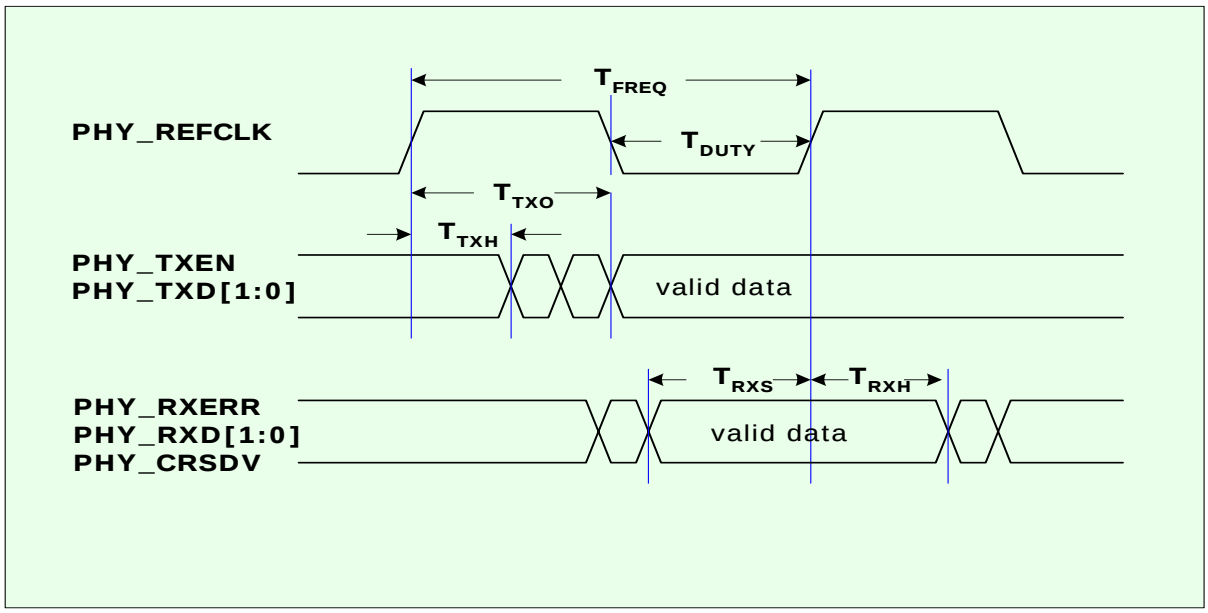
Data Signal Rise and Fall Time

USB Transceiver AC Characteristics

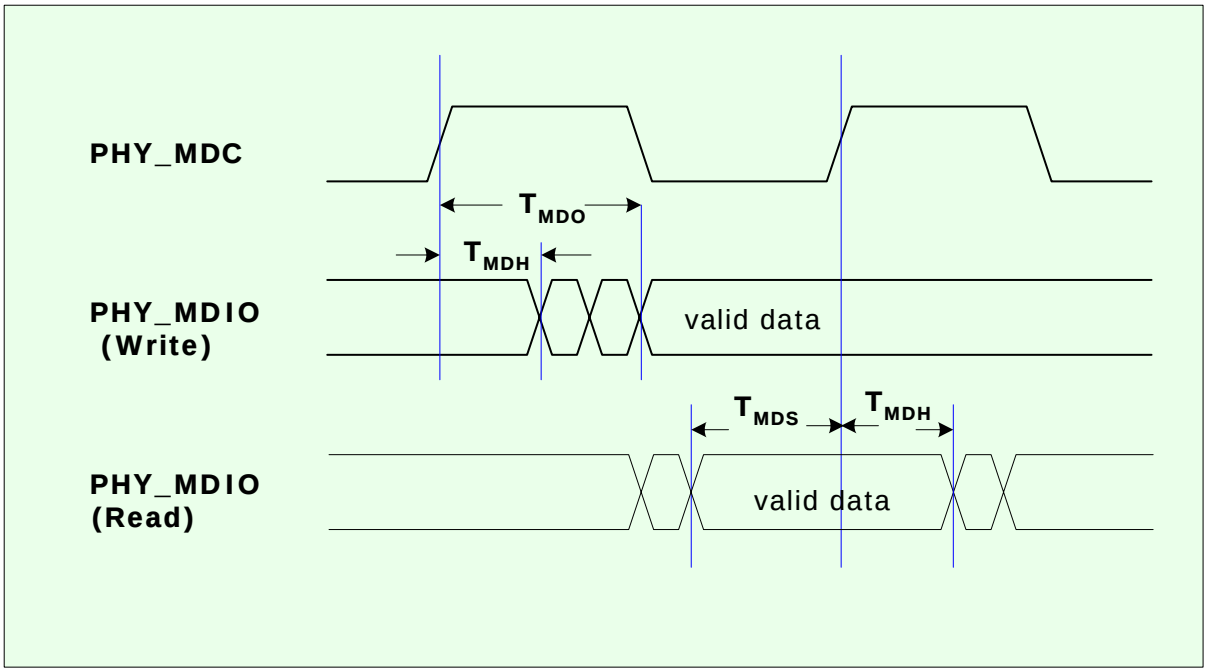
SYMBOL	DESCRIPTION	CONDITIONS	MIN	MAX	UNIT
T_R	Rise Time	CL = 50 pF	4	20	ns
T_F	Fall Time	CL = 50 pF	4	20	ns
T_{RFM}	Rise/Fall Time Matching		90	110	%
T_{DRATE}	Full Speed Data Rate	Average bit rate (12 Mb/s ± 0.25%)	11.97	12.03	Mbps

7.3.4 EMC RMII AC Characteristics

The signal timing characteristics conforms to the guidelines specified in IEEE Std. 802.3.

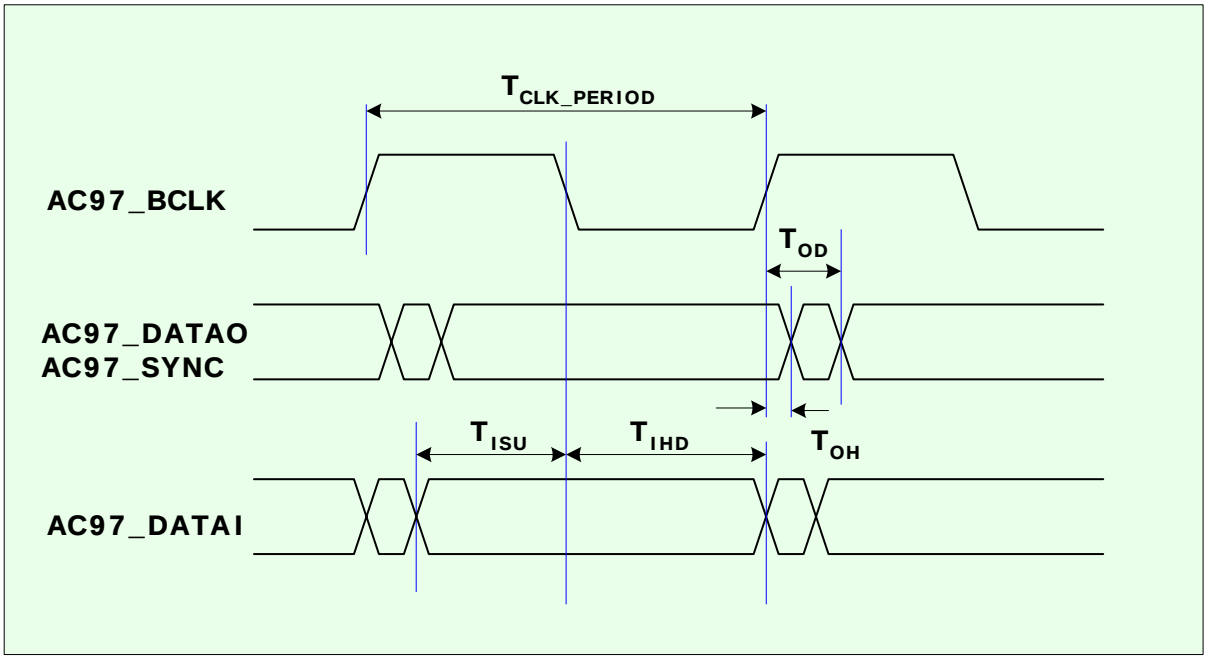


SYMBOL	DESCRIPTION	MIN	TYP	MAX	UNIT
T_{FREQ}	RMII reference clock frequency		50		MHz
T_{DUTY}	RMII clock duty	35%	50%	65%	ns
T_{TXO}	Transmit data output delay	5	-	15	ns
T_{TXH}	Transmit data hold time	2	-	-	ns
T_{RXS}	Receive data setup time	4	-	-	ns
T_{RXH}	Receive data hold time	2	-	-	ns

