# HMS30C7110

## Multipurpose Network Processor

**(ARM Based 32-Bit Microprocessor)**

# Datasheet

**Version 1.5**

**MagnaChip Semiconductor Ltd.**

## Copyright. 2002 Magnachip Semiconductor Inc.

Magnachip Semiconductor Inc.

#1, Hyangjeong-dong, Heungduk-gu, Cheongju-si, Chungcheonbuk-do, Republic of Korea

Homepage: http://www.Magnachip.com

Technical Support Homepage: will be construction as soon as possible

**H.Q. of Magnachip Semiconductor Inc.**                                              **Marketing SiteSales in Korea**

Telephone: 82-(0)43-270-4070          Telephone: 82-(0)43-270-4085          Telephone: 82-(0)2-3459-3738

Facsimile: 82-(0)43-270-4099          Facsimile: 82-(0)43-270-4099          Facsimile: 82-(0)2-3459-3945

**World Wide Sales Network**

**U.S.A.**                                        **Taiwan**                                      **Hong Kong**

Telephone: 1-408-232-8757             Telephone: 886-(0)2-2500-8357        Telephone: 852-2971-1640

Facsimile: 1-408-232-8135             Facsimile: 886-(0)2-2509-8977        Facsimile: 852-2971-1622

*HMS30C7110 Datasheet, ver1.5*

28 July. 03

**Magnachip Semiconductor, Corp. may make changes to specification and product description at any time without notice.**

## Revision History

| Rev. 0.1 | 2002-07-31 | First draft |
|---|---|---|
| Rev. 1.01 | 2002-10-31 | |
| Rev. 1.1 | 2002-12-31 | |
| Rev. 1.2 | 2003-03-17 | Correction on I/O and Register Map |
| Rev. 1.3 | 2003-04-13 | Correction on Register Description |
| Rev. 1.4 | 2003-07-18 | Correction on Register Description |
| Rev. 1.5 | 2003-07-28 | Add I/O Pin Description and add Detail address<br>Add PLL register Map |

# Contents

# Figures

# Tables

# 1. Product Overview

The HMS30C7110® is a low-cost and high performance network processor for telecommunication equipments such as IP sharer, Wireless Local Area Network Access Point (WLAN-AP), Managed switch/Bridge/Hub, routers, and other devices that provide high-speed data networking. The HMS30C7110® includes an ARM7TDMI processor with unified cache, and common network functions such as dual Ethernet MAC, PCMCIA, and other common peripherals.

# 1.1. Summary of HMS30C7110® features

**System**

■ 32-bit ARM7 Processor Core with cache

■ High Speed System Bus & Peripheral Bus

■ Supports both Big-Endian / Little-Endian modes

■ JTAG-based debug solution

**Cache**

■ 4K Byte unified (data + instruction)

■ 4-way set associative

■ Write back policy

■ Read miss fill up

■ 8 word write buffer

**Ethernet MAC**

■ Dual MAC with 10/100Base T

■ MII, RMII and 7-wire interface

■ 8 Ethernet MAC addresses

■ Internal loop-back mode

■ Allows programming of PHY active level signal

■ Pad insertion for minimum frame size (64 bytes)

■ Programmable maximum frame length (up to 2000 bytes)

**DMA Controller**

■ 2-channel programmable DMA

■ UART DMA support

**UART**

■ Independent 2-channel Tx/Rx

■ Full duplex mode

■ Hardware flow control for one channel

**Version 1.5**

■ 16 byte internal Rx/Tx FIFOs to reduce data latency

■ Even, odd, forced one, and forced zero parity mode

**Interrupt Controller**

■ Handles up to 32 interrupt sources

■ Supports hardware priority function

■ Programmable IRQ/FIQ mode

**SPI**

■ Operates as master mode

■ Configurable address and data width

**Timer**

■ Three channels of 32-bit timer

■ Selectable channel clock speed

**GPIO**

■ Up to 13 GPI, 9 GPO (General Purpose Input Output) ports

■ Supports external interrupt inputs

**Memory Controller**

■ Supports up to three 4MB banks (Flash/ROM) and one 128MB bank (SDRAM)

■ Supports all major SDRAM/ROM/Flash memories

■ Supports 32-bit data width for SDRAM

■ Supports 8-bit, 16-bit, and 32-bit data width for Flash/ROM

**PCMCIA Controller**

■ Supports 16-bit PC Card host interface

■ Supports CardBus

Note: 16-bit PCMCIA is designated as "16-bit PC Card" and 32-bit PCMCIA as "CardBus". PC card is the card suitable for insertion in PCMCIA slot.

**Physical Characteristics**

- ■ Operating Voltage: Core 2.5V, I/O 3.3V

- ■ Technology: 0.25um CMOS

- ■ Operating Temperature: 0 ~ 70° C

- ■ Operating Frequency: 70 MHz

- ■ Package Type: 208 PQFP

# 1.2. Block Diagram
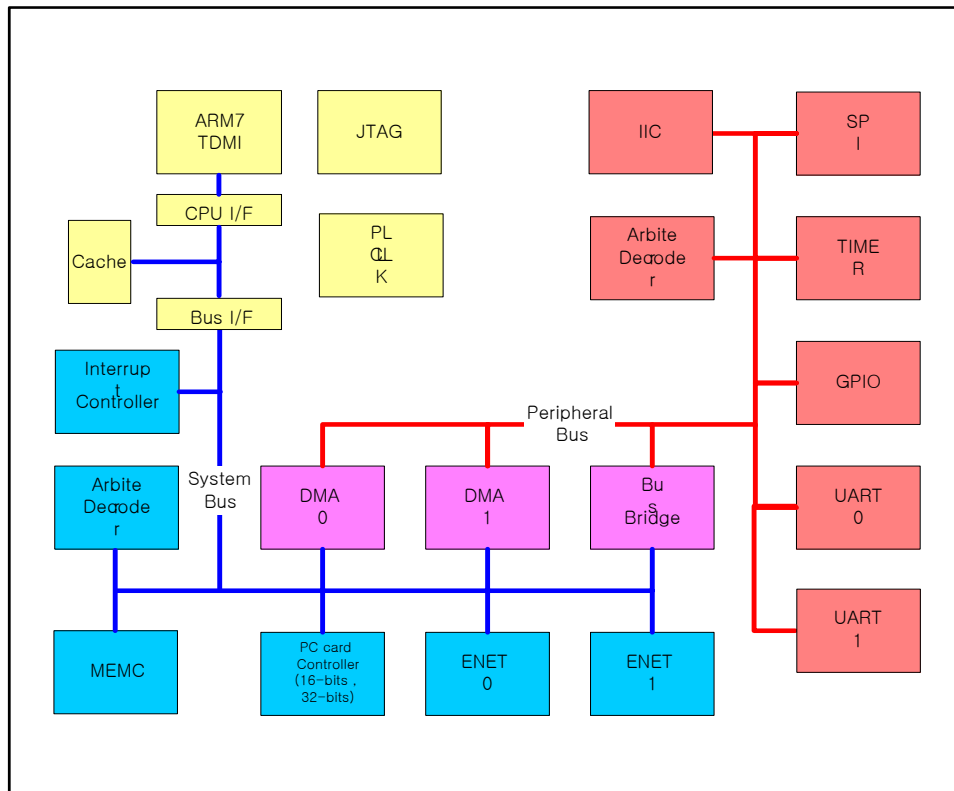


**Figure 1.1 HMS30C7110® Block Diagram**

# 1.3. Pin Assignments

**Table 1.1 PQFP Pin List**

| Pin # | Pin Name | I O | Pad Type | Description |
|---|---|---|---|---|
| 1 | MODE | I | PICD | Operation Mode |
| 2 | TESTSE | I | PICD | Test Scan Enable. Refer to the description for Pin 1 |
| 3 | nRESET | I | PICS | System Reset, active low |
| 4 | VSSP | P | | VSS for IO |
| 5 | MCLK | I | PICS | SDRAM Memory Feedback Clock |
| 6 | SCLK | O | POC8A | SDRAM clock |
| 7 | VSSI | P | | VSS for core |
| 8 | ADDR21/Endian | BD | PBCD8A | Address Bus/Endian at Reset period. . Pulled down internally over 50 Kohm. 0: Little Endian (Default) 1: Big Endian |
| 9 | ADDR20/Boot32 | BD | PBCD8A | Address Bus/Boot32 at Reset period. Pulled down internally over 50 Kohm. 0: 16-bit booting(Default) 1: 32-bit booting |
| 10 | ADDR19/Boot16-8 | BU | PBCU8A | Address Bus/Boot16-8 at Reset. Pulled up internally over 50 Kohm. 0: 8-bit booting 1: 16-bit booting(Default) |