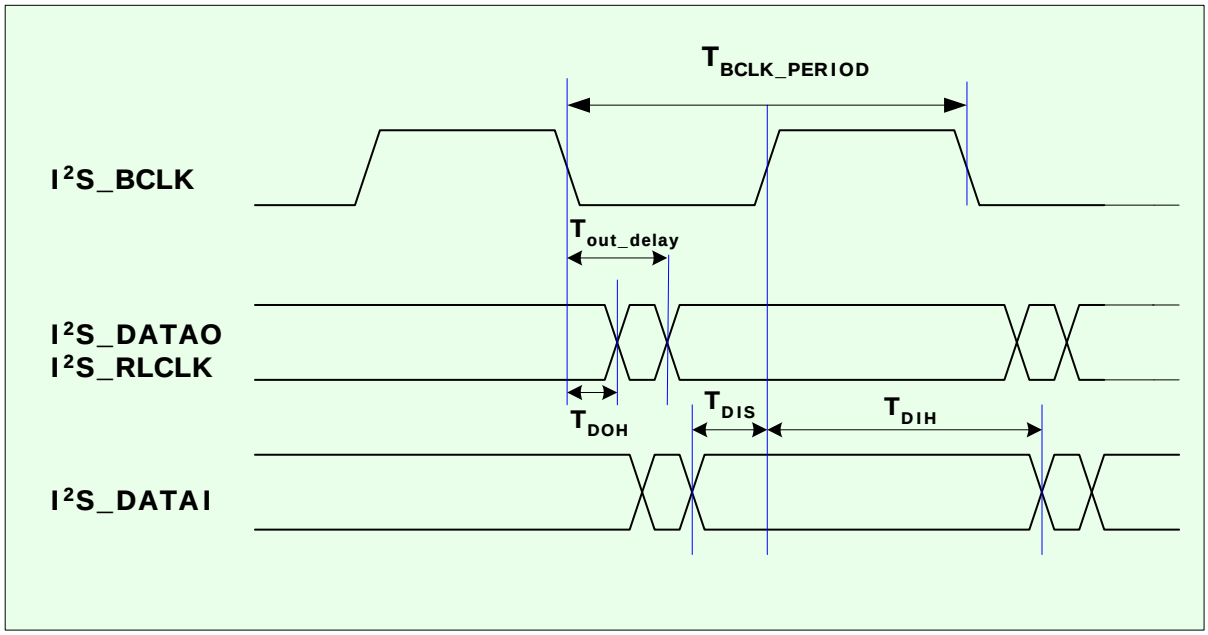


SYMBOL	DESCRIPTION	MIN	MAX	UNIT
T_{MDO}	MDIO Output Delay Time	0	15	ns
T_{MDSU}	MDIO Setup Time	5		ns
T_{MDH}	MDIO Hold Time	5		ns

7.3.5 AC97/I2S Interface AC Characteristics

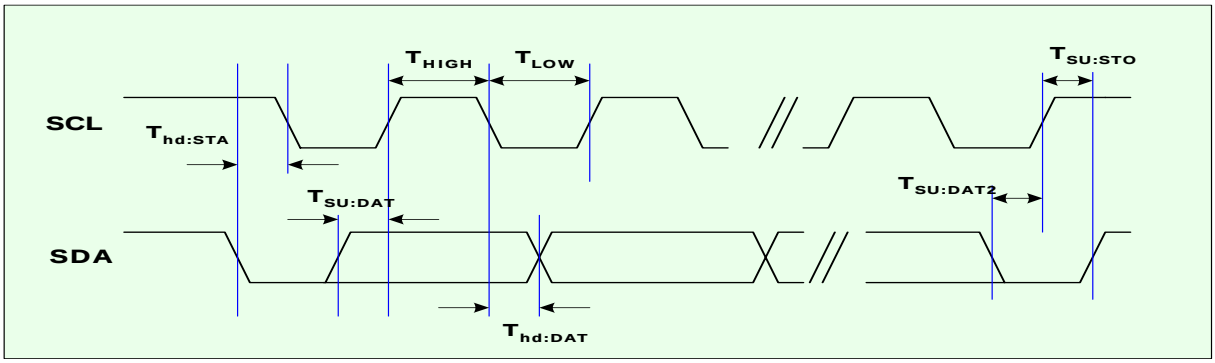


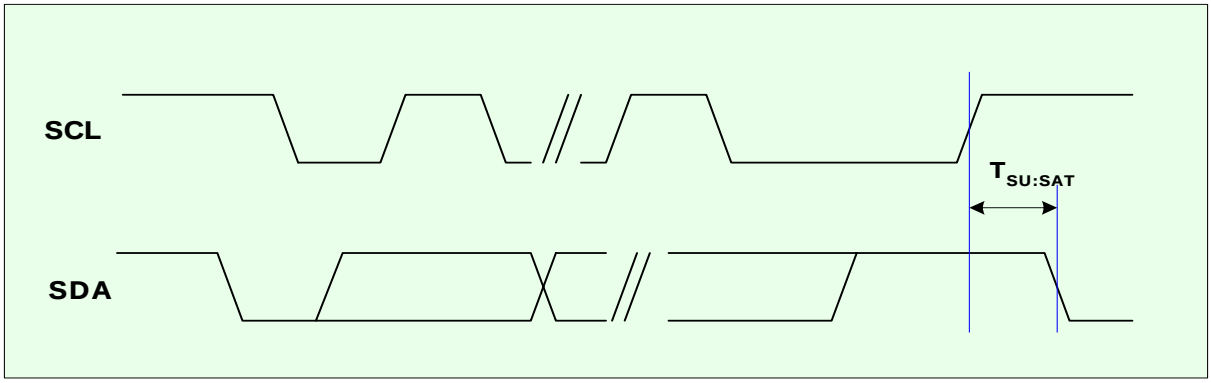
SYMBOLS	DESCRIPTION	MIN	TYP.	MAX	UNIT
T_{CLK_PERIOD}	AC97 Bit Clock Frequency	--	12.288	--	MHz
T_{OD}	AC97_DATAO and AC97_SYNC output delay from AC97_BCLK rising edge	--	--	30	ns
T_{OH}	AC97_DATAO and AC97_SYNC output hold time from AC97_BCLK rising edge	5	--	--	ns
T_{ISU}	AC97_DATAI input setup time to AC97_BCLK falling edge	10	--	--	ns
T_{IHD}	AC97_DATAI input hold time from AC97_BCLK falling edge	5	--	--	ns



SYMBOLS	DESCRIPTION	MIN	MAX	UNIT
T_{BCLK_PERIOD}	IIS Bit Clock Frequency	Note:depend on codec spec. and register setting		MHz
T_{out_delay}	IIS_DATAO and IIS_RLCLK output delay from IIS_BCLK falling edge	--	30	ns
T_{DOH}	IIS_DATAO and IIS_RLCLK data output hold time from IIS_BCLK falling edge	0	--	ns
T_{DIS}	IIS_DATAI input setup time to IIS_BCLK rising edge	10	--	ns
T_{DIH}	IIS_DATAI input hold time from IIS_BCLK rising edge	100	--	ns

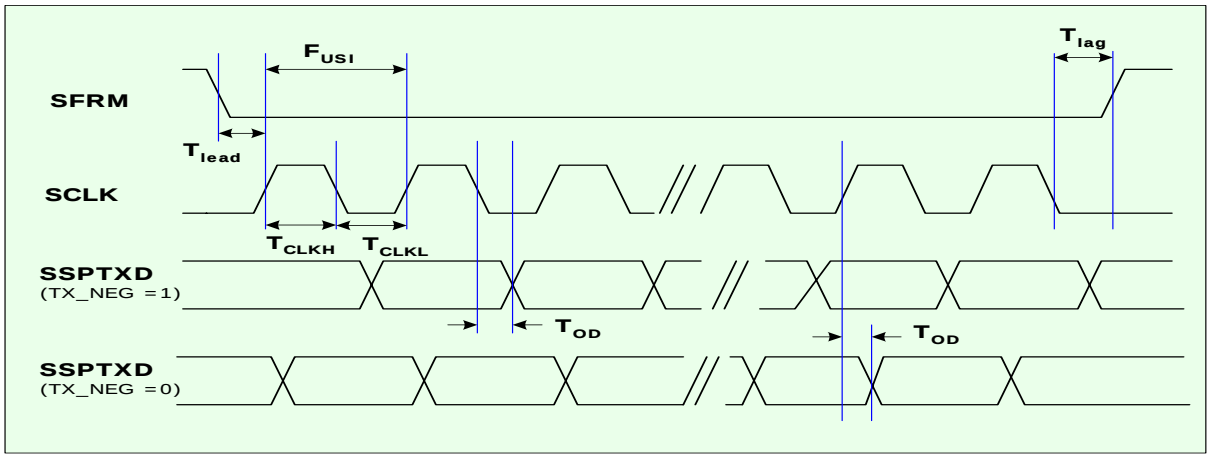
7.3.6 I²C Interface AC Characteristics

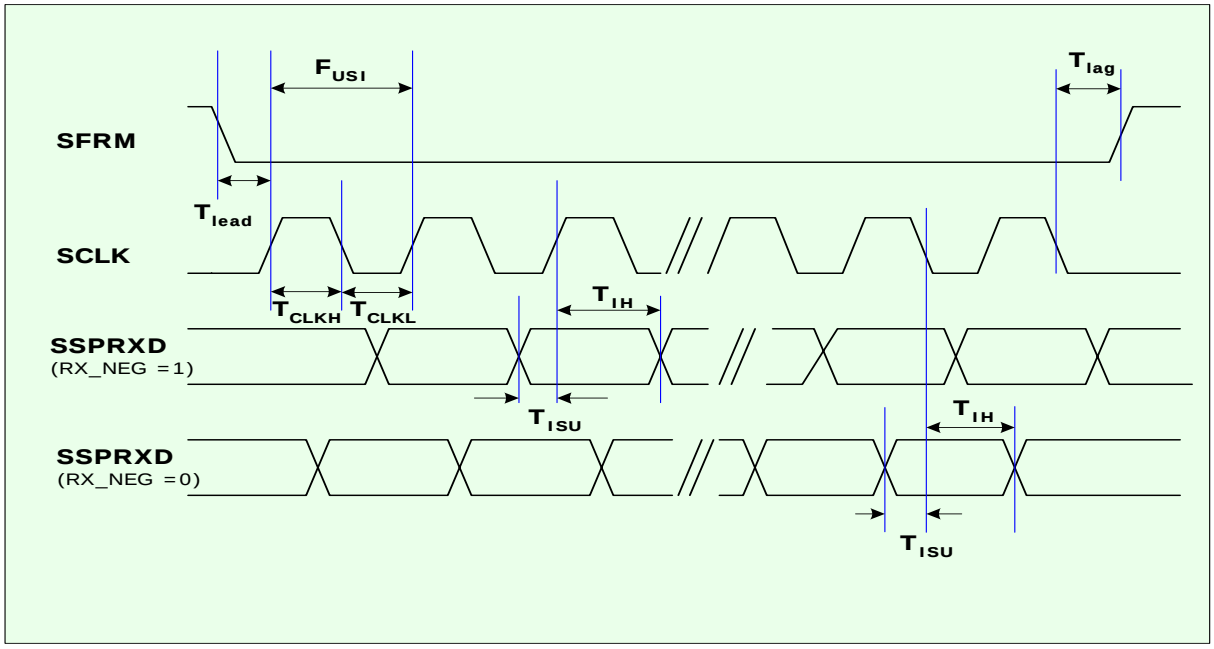




SYMBOL	DESCRIPTION	MIN	MAX	UNIT
T_{HIGH}	I²C Clock high time	1	-	us
T_{LOW}	I²C clock low time	1	-	us
$T_{hd:STA}$	Start condition hold time	1	-	us
$T_{SU:DAT}$	Receive data setup time	0.1	-	us
	Transmit data output delay	-	0.5	us
$T_{HD:DAT}$	Receive data hold time	1	-	us
	Transmit data hold time	0	0.9	us
$T_{SU:DAT2}$	SDA setup time (before STOP condition)	0.5	-	us
$T_{SU:STO}$	Stop condition setup time	1	-	us
$T_{SU:STA}$	Restart condition setup time	1.5	-	us

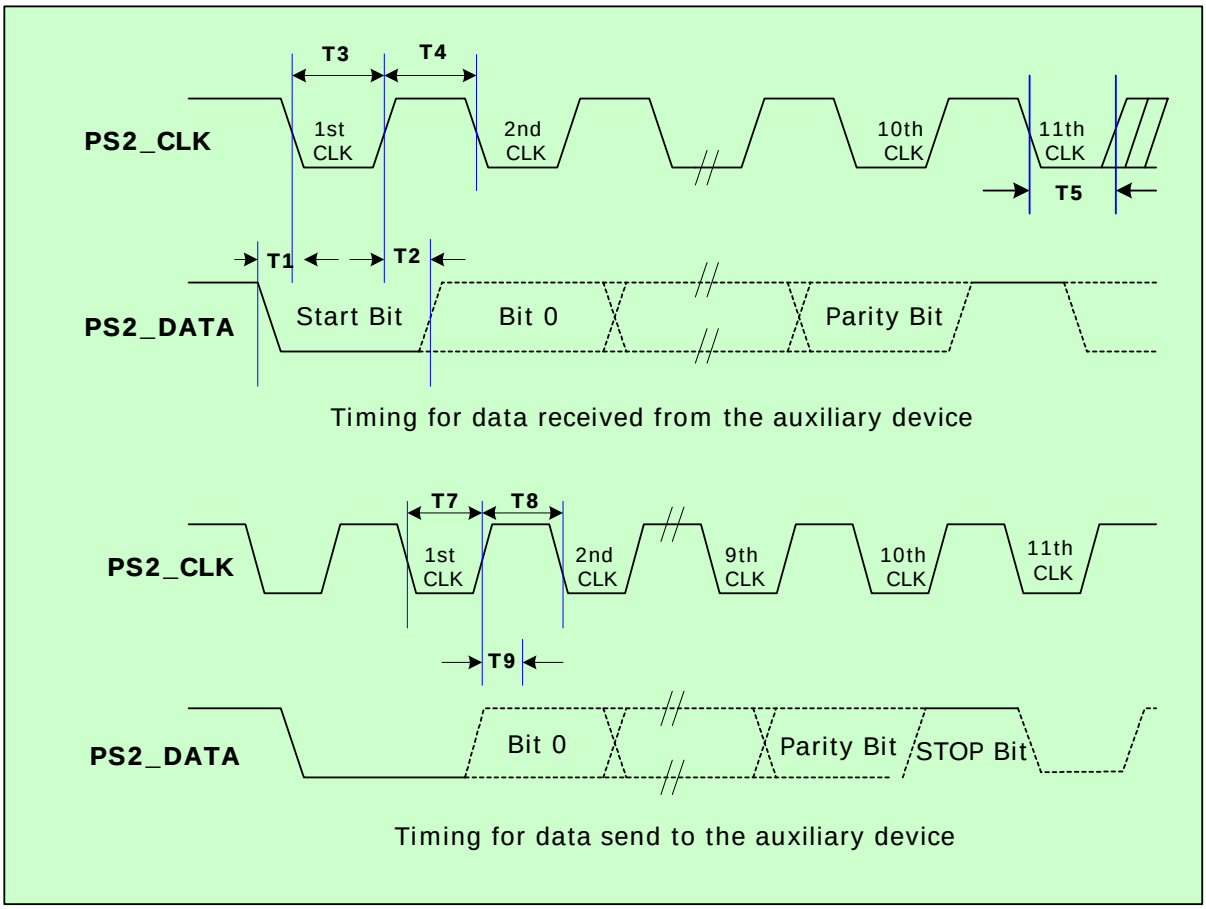
7.3.7 USI Interface AC Characteristics





SYMBOL	DESCRIPTION	MIN	MAX	UNIT
F_{USI}	USI clock frequency	-	20	MHz
T_{CLKH}	USI clock high time	12.5	-	ns
T_{CLKL}	USI clock low time	-	-	ns
T_{ISU}	Data input setup time	-	14	ns
T_{IH}	Data input hold time	0	-	ns
T_{lead}	USI enable lead time	12.5	-	ns
T_{lag}	USI enable lag time	12.5	-	ns
T_{OD}	USI output data valid time	-	30	ns