| 11 | ADDR18/Bypass | BU | PBCU8A | Address Bus/Bypass Internal Power-On-Reset Circuit. Pulled up internally over 50 Kohm. The active output width is over 1.5 Sec when 10MHz main clock is used. 0: Use internal Power-On-Reset ciruit 1: Bypass internal Power-On-Reset circuit (Default) |

For Pin 1 (MODE), Operation Mode:

| MODE | TESTSE | Contents |
|---|---|---|
| 0 | 0 | Normal Operation |
| 0 | 1 | NANDTREE/BIST/PLL Test |
| 1 | 0 | Parallel Capture for ATPG |
| 1 | 1 | Scan Shift for ATPG |

| | | | | Provide external 10 Kohm pull-down to use internal RESET |
|---|---|---|---|---|
| 12 | ADDR17 | O | POC8A | Address Bus |
| 13 | VCCI | P | | VDD for core |
| 14 | ADDR16 | O | POC8A | Address Bus |
| 15 | ADDR15 | O | POC8A | Address Bus |
| 16 | ADDR14/BA1 | O | POC8A | Address Bus/SDRAM Bank Address 1 |
| 17 | ADDR13/BA0 | O | POC8A | Address Bus/SDRAM Bank Address 0 |
| 18 | ADDR12 | O | POC8A | Address Bus |
| 19 | ADDR11 | O | POC8A | Address Bus |
| 20 | VCCP | P | | VDD for IO |
| 21 | ADDR10 | O | POC8A | Address Bus |
| 22 | ADDR9 | O | POC8A | Address Bus |
| 23 | ADDR8 | O | POC8A | Address Bus |
| 24 | ADDR7 | O | POC8A | Address Bus |
| 25 | ADDR6 | O | POC8A | Address Bus |
| 26 | VSSP | P | | VSS for IO |
| 27 | ADDR5 | O | POC8A | Address Bus |
| 28 | ADDR4 | O | POC8A | Address Bus |
| 29 | ADDR3 | O | POC8A | Address Bus |
| 30 | ADDR2 | O | POC8A | Address Bus |
| 31 | ADDR1 | O | POC8A | Address Bus |
| 32 | ADDR0 | O | POC8A | Address Bus |
| 33 | VSSI | P | | VSS for core |
| 34 | DATA31 | BU | PBCU8A | Data Bus |
| 35 | DATA30 | BU | PBCU8A | Data Bus |
| 36 | DATA29 | BU | PBCU8A | Data Bus |
| 37 | DATA28 | BU | PBCU8A | Data Bus |
| 38 | DATA27 | BU | PBCU8A | Data Bus |
| 39 | DATA26 | BU | PBCU8A | Data Bus |
| 40 | VSSP | P | | VSS for IO |
| 41 | DATA25 | BU | PBCU8A | Data Bus |
| 42 | DATA24 | BU | PBCU8A | Data Bus |

**Version 1.5**

| 43 | DATA23 | BU | PBCU8A | Data Bus |
|----|--------|----|--------|----------|
| 44 | DATA22 | BU | PBCU8A | Data Bus |
| 45 | DATA21 | BU | PBCU8A | Data Bus |
| 46 | DATA20 | BU | PBCU8A | Data Bus |
| 47 | VCCP | P | | VDD for IO |
| 48 | DATA19 | BU | PBCU8A | Data Bus |
| 49 | DATA18 | BU | PBCU8A | Data Bus |
| 50 | DATA17 | BU | PBCU8A | Data Bus |
| 51 | DATA16 | BU | PBCU8A | Data Bus |
| 52 | DATA15 | BU | PBCU8A | Data Bus |
| 53 | VSSP | P | | VSS for IO |
| 54 | DATA14 | BU | PBCU8A | Data Bus |
| 55 | DATA13 | BU | PBCU8A | Data Bus |
| 56 | DATA12 | BU | PBCU8A | Data Bus |
| 57 | DATA11 | BU | PBCU8A | Data Bus |
| 58 | DATA10 | BU | PBCU8A | Data Bus |
| 59 | DATA9 | BU | PBCU8A | Data Bus |
| 60 | VCCI | P | | VDD for core |
| 61 | DATA8 | BU | PBCU8A | Data Bus |
| 62 | DATA7 | BU | PBCU8A | Data Bus |
| 63 | DATA6 | BU | PBCU8A | Data Bus |
| 64 | DATA5 | BU | PBCU8A | Data Bus |
| 65 | DATA4 | BU | PBCU8A | Data Bus |
| 66 | DATA3 | BU | PBCU8A | Data Bus |
| 67 | VSSI | P | | VSS for core |
| 68 | DATA2 | BU | PBCU8A | Data Bus |
| 69 | DATA1 | BU | PBCU8A | Data Bus |
| 70 | DATA0 | BU | PBCU8A | Data Bus |
| 71 | nSCS | O | POC8A | SDRAM Chip Select |
| 72 | nRAS | O | POC8A | Row Address Strobe |
| 73 | nCAS | O | POC8A | Column Address Strobe |
| 74 | nSWE | O | POC8A | SDRAM Write Enable |

**Version 1.5**

| 75 | VCCP | P | | VDD for IO |
|---|---|---|---|---|
| 76 | DQM3 | O | POC8A | SDRAM Data Mask 3 |
| 77 | DQM2 | O | POC8A | SDRAM Data Mask 2 |
| 78 | DQM1 | O | POC8A | SDRAM Data Mask 1 |
| 79 | DQM0 | O | POC8A | SDRAM Data Mask 0 |
| 80 | GPI[11] | B | PBC4A | GPIO input bit 11 |
| 81 | GPI[12] | B | PBC4A | GPIO input bit 12 |
| 82 | VCCI | P | | VDD for core |
| 83 | nRCS2 | O | POC4A | ROM/FLASH Chip Select |
| 84 | nRCS1 | O | POC4A | ROM/FLASH Chip Select |
| 85 | nRCS0 | O | POC4A | ROM/FLASH Chip Select |
| 86 | nROE | O | POC8A | ROM/FLASH Output Enable |
| 87 | nRWE | O | POC8A | ROM/FLASH Write Enable |
| 88 | VCCP | P | | VDD for IO |
| 89 | E0_COL/ /GPI[3] | I | PIC | Collision/GPI[3] |
| 90 | E0_CRS/E0_CRSDV | I | PIC | Carrier Sense from the Ethernet PHY |
| 91 | E0_TXD3/ GPO[5] | O | POC4A | Transmit Data from the Ethernet PHY/GPO[5] |
| 92 | E0_TXD2/GPO[4] | O | POC4A | Transmit Data from the Ethernet PHY/GPO[4] |
| 93 | E0_TXD1/ E0_TXD1 | O | POC4A | Transmit Data from the Ethernet PHY/ |
| 94 | E0_TXD0/ E0_TXD0 | O | POC4A | Transmit Data from the Ethernet PHY |
| 95 | E0_TXEN/ E0_TXEN | O | POC4A | Transmit Data Enable to the Ethernet PHY |
| 96 | VSSP | P | | VSS for IO |
| 97 | E0_TXCLK | I | PICS | Transmit Clock from the Ethernet PHY |
| 98 | E0_TXER/ GPO[3] | O | POC4A | Transmit Data Error to the Ethernet PHY/GPO[3] |
| 99 | E0_RXER/ E0_RXER | I | PIC | Receive Data Error from the Ethernet PHY |
| 100 | E0_RXCLK/E0_CLK | I | PICS | Receive Clock from the Ethernet PHY |
| 101 | E0_RXDV/ GPI[6] | I | PIC | Receive Data Valid from the Ethernet PHY/GPI[6] |
| 102 | E0_RXD0 /E0_RXD0 | I | PIC | Receive Data from the Ethernet PHY |
| 103 | E0_RXD1/ E0_RXD1 | I | PIC | Receive Data from the Ethernet PHY |
| 104 | E0_RXD2/ GPI[4] | I | PIC | Receive Data from the Ethernet PHY/GPI[4] |
| 105 | E0_RXD3/ GPI[5] | I | PIC | Receive Data from the Ethernet PHY/GPI[5] |
| 106 | E0_MDC | O | POC4A | Receive Data from the Ethernet PHY |