7.3.8 PS2 Interface AC Characteristics



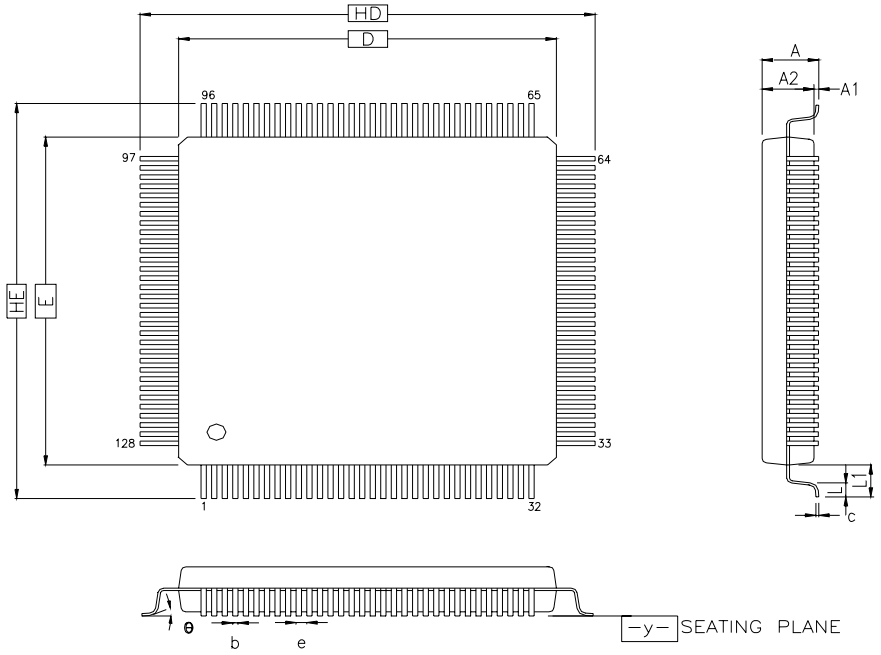
SYMBOL	DESCRIPTION	MIN.	MAX.	UNIT
T1	Time from DATA transition to falling edge of CLK	5	25	us
T2	Time form rising edge of CLK to DATA transition	5	T4-5	us
T3	Duration of CLK inactive	30	50	us
T4	Duration of clock active	30	50	us
T5	Time to auxiliary device inhibit after clock 11 to ensure the auxiliary device does not start another transmission	0	50	us
T7	Duration of CLK inactive	30	50	us
T8	Duration of CLK active	30	50	us
T9	Time to fom inactive to active CLK transition, used to time when the auxiliary device samples DATA	30	50	us

8. ORDERING INFORMATION

PART NUMBER	NAME	PACKAGE DESCRIPTION
W90N745CD	LQFP128	128 Leads, body 14 x 14 x 1.4 mm
W90N745CDG	LQFP128	128 Leads, body 14 x 14 x 1.4 mm, Lead free package

9. PACKAGE SPECIFICATIONS

128L LQFP (14X14X1.4 mm footprint 2.0mm)



COTROL DIMENSIONS ARE IN MILLIMETERS.

SYMBOL	MILLIMETER			INCH		
	MIN.	NOM.	MAX.	MIN.	NOM.	MAX.
A	—	—	1.60	—	—	0.063
A1	0.05	—	0.15	0.002	—	0.006
A2	1.35	1.40	1.45	0.053	0.055	0.057
HD	16.00 BSC.			0.630 BSC.		
D	14.00 BSC.			0.551 BSC.		
HE	16.00 BSC.			0.630 BSC.		
E	14.00 BSC.			0.551 BSC.		
b	0.13	0.16	0.23	0.005	0.006	0.009
e	0.40 BSC.			0.016 BSC.		
θ	0°	3.5°	7°	0°	3.5°	7°
c	0.09	—	0.20	0.004	—	0.008
L	0.45	0.60	0.75	0.018	0.024	0.030
L_1	1.00 REF			0.039 REF		
y	–	–	0.1	–	–	0.004

10. APPENDIX A: W90N745 REGISTERS MAPPING TABLE

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

System Manager Control Registers Map

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
PDID	0xFFF0_0000	R	Product Identifier Register	0xX090.0710
ARBCON	0xFFF0_0004	R/W	Arbitration Control Register	0x0000_0000
PLLCON	0xFFF0_0008	R/W	PLL Control Register	0x0000_2F01
CLKSEL	0xFFF0_000C	R/W	Clock Select Register	0x1FFF_3FX8
PLLCON1	0xFFF0_0010	R/W	PLL Control Register 2	0x0001_0000
I²SCKCON	0xFFF0_0014	R/W	Audio I²S Clock Control Register	0x0000_0000
IRQWAKECON	0xFFF0_0020	R/W	IRQ Wakeup Control register	0x0000_0000
IRQWAKEFLAG	0xFFFF_0024	R/W	IRQ wakeup Flag Register	0x0000_0000
PMCON	0xFFF0_0028	R/W	Power Manager Control Register	0x0000_0000
USBTxrCON	0xFFF0_0030	R/W	USB Transceiver Control Register	0x0000_0000

External Bus Interface Control Registers Map

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
EBICON	0xFFF0_1000	R/W	EBI control register	0x0001_0000
ROMCON	0xFFF0_1004	R/W	ROM/FLASH control register	0x0000_0XFC
SDCONF0	0xFFF0_1008	R/W	SDRAM bank 0 configuration register	0x0000_0800
SDCONF1	0xFFF0_100C	R/W	SDRAM bank 1 configuration register	0x0000_0800
SDTIME0	0xFFF0_1010	R/W	SDRAM bank 0 timing control register	0x0000_0000
SDTIME1	0xFFF0_1014	R/W	SDRAM bank 1 timing control register	0x0000_0000
EXT0CON	0xFFF0_1018	R/W	External I/O 0 control register	0x0000_0000
EXT1CON	0xFFF0_101C	R/W	External I/O 1 control register	0x0000_0000
EXT2CON	0xFFF0_1020	R/W	External I/O 2 control register	0x0000_0000
EXT3CON	0xFFF0_1024	R/W	External I/O 3 control register	0x0000_0000
CKSKEW	0xFFF0_1F00	R/W	Clock skew control register (for testing)	0xXXXX_0038

Cache Control Registers Map

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
CAHCNF	0xFFF0_2000	R/W	Cache configuration register	0x0000_0000
CAHCON	0xFFF0_2004	R/W	Cache control register	0x0000_0000
CAHADR	0xFFF0_2008	R/W	Cache address register	0x0000_0000

EMC Registers Map

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
CAMCMR	0xFFF0_3000	R/W	CAM Command Register	0x0000_0000
CAMEN	0xFFF0_3004	R/W	CAM Enable Register	0x0000_0000
CAM0M	0xFFF0_3008	R/W	CAM0 Most Significant Word Register	0x0000_0000
CAM0L	0xFFF0_300C	R/W	CAM0 Least Significant Word Register	0x0000_0000
CAM1M	0xFFF0_3010	R/W	CAM1 Most Significant Word Register	0x0000_0000
CAM1L	0xFFF0_3014	R/W	CAM1 Least Significant Word Register	0x0000_0000
CAM2M	0xFFF0_3018	R/W	CAM2 Most Significant Word Register	0x0000_0000
CAM2L	0xFFF0_301C	R/W	CAM2 Least Significant Word Register	0x0000_0000
CAM3M	0xFFF0_3020	R/W	CAM3 Most Significant Word Register	0x0000_0000
CAM3L	0xFFF0_3024	R/W	CAM3 Least Significant Word Register	0x0000_0000
CAM4M	0xFFF0_3028	R/W	CAM4 Most Significant Word Register	0x0000_0000
CAM4L	0xFFF0_302C	R/W	CAM4 Least Significant Word Register	0x0000_0000
CAM5M	0xFFF0_3030	R/W	CAM5 Most Significant Word Register	0x0000_0000
CAM5L	0xFFF0_3034	R/W	CAM5 Least Significant Word Register	0x0000_0000
CAM6M	0xFFF0_3038	R/W	CAM6 Most Significant Word Register	0x0000_0000
CAM6L	0xFFF0_303C	R/W	CAM6 Least Significant Word Register	0x0000_0000
CAM7M	0xFFF0_3040	R/W	CAM7 Most Significant Word Register	0x0000_0000
CAM7L	0xFFF0_3044	R/W	CAM7 Least Significant Word Register	0x0000_0000
CAM8M	0xFFF0_3048	R/W	CAM8 Most Significant Word Register	0x0000_0000
CAM8L	0xFFF0_304C	R/W	CAM8 Least Significant Word Register	0x0000_0000
CAM9M	0xFFF0_3050	R/W	CAM9 Most Significant Word Register	0x0000_0000
CAM9L	0xFFF0_3054	R/W	CAM9 Least Significant Word Register	0x0000_0000
CAM10M	0xFFF0_3058	R/W	CAM10 Most Significant Word Register	0x0000_0000
CAM10L	0xFFF0_305C	R/W	CAM10 Least Significant Word Register	0x0000_0000
CAM11M	0xFFF0_3060	R/W	CAM11 Most Significant Word Register	0x0000_0000
CAM11L	0xFFF0_3064	R/W	CAM11 Least Significant Word Register	0x0000_0000