**Version 1.5**

| 107 | E0_MDIO | B | PBCC4A | Receive Data from the Ethernet PHY |
|---|---|---|---|---|
| 108 | PLLVCC | P | | VDD for PLL |
| 109 | CLKI | I | POSC1 | Main Clock In for crystal (10~20MHz) |
| 110 | CLKO | O | POSC1 | Main Clock Out for crystal (10~20MHz) |
| 111 | PLLVSS | P | | VSS for PLL |
| 112 | E1_COL/ GPI[7] | I | PIC | Collision/GPI[7] |
| 113 | E1_CRS/E1_CRSDV | I | PIC | Carrier Sense from the Ethernet PHY |
| 114 | E1_TXD3/ GPO[8] | O | POC4A | Transmit Data from the Ethernet PHY/GPO[8] |
| 115 | E1_TXD2/ GPO[7] | O | POC4A | Transmit Data from the Ethernet PHY/GPO[7] |
| 116 | E1_TXD1/ E1_TXD1 | O | POC4A | Transmit Data from the Ethernet PHY |
| 117 | E1_TXD0/ E1_TXD0 | O | POC4A | Transmit Data from the Ethernet PHY |
| 118 | E1_TXEN/ E1_TXEN | O | POC4A | Transmit Data Enable to the Ethernet PHY |
| 119 | E1_TXCLK | I | PICS | Transmit Clock from the Ethernet PHY |
| 120 | E1_TXER/ GPO[6] | O | POC4A | Transmit Data Error to the Ethernet PHY/GPO[6] |
| 121 | VSSP | P | | VSS for IO |
| 122 | E1_RXER/ E1_RXER | I | PIC | Receive Data Error from the Ethernet PHY |
| 123 | E1_RXCLK/ E1_CLK | I | PICS | Receive Clock from the Ethernet PHY |
| 124 | E1_RXDV/ GPI[10] | I | PIC | Receive Data Valid from the Ethernet PHY/GPI[10] |
| 125 | E1_RXD0/ E1_RXD0 | I | PIC | Receive Data from the Ethernet PHY |
| 126 | E1_RXD1/ E1_RXD1 | I | PIC | Receive Data from the Ethernet PHY |
| 127 | E1_RXD2/ GPI[8] | I | PIC | Receive Data from the Ethernet PHY/GPI[8] |
| 128 | E1_RXD3/ GPI[9] | I | PIC | Receive Data from the Ethernet PHY/GPI[9] |
| 129 | VCCP | P | | VDD for IO |
| 130 | UTXD | O | POC4A | UART channel Tx Data |
| 131 | URXD | I | PIC | UART channel Rx Data |
| 132 | nCCD_02/CDO | B | PBC4A | CD1 input/ SPI Data Output |
| 133 | nCCD_01/CCLK | B | PBC4A | CD2 input/ SPI Clock Output |
| 134 | CVS2/nCCS | B | PBC4A | VS 1/ SPI Chip Select |
| 135 | CVS1/CDI | B | PBC4A | VS 1/ SPI Data Input |
| 136 | RESET_nCRST | O | POC8A | Card Reset |
| 137 | VSSP | P | | VSS for IO |
| 138 | nWAIT_nCSERR/RxD | BU | PBCU4A | Wait from 16-bit PC Card / RxData for UART 1 |

**Version 1.5**

| 139 | nCE2_CAD10 | BU | PBCU4A | Chip enable 0 for 16-bit PC Card/CardBus Address Data |
| 140 | nCE1_nCBE0 | BU | PBCU4A | Chip enable for 16-bit PC Card/CardBus CBE0 |
| 141 | nWE_nCGNT | BU | PBCU4A | Write enable for 16-bit PC Card /CardBus CGNT |
| 142 | nOE_CAD11 | BU | PBCU4A | Read enable for 16-bit PC Card /CardBus Address Data |
| 143 | WP/nIOIS16_nCCLKRUN | BU | PBCU4A | Write Protect / IO Size 16 (*DMA Request for IO mode 3)/CardBus CCLKRUN |
| 144 | nREG_nCBE3 / DACK | BU | PBCU4A | Attribute Memory Select for 16-bit PC Card / DMA Acknowledge |
| 145 | VCCI | P | | VDD for core |
| 146 | nIORD_CAD13 | BU | PBCU4A | I/O Read for 16-bit PC Card /CardBus Address Data |
| 147 | nIOWR_CAD15 | BU | PBCU4A | I/O Write for 16-bit PC Card /CardBus Address Data |
| 148 | nINPACK_nCREQ | BU | PBCU4A | Input Acknowledge for 16-bit PC Card (*DMA Request for IO mode 2)/CardBus CREQ |
| 149 | nSTSCHG_CSTSCHG | BU | PBCU4A | Status Change from 16-bit PC Card / CTS for UART 1 |
| 150 | READY/nIREQ_nCINT | BU | PBCU4A | Initialization Ready from 16-bit PC Card/CardBus Interrupt Request |
| 151 | D15_CAD8 | BU | PBCU4A | 16-bit PC CARD Data/CardBus Address Data |
| 152 | VSSI | P | | VSS for core |
| 153 | D14 | BU | PBCU4A | 16-bit PC CARD Data |
| 154 | D13_CAD6 | BU | PBCU4A | 16-bit PC CARD Data/CardBus Address Data |
| 155 | D12_CAD4 | BU | PBCU4A | 16-bit PC CARD Data/CardBus Address Data |
| 156 | D11_CAD2 | BU | PBCU4A | 16-bit PC CARD Data/CardBus Address Data |
| 157 | D10_CAD31 | BU | PBCU4A | 16-bit PC CARD Data/CardBus Address Data |
| 158 | D9_CAD30 | BU | PBCU4A | 16-bit PC CARD Data/CardBus Address Data |
| 159 | VCCP | P | | VDD for IO |
| 160 | D8_CAD28 | BU | PBCU4A | 16-bit PC CARD Data/CardBus Address Data |
| 161 | D7_CAD7 | BU | PBCU4A | 16-bit PC CARD Data/CardBus Address Data |
| 162 | D6_CAD5 | BU | PBCU4A | 16-bit PC CARD Data/CardBus Address Data |
| 163 | D5_CAD3 | BU | PBCU4A | 16-bit PC CARD Data/CardBus Address Data |
| 164 | D4_CAD1 | BU | PBCU4A | 16-bit PC CARD Data/CardBus Address Data |
| 165 | D3_CAD0 | BU | PBCU4A | 16-bit PC CARD Data/CardBus Address Data |
| 166 | VSSI | P | | VSS for core |
| 167 | D2 | BU | PBCU4A | 16-bit PC CARD Data |
| 168 | D1_CAD29 | BU | PBCU4A | 16-bit PC CARD Data/CardBus Address Data |