EMC Registers Map, continued

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
CAM12M	0xFFF0_3068	R/W	CAM12 Most Significant Word Register	0x0000_0000
CAM12L	0xFFF0_306C	R/W	CAM12 Least Significant Word Register	0x0000_0000
CAM13M	0xFFF0_3070	R/W	CAM13 Most Significant Word Register	0x0000_0000
CAM13L	0xFFF0_3074	R/W	CAM13 Least Significant Word Register	0x0000_0000
CAM14M	0xFFF0_3078	R/W	CAM14 Most Significant Word Register	0x0000_0000
CAM14L	0xFFF0_307C	R/W	CAM14 Least Significant Word Register	0x0000_0000
CAM15M	0xFFF0_3080	R/W	CAM15 Most Significant Word Register	0x0000_0000
CAM15L	0xFFF0_3084	R/W	CAM15 Least Significant Word Register	0x0000_0000
TXDLSA	0xFFF0_3088	R/W	Transmit Descriptor Link List Start Address Register	0xFFFF_FFFC
RXDLSA	0xFFF0_308C	R/W	Receive Descriptor Link List Start Address Register	0xFFFF_FFFC
MCMDR	0xFFF0_3090	R/W	MAC Command Register	0x0000_0000
MIID	0xFFF0_3094	R/W	MII Management Data Register	0x0000_0000
MIIDA	0xFFF0_3098	R/W	MII Management Control and Address Register	0x0090_0000
FFTCR	0xFFF0_309C	R/W	FIFO Threshold Control Register	0x0000_0101
TSDR	0xFFF0_30A0	W	Transmit Start Demand Register	Undefined
RSDR	0xFFF0_30A4	W	Receive Start Demand Register	Undefined
DMARFC	0xFFF0_30A8	R/W	Maximum Receive Frame Control Register	0x0000_0800
MIEN	0xFFF0_30AC	R/W	MAC Interrupt Enable Register	0x0000_0000
MISTA	0xFFF0_30B0	R/W	MAC Interrupt Status Register	0x0000_0000
MGSTA	0xFFF0_30B4	R/W	MAC General Status Register	0x0000_0000
MPCNT	0xFFF0_30B8	R/W	Missed Packet Count Register	0x0000_7FFF
MRPC	0xFFF0_30BC	R	MAC Receive Pause Count Register	0x0000_0000
MRPCC	0xFFF0_30C0	R	MAC Receive Pause Current Count Register	0x0000_0000
MREPC	0xFFF0_30C4	R	MAC Remote Pause Count Register	0x0000_0000
DMARFS	0xFFF0_30C8	R/W	DMA Receive Frame Status Register	0x0000_0000
CTXDSA	0xFFF0_30CC	R	Current Transmit Descriptor Start Address Register	0x0000_0000
CTXBSA	0xFFF0_30D0	R	Current Transmit Buffer Start Address Register	0x0000_0000
CRXDSA	0xFFF0_30D4	R	Current Receive Descriptor Start Address Register	0x0000_0000
CRXBSA	0xFFF0_30D8	R	Current Receive Buffer Start Address Register	0x0000_0000

EMC Registers Map, continued

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
RXFSM	0xFFF0_3200	R	Receive Finite State Machine Register	0x0081_1101
TXFSM	0xFFF0_3204	R	Transmit Finite State Machine Register	0x0101_1101
FSM0	0xFFF0_3208	R	Finite State Machine Register 0	0x0001_0101
FSM1	0xFFF0_320C	R	Finite State Machine Register 1	0x1100_0100
DCR	0xFFF0_3210	R/W	Debug Configuration Register	0x0000_003F
DMMIR	0xFFF0_3214	R	Debug Mode MAC Information Register	0x0000_0000
BISTR	0xFFF0_3300	R/W	BIST Mode Register	0x0000_0000

GDMA Registers Map

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
GDMA_CTL0	0xFFF0_4000	R/W	Channel 0 Control Register	0x0000_0000
GDMA_SRCB0	0xFFF0_4004	R/W	Channel 0 Source Base Address Register	0x0000_0000
GDMA_DSTB0	0xFFF0_4008	R/W	Channel 0 Destination Base Address Register	0x0000_0000
GDMA_TCNT0	0xFFF0_400C	R/W	Channel 0 Transfer Count Register	0x0000_0000
GDMA_CSRC0	0xFFF0_4010	R	Channel 0 Current Source Address Register	0x0000_0000
GDMA_CDST0	0xFFF0_4014	R	Channel 0 Current Destination Address Register	0x0000_0000
GDMA_CTCNT0	0xFFF0_4018	R	Channel 0 Current Transfer Count Register	0x0000_0000
GDMA_CTL1	0xFFF0_4020	R/W	Channel 1 Control Register	0x0000_0000
GDMA_SRCB1	0xFFF0_4024	R/W	Channel 1 Source Base Address Register	0x0000_0000
GDMA_DSTB1	0xFFF0_4028	R/W	Channel 1 Destination Base Address Register	0x0000_0000
GDMA_TCNT1	0xFFF0_402C	R/W	Channel 1 Transfer Count Register	0x0000_0000
GDMA_CSRC1	0xFFF0_4030	R	Channel 1 Current Source Address Register	0x0000_0000
GDMA_CDST1	0xFFF0_4034	R	Channel 1 Current Destination Address Register	0x0000_0000
GDMA_CTCNT1	0xFFF0_4038	R	Channel 1 Current Transfer Count Register	0x0000_0000

USB Host Controller Registers Map

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
OpenHCI Registers				
HcRevision	0xFFF0_5000	R	Host Controller Revision Register	0x0000_0010
HcControl	0xFFF0_5004	R/W	Host Controller Control Register	0x0000_0000
HcCommandStatus	0xFFF0_5008	R/W	Host Controller Command Status Register	0x0000_0000
HcInterruptStatus	0xFFF0_500C	R/W	Host Controller Interrupt Status Register	0x0000_0000
HcInterruptEnbale	0xFFF0_5010	R/W	Host Controller Interrupt Enable Register	0x0000_0000
HcInterruptDisbale	0xFFF0_5014	R/W	Host Controller Interrupt Disable Register	0x0000_0000
HcHCCA	0xFFF0_5018	R/W	Host Controller Communication Area Register	0x0000_0000
HcPeriodCurrentED	0xFFF0_501C	R/W	Host Controller Period Current ED Register	0x0000_0000
HcControlHeadED	0xFFF0_5020	R/W	Host Controller Control Head ED Register	0x0000_0000
HcControlCurrentED	0xFFF0_5024	R/W	Host Controller Control Current ED Register	0x0000_0000
HcBulkHeadEd	0xFFF0_5028	R/W	Host Controller Bulk Head ED Register	0x0000_0000
HcBulkCurrentED	0xFFF0_502C	R/W	Host Controller Bulk Current ED Register	0x0000_0000
HcDoneHeadED	0xFFF0_5030	R/W	Host Controller Done Head Register	0x0000_0000
HcFmInterval	0xFFF0_5034	R/W	Host Controller Frame Interval Register	0x0000_2EDF
HcFrameRemaining	0xFFF0_5038	R	Host Controller Frame Remaining Register	0x0000_0000
HcFmNumber	0xFFF0_503C	R	Host Controller Frame Number Register	0x0000_0000
HcPeriodicStart	0xFFF0_5040	R/W	Host Controller Periodic Start Register	0x0000_0000
HcLSThreshold	0xFFF0_5044	R/W	Host Controller Low Speed Threshold Register	0x0000_0628
HcRhDescriptorA	0xFFF0_5048	R/W	Host Controller Root Hub Descriptor A Register	0x0100_0002
HcRhDescriptorB	0xFFF0_504C	R/W	Host Controller Root Hub Descriptor B Register	0x0000_0000
HcRhStatus	0xFFF0_5050	R/W	Host Controller Root Hub Status Register	0x0000_0000
HcRhPortStatus [1]	0xFFF0_5054	R/W	Host Controller Root Hub Port Status [1]	0x0000_0000
HcRhPortStatus [2]	0xFFF0_5058	R/W	Host Controller Root Hub Port Status [2]	0x0000_0000
USB Configuration Registers				
TestModeEnable	0xFFF0_5200	R/W	USB Test Mode Enable Register	0x0XXX_XXXX
OperationalModeEnable	0xFFF0_5204	R/W	USB Operational Mode Enable Register	0x0000_0000

USB Device Register Map

REGISTER	OFFSET	R/W	DESCRIPTION	RESET VALUE
USB_CTL	0xFFF0_6000	R/W	USB control register	0x0000_0000
VCMD	0xFFF0_6004	R/W	USB class or vendor command register	0x0000_0000
USB_IE	0xFFF0_6008	R/W	USB interrupt enable register	0x0000_0000
USB_IS	0xFFF0_600C	R	USB interrupt status register	0x0000_0000
USB_IC	0xFFF0_6010	R/W	USB interrupt status clear register	0x0000_0000
USB_IFSTR	0xFFF0_6014	R/W	USB interface and string register	0x0000_0000
USB_ODATA0	0xFFF0_6018	R	USB control transfer-out port 0 register	0x0000_0000
USB_ODATA1	0xFFF0_601C	R	USB control transfer-out port 1 register	0x0000_0000
USB_ODATA2	0xFFF0_6020	R	USB control transfer-out port 2 register	0x0000_0000
USB_ODATA3	0xFFF0_6024	R	USB control transfer-out port 3 register	0x0000_0000
USB_IDATA0	0xFFF0_6028	R/W	USB transfer-in data port0 register	0x0000_0000
USB_IDATA1	0xFFF0_602C	R/W	USB control transfer-in data port 1	0x0000_0000
USB_IDATA2	0xFFF0_6030	R/W	USB control transfer-in data port 2	0x0000_0000
USB_IDATA3	0xFFF0_6034	R/W	USB control transfer-in data port 3	0x0000_0000
USB_SIE	0xFFF0_6038	R	USB SIE status Register	0x0000_0000
USB_ENG	0xFFF0_603C	R/W	USB Engine Register	0x0000_0000
USB_CTLS	0xFFF0_6040	R	USB control transfer status register	0x0000_0000
USB_CONFD	0xFFF0_6044	R/W	USB Configured Value register	0x0000_0000
EPA_INFO	0xFFF0_6048	R/W	USB endpoint A information register	0x0000_0000

EPA_CTL	0xFFF0_604C	R/W	USB endpoint A control register	0x0000_0000
EPA_IE	0xFFF0_6050	R/W	USB endpoint A Interrupt Enable register	0x0000_0000
EPA_IC	0xFFF0_6054	W	USB endpoint A interrupt clear register	0x0000_0000
EPA_IS	0xFFF0_6058	R	USB endpoint A interrupt status register	0x0000_0000
EPA_ADDR	0xFFF0_605C	R/W	USB endpoint A address register	0x0000_0000
EPA_LENTH	0xFFF0_6060	R/W	USB endpoint A transfer length register	0x0000_0000
EPB_INFO	0xFFF0_6064	R/W	USB endpoint B information register	0x0000_0000
EPB_CTL	0xFFF0_6068	R/W	USB endpoint B control register	0x0000_0000
EPB_IE	0xFFF0_606C	R/W	USB endpoint B Interrupt Enable register	0x0000_0000
EPB_IC	0xFFF0_6070	W	USB endpoint B interrupt clear register	0x0000_0000
EPB_IS	0xFFF0_6074	R	USB endpoint B interrupt status register	0x0000_0000
EPB_ADDR	0xFFF0_6078	R/W	USB endpoint B address register	0x0000_0000

USB Device Register Map, continued

REGISTER	OFFSET	R/W	DESCRIPTION	RESET VALUE
EPB_LENTH	0xFFF0_607C	R/W	USB endpoint B transfer length register	0x0000_0000
EPC_INFO	0xFFF0_6080	R/W	USB endpoint C information register	0x0000_0000
EPC_CTL	0xFFF0_6084	R/W	USB endpoint C control register	0x0000_0000
EPC_IE	0xFFF0_6 088	R/W	USB endpoint C Interrupt Enable register	0x0000_0000
EPC_IC	0xFFF0_608C	W	USB endpoint C interrupt clear register	0x0000_0000
EPC_IS	0xFFF0_6090	R	USB endpoint C interrupt status register	0x0000_0000
EPC_ADDR	0xFFF0_6094	R/W	USB endpoint C address register	0x0000_0000
EPC_LENTH	0xFFF0_6098	R/W	USB endpoint C transfer length register	0x0000_0000
EPA_XFER	0xFFF0_609C	R/W	USB endpoint A remain transfer length register	0x0000_0000
EPA_PKT	0xFFF0_60A0	R/W	USB endpoint A remain packet length register	0x0000_0000
EPB_XFER	0xFFF0_60A4	R/W	USB endpoint B remain transfer length register	0x0000_0000
EPB_PKT	0xFFF0_60A8	R/W	USB endpoint B remain packet length register	0x0000_0000
EPC_XFER	0xFFF0_60AC	R/W	USB endpoint C remain transfer length register	0x0000_0000
EPC_PKT	0xFFF0_60B0	R/W	USB endpoint C remain packet length register	0x0000_0000

Audio Control Register Map

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
ACTL_CON	0xFFF0_9000	R/W	Audio controller control register	0x0000_0000
ACTL_RESET	0xFFF0_9004	R/W	Sub block reset control register	0x0000_0000
ACTL_RDSTB	0xFFF0_9008	R/W	DMA destination base address register for record	0x0000_0000
ACTL_RDST_LENGTH	0xFFF0_900C	R/W	DMA destination length register for record	0x0000_0000
ACTL_RDSTC	0xFFF0_9010	R	DMA destination current address register for record	0x0000_0000
ACTL_RSR	0xFFF0_9014	R/W	Record status register	0x0000_0000
ACTL_PDSTB	0xFFF0_9018	R/W	DMA destination base address register for play	0x0000_0000
ACTL_PDST_LENGTH	0xFFF0_901C	R/W	DMA destination length register for play	0x0000_0000

Audio Control Register Map, continued

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
ACTL_PDSTC	0xFFF0_9020	R	DMA destination current address register for play	0x0000_0000
ACTL_PSR	0xFFF0_9024	R/W	Play status register	0x0000_0004
ACTL_I²SCON	0xFFF0_9028	R/W	I²S control register	0x0000_0000
ACTL_ACCON	0xFFF0_902C	R/W	AC-link control register	0x0000_0000
ACTL_ACOS0	0xFFF0_9030	R/W	AC-link out slot 0	0x0000_0000
ACTL_ACOS1	0xFFF0_9034	R/W	AC-link out slot 1	0x0000_0080
ACTL_ACOS2	0xFFF0_9038	R/W	AC-link out slot 2	0x0000_0000
ACTL_ACIS0	0xFFF0_903C	R	AC-link in slot 0	0x0000_0000
ACTL_ACIS1	0xFFF0_9040	R	AC-link in slot 1	0x0000_0000
ACTL_ACIS2	0xFFF0_9044	R	AC-link in slot 2	0x0000_0000

Cache Controller Test Registers Map

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
CTEST0	0xFFF6_0000	R/W	Cache test register 0	0x0000_0000
CTEST1	0xFFF6_0004	R	Cache test register 1	0x0000_0000

UART0 Control Registers Map

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
UART0_RBR	0xFFF8_0000	R	Receive Buffer Register (DLAB = 0)	Undefined
UART0_THR	0xFFF8_0000	W	Transmit Holding Register (DLAB = 0)	Undefined
UART0_IER	0xFFF8_0004	R/W	Interrupt Enable Register (DLAB = 0)	0x0000_0000
UART0_DLL	0xFFF8_0000	R/W	Divisor Latch Register (LS) (DLAB = 1)	0x0000_0000
UART0_DLM	0xFFF8_0004	R/W	Divisor Latch Register (MS) (DLAB = 1)	0x0000_0000
UART0_IIR	0xFFF8_0008	R	Interrupt Identification Register	0x8181_8181
UART0_FCR	0xFFF8_0008	W	FIFO Control Register	Undefined
UART0_LCR	0xFFF8_000C	R/W	Line Control Register	0x0000_0000
UART0_LSR	0xFFF8_0014	R	Line Status Register	0x6060_6060
UART0_TOR	0xFFF8_001C	R	Time Out Register	0x0000_0000

High Speed UART1 Control Registers Map

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
UART1_RBR	0xFFF8_0100	R	Receive Buffer Register (DLAB = 0)	Undefined
UART1_THR	0xFFF8_0100	W	Transmit Holding Register (DLAB = 0)	Undefined
UART1_IER	0xFFF8_0104	R/W	Interrupt Enable Register (DLAB = 0)	0x0000_0000
UART1_DLL	0xFFF8_0100	R/W	Divisor Latch Register (LS) (DLAB = 1)	0x0000_0000
UART1_DLM	0xFFF8_0104	R/W	Divisor Latch Register (MS) (DLAB = 1)	0x0000_0000
UART1_IIR	0xFFF8_0108	R	Interrupt Identification Register	0x8181_8181
UART1_FCR	0xFFF8_0108	W	FIFO Control Register	Undefined
UART1_LCR	0xFFF8_010C	R/W	Line Control Register	0x0000_0000
UART1_MCR	0xFFF8_0110	R/W	Modem Control Register	0x0000_0000
UART1_LSR	0xFFF8_0114	R	Line Status Register	0x6060.6060
UART1_MSR	0xFFF8_0118	R	MODEM Status Register	0x0000_0000
UART1_TOR	0xFFF8_011C	R	Time Out Register	0x0000_0000
UART1_UBCR	0xFFF8_0120	R/W	UART1 Bluetooth Control Register	0x0000_0000

UART2 Control Register Map

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
UART2_RBR	0xFFF8_0200	R	Receive Buffer Register (DLAB = 0)	Undefined
UART2_THR	0xFFF8_0200	W	Transmit Holding Register (DLAB = 0)	Undefined
UART2_IER	0xFFF8_0204	R/W	Interrupt Enable Register (DLAB = 0)	0x0000_0000
UART2_DLL	0xFFF8_0200	R/W	Divisor Latch Register (LS) (DLAB = 1)	0x0000_0000
UART2_DLM	0xFFF8_0204	R/W	Divisor Latch Register (MS) (DLAB = 1)	0x0000_0000
UART2_IIR	0xFFF8_0208	R	Interrupt Identification Register	0x8181_8181
UART2_FCR	0xFFF8_0208	W	FIFO Control Register	Undefined
UART2_LCR	0xFFF8_020C	R/W	Line Control Register	0x0000_0000
UART2_MCR	0xFFF8_0210	R/W	Modem Control Register	0x0000_0000
UART2_LSR	0xFFF8_0214	R	Line Status Register	0x6060_6060
UART2_MSR	0xFFF8_0218	R	MODEM Status Register	0x0000_0000
UART2_TOR	0xFFF8_021C	R	Time Out Register	0x0000_0000
UART2_IRCR	0xFFF8_0220	R/W	IrDA Control Register	0x0000_0040

UART3 Control Register Map

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
UART3_RBR	0xFFF8_0300	R	Receive Buffer Register (DLAB = 0)	Undefined
UART3_THR	0xFFF8_0300	W	Transmit Holding Register (DLAB = 0)	Undefined
UART3_IER	0xFFF8_0304	R/W	Interrupt Enable Register (DLAB = 0)	0x0000_0000
UART3_DLL	0xFFF8_0300	R/W	Divisor Latch Register (LS) (DLAB = 1)	0x0000_0000
UART3_DLM	0xFFF8_0304	R/W	Divisor Latch Register (MS) (DLAB = 1)	0x0000_0000
UART3_IIR	0xFFF8_0308	R	Interrupt Identification Register	0x8181_8181
UART3_FCR	0xFFF8_0308	W	FIFO Control Register	Undefined
UART3_LCR	0xFFF8_030C	R/W	Line Control Register	0x0000_0000
UART3_MCR	0xFFF8_0310	R/W	Modem Control Register	0x0000_0000
UART3_LSR	0xFFF8_0314	R	Line Status Register	0x6060_6060
UART3_MSR	0xFFF8_0318	R	MODEM Status Register	0x0000_0000
UART3_TOR	0xFFF8_031C	R	Time Out Register	0x0000_0000

Timer Control Registers Map

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
TCR0	0xFFF8_1000	R/W	Timer Control Register 0	0x0000_0005
TCR1	0xFFF8_1004	R/W	Timer Control Register 1	0x0000_0005
TICR0	0xFFF8_1008	R/W	Timer Initial Control Register 0	0x0000_00FF
TICR1	0xFFF8_100C	R/W	Timer Initial Control Register 1	0x0000_00FF
TDR0	0xFFF8_1010	R	Timer Data Register 0	0x0000_0000
TDR1	0xFFF8_1014	R	Timer Data Register 1	0x0000_0000
TISR	0xFFF8_1018	R/C	Timer Interrupt Status Register	0x0000_0000
WTCR	0xFFF8_101C	R/W	Watchdog Timer Control Register	0x0000_0000

AIC Control Registers Map

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
AIC_SCR1	0xFFF8_2004	R/W	Source Control Register 1	0x0000_0047
AIC_SCR2	0xFFF8_2008	R/W	Source Control Register 2	0x0000_0047
AIC_SCR3	0xFFF8_200C	R/W	Source Control Register 3	0x0000_0047
AIC_SCR4	0xFFF8_2010	R/W	Source Control Register 4	0x0000_0047
AIC_SCR5	0xFFF8_2014	R/W	Source Control Register 5	0x0000_0047
AIC_SCR6	0xFFF8_2018	R/W	Source Control Register 6	0x0000_0047
AIC_SCR7	0xFFF8_201C	R/W	Source Control Register 7	0x0000_0047
AIC_SCR8	0xFFF8_2020	R/W	Source Control Register 8	0x0000_0047
AIC_SCR9	0xFFF8_2024	R/W	Source Control Register 9	0x0000_0047
AIC_SCR10	0xFFF8_2028	R/W	Source Control Register 10	0x0000_0047
AIC_SCR11	0xFFF8_202C	R/W	Source Control Register 11	0x0000_0047
AIC_SCR12	0xFFF8_2030	R/W	Source Control Register 12	0x0000_0047
AIC_SCR13	0xFFF8_2034	R/W	Source Control Register 13	0x0000_0047
AIC_SCR14	0xFFF8_2038	R/W	Source Control Register 14	0x0000_0047
AIC_SCR15	0xFFF8_203C	R/W	Source Control Register 15	0x0000_0047
AIC_SCR16	0xFFF8_2040	R/W	Source Control Register 16	0x0000_0000
AIC_SCR17	0xFFF8_2044	R/W	Source Control Register 17	0x0000_0000
AIC_SCR18	0xFFF8_2048	R/W	Source Control Register 18	0x0000_0000
AIC_SCR19	0xFFF8_204C	R/W	Source Control Register 19	0x0000_0047
AIC_SCR20	0xFFF8_2050	R/W	Source Control Register 20	0x0000_0047
AIC_SCR21	0xFFF8_2054	R/W	Source Control Register 21	0x0000_0047
AIC_SCR22	0xFFF8_2058	R/W	Source Control Register 22	0x0000_0047
AIC_SCR23	0xFFF8_205C	R/W	Source Control Register 23	0x0000_0047
AIC_SCR24	0xFFF8_2060	R/W	Source Control Register 24	0x0000_0047
AIC_SCR25	0xFFF8_2064	R/W	Source Control Register 25	0x0000_0047
AIC_SCR26	0xFFF8_2068	R/W	Source Control Register 26	0x0000_0047
AIC_SCR27	0xFFF8_206C	R/W	Source Control Register 27	0x0000_0047
AIC_SCR28	0xFFF8_2070	R/W	Source Control Register 28	0x0000_0047
AIC_SCR29	0xFFF8_2074	R/W	Source Control Register 29	0x0000_0047
AIC_SCR30	0xFFF8_2078	R/W	Source Control Register 30	0x0000_0047
AIC_SCR31	0xFFF8_207C	R/W	Source Control Register 31	0x0000_0047

AIC Control Registers Map, continued

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
AIC_IRSR	0xFFF8_2100	R	Interrupt Raw Status Register	0x0000_0000
AIC_IASR	0xFFF8_2104	R	Interrupt Active Status Register	0x0000_0000
AIC_ISR	0xFFF8_2108	R	Interrupt Status Register	0x0000_0000
AIC_IPER	0xFFF8_210C	R	Interrupt Priority Encoding Register	0x0000_0000
AIC_ISNR	0xFFF8_2110	R	Interrupt Source Number Register	0x0000_0000
AIC_IMR	0xFFF8_2114	R	Interrupt Mask Register	0x0000_0000
AIC_OISR	0xFFF8_2118	R	Output Interrupt Status Register	0x0000_0000
AIC_MECR	0xFFF8_2120	W	Mask Enable Command Register	Undefined
AIC_MDCR	0xFFF8_2124	W	Mask Disable Command Register	Undefined
AIC_SSCR	0xFFF8_2128	W	Source Set Command Register	Undefined
AIC_SCCR	0xFFF8_212C	W	Source Clear Command Register	Undefined
AIC_EOSCR	0xFFF8_2130	W	End of Service Command Register	Undefined
AIC_TEST	0xFFF8_2200	W	ICE/Debug mode Register	Undefined

GPIO Control Register Map

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
GPIO_CFG0	0xFFF8_3000	R/W	GPIO Port0 Configuration Register	0x0000_0000
GPIO_DIR0	0xFFF8_3004	R/W	GPIO Port0 Direction Control Register	0x0000_0000
GPIO_DATAOUT0	0xFFF8_3008	R/W	GPIO Port0 Data Output Register	0x0000_0000
GPIO_DATAIN0	0xFFF8_300C	R	GPIO Port0 Data Input Register	0xXXXX_XXXX
GPIO_CFG1	0xFFF8_3010	R/W	GPIO port1 configuration register	0x0000_0000
GPIO_DIR1	0xFFF8_3014	R/W	GPIO port1 direction control register	0x0000_0000
GPIO_DATAOUT1	0xFFF8_3018	R/W	GPIO port1 data output register	0x0000_0000
GPIO_DATAIN1	0xFFF8_301C	R	GPIO port1 data input register	0xXXXX_XXXX
GPIO_CFG2	0xFFF8_3020	R/W	GPIO Port2 Configuration Register	0x0000_0000
GPIO_DIR2	0xFFF8_3024	R/W	GPIO Port2 Direction Control Register	0x0000_0000
GPIO_DATAOUT2	0xFFF8_3028	R/W	GPIO Port2 Data Output Register	0x0000_0000
GPIO_DATAIN2	0xFFF8_302C	R	GPIO Port2 Data Input Register	0xXXXX_XXXX
GPIO_CFG4	0xFFF8_3040	R/W	GPIO Port4 Configuration Register	0x0015_5555
GPIO_DIR4	0xFFF8_3044	R/W	GPIO Port4 Direction Control Register	0x0000_0000
GPIO_DATAOUT4	0xFFF8_3048	R/W	GPIO Port4 Data Output Register	0x0000_0000
GPIO_DATAIN4	0xFFF8_304C	R	GPIO Port4 Data Input Register	0xXXXX_XXXX

GPIO Control Register Map, continued

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
GPIO_CFG5	0xFFF8_3050	R/W	GPIO Port5 Configuration Register	0x0000_0000
GPIO_DIR5	0xFFF8_3054	R/W	GPIO Port5 Direction Control Register	0x0000_0000
GPIO_DATAOUT5	0xFFF8_3058	R/W	GPIO Port5 Data Output Register	0x0000_0000
GPIO_DATAIN5	0xFFF8_305C	R	GPIO Port5 Data Input Register	0xXXXX_XXXX
GPIO_DBNCECON	0xFFF8_3070	R/W	GPIO Input Debounce Control Register	0x0000_0000
GPIO_XICFG	0xFFF8_3074	R/W	Extend Interrupt Configure Register	0xXXXX_XXX0
GPIO_XISTATUS	0xFFF8_3078	R/W	Extend Interrupt Status Register	0xXXXX_XXX0

I²C Register Map

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
I²C Interface 0				
I²C_CSR0	0xFFF8_6000	R/W	I²C0 Control and Status Register	0x0000_0000
I²C_DIVIDER0	0xFFF8_6004	R/W	I²C0 Clock Prescale Register	0x0000_0000
I²C_CMDR0	0xFFF8_6008	R/W	I²C0 Command Register	0x0000_0000
I²C_SWR0	0xFFF8_600C	R/W	I²C0 Software Mode Control Register	0x0000_003F
I²C_RxR0	0xFFF8_6010	R	I²C0 Data Receive Register	0x0000_0000
I²C_TxR0	0xFFF8_6014	R/W	I²C0 Data Transmit Register	0x0000_0000
I²C Interface 1				
I²C_CSR1	0xFFF8_6000	R/W	I²C1 Control and Status Register	0x0000_0000
I²C_DIVIDER1	0xFFF8_6004	R/W	I²C1 Clock Prescale Register	0x0000_0000
I²C_CMDR1	0xFFF8_6008	R/W	I²C1 Command Register	0x0000_0000
I²C_SWR1	0xFFF8_600C	R/W	I²C1 Software Mode Control Register	0x0000_003F
I²C_RxR1	0xFFF8_6010	R	I²C1 Data Receive Register	0x0000_0000
I²C_TxR1	0xFFF8_6014	R/W	I²C1 Data Transmit Register	0x0000_0000

USI Register Map

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
USI_CNTRL	0xFFF8_6200	R/W	Control and Status Register	0x0000_0004
USI_DIVIDER	0xFFF8_6204	R/W	Clock Divider Register	0x0000_0000
USI_SSR	0xFFF8_6208	R/W	Slave Select Register	0x0000_0000
Reserved	0xFFF8_620C	N/A	Reserved	N/A
USI_Rx0	0xFFF8_6210	R	Data Receive Register 0	0x0000_0000
USI_Rx1	0xFFF8_6214	R	Data Receive Register 1	0x0000_0000
USI_Rx2	0xFFF8_6218	R	Data Receive Register 2	0x0000_0000
USI_Rx3	0xFFF8_621C	R	Data Receive Register 3	0x0000_0000
USI_Tx0	0xFFF8_6210	W	Data Transmit Register 0	0x0000_0000
USI_Tx1	0xFFF8_6214	W	Data Transmit Register 1	0x0000_0000
USI_Tx2	0xFFF8_6218	W	Data Transmit Register 2	0x0000_0000
USI_Tx3	0xFFF8_621C	W	Data Transmit Register 3	0x0000_0000

PWM Control Registers Map

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
PWM_PPR	0xFFF8_7000	R/W	PWM Prescaler Register	0x0000_0000
PWM_CSR	0xFFF8_7004	R/W	PWM Clock Select Register	0x0000_0000
PWM_PCR	0xFFF8_7008	R/W	PWM Control Register	0x0000_0000
PWM_CNR0	0xFFF8_700C	R/W	PWM Counter Register 0	0x0000_0000
PWM_CMR0	0xFFF8_7010	R/W	PWM Comparator Register 0	0x0000_0000
PWM_PDR0	0xFFF8_7014	R	PWM Data Register 0	0x0000_0000
PWM_CNR1	0xFFF8_7018	R/W	PWM Counter Register 1	0x0000_0000
PWM_CMR1	0xFFF8_701C	R/W	PWM Comparator Register 1	0x0000_0000
PWM_PDR1	0xFFF8_7020	R	PWM Data Register 1	0x0000_0000
PWM_CNR2	0xFFF8_7024	R/W	PWM Counter Register 2	0x0000_0000
PWM_CMR2	0xFFF8_7028	R/W	PWM Comparator 2	0x0000_0000
PWM_PDR2	0xFFF8_702C	R	PWM Data Register 2	0x0000_0000
PWM_CNR3	0xFFF8_7030	R/W	PWM Counter Register 3	0x0000_0000
PWM_CMR3	0xFFF8_7034	R/W	PWM Comparator Register 3	0x0000_0000
PWM_PDR3	0xFFF8_7038	R	PWM Data Register 3	0x0000_0000
PWM_PIER	0xFFF8_703C	R/W	PWM Interrupt Enable Register	0x0000_0000
PWM_PIIR	0xFFF8_7040	R/C	PWM Interrupt Indication Register	0x0000_0000

KPI Control Register Map

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
KPICONF	0xFFF8_8000	R/W	Keypad controller configuration Register	0x0000_0000
KPI3KCONF	0xFFF8_8004	R/W	Keypad controller 3-keys configuration register	0x0000_0000
KPILPCONF	0xFFF8_8008	R/W	Keypad controller low power configuration register	0x0000_0000
KPISTATUS	0xFFF8_800C	R/O	Keypad controller status register	0x0000_0000

PS2 Control Register Map

REGISTER	ADDRESS	R/W/C	DESCRIPTION	RESET VALUE
PS2CMD	0xFFF8_9000	R/W	PS2 Host Controller Command Register	0x0000_0000
PS2STS	0xFFF8_9004	R/W	PS2 Host Controller Status Register	0x0000_0000
PS2SCANCODE	0xFFF8_9008	RO	PS2 Host Controller RX Scan Code Register	0x0000_0000
PS2ASCII	0xFFF8_900C	RO	PS2 Host Controller RX ASCII Code Register	0x0000_0000

Important Notice

Winbond products are not designed, intended, authorized or warranted for use as components in systems or equipment intended for surgical implantation, atomic energy control instruments, airplane or spaceship instruments, transportation instruments, traffic signal instruments, combustion control instruments, or for other applications intended to support or sustain life. Further more, Winbond products are not intended for applications wherein failure of Winbond products could result or lead to a situation wherein personal injury, death or severe property or environmental damage could occur.

Winbond customers using or selling these products for use in such applications do so at their own risk and agree to fully indemnify Winbond for any damages resulting from such improper use or sales.