**Version 1.5**

| 169 | D0_CAD27 | BU | PBCU4A | 16-bit PC CARD Data/CardBus Address Data |
| 170 | A25_CAD19/TXD1 | BU | PBCU4A | 16-bit Pc Card Address/CardBus Address Data / TxData for UART 1 |
| 171 | A24_CAD17 | BU | PBCU4A | 16-bit PC Card Address/CardBus Address Data |
| 172 | VSSP | P | | VSS for IO |
| 173 | A23_nCFRAME | BU | PBCU4A | 16-bit PC Card Address/CardBus Frame |
| 174 | A22_nCTRDY | BU | PBCU4A | 16-bit PC Card Address/CardBus TRDY |
| 175 | A21_nCDEVSEL | BU | PBCU4A | 16-bit PC Card Address/CardBus Dev Sel |
| 176 | A20_nCSTOP | BU | PBCU4A | 16-bit PC Card Address/CardBus STOP |
| 177 | A19_nCBLOCK | BU | PBCU4A | 16-bit PC Card Address/CardBus CBLOCK |
| 178 | A18 | B | PBC4A | 16-bit PC Card Address |
| 179 | A17_CAD16 | BU | PBCU4A | 16-bit PC Card Address/CardBus Address Data |
| 180 | VCCP | P | | VDD for IO |
| 181 | A16_CCLK | B | PBC8A | 16-bit PC Card Address/CCLK |
| 182 | A15_nCIRDY | BU | PBCU4A | 16-bit PC Card Address/CIRDY |
| 183 | A14_nCPERR | BU | PBCU4A | 16-bit PC Card Address/CPERR |
| 184 | A13_CPAR | BU | PBCU4A | 16-bit PC Card Address/CPAR |
| 185 | A12_nCCBE2 | BU | PBCU4A | 16-bit PC Card Address/CCBE2 |
| 186 | VCCI | P | | VDD for core |
| 187 | A11_CAD12 | BU | PBCU4A | 16-bit PC Card Address/CardBus Address Data |
| 188 | A10_CAD9 | BU | PBCU4A | 16-bit PC Card Address/CardBus Address Data |
| 189 | A9_CAD14 | BU | PBCU4A | 16-bit PC Card Address/CardBus Address Data |
| 190 | A8_nCCBE1 | BU | PBCU4A | 16-bit PC Card Address/CardBus CCBE1 |
| 191 | A7_CAD18 | BU | PBCU4A | 16-bit PC Card Address/CardBus Address Data |
| 192 | A6_CAD20 | BU | PBCU4A | 16-bit PC Card Address/CardBus Address Data |
| 193 | VSSP | P | | VSS for IO |
| 194 | A5_CAD21 | BU | PBCU4A | 16-bit PC Card Address/CardBus Address Data |
| 195 | A4_CAD22 | BU | PBCU4A | 16-bit PC Card Address/CardBus Address Data |
| 196 | A3_CAD23 | BU | PBCU4A | 16-bit PC Card Address/CardBus Address Data |
| 197 | A2_CAD24 | BU | PBCU4A | 16-bit PC Card Address/CardBus Address Data |
| 198 | A1_CAD25 | BU | PBCU4A | 16-bit PC Card Address/CardBus Address Data |
| 199 | A0_CAD26 | BU | PBCU4A | 16-bit PC Card Address/CardBus Address Data |
| 200 | VCCP | P | | VDD for IO |

**Version 1.5**

| 201 | TMS | ID | PICDD | JTAG Test Mode |
|---|---|---|---|---|
| 202 | TCK | I | PICS | JTAG Clock |
| 203 | TRST | ID | PICSD | JTAG Reset |
| 204 | TDI | IU | PICDU | JTAG Data In |
| 205 | TDO | O | POC4A | JTAG Data Out |
| 206 | GPIO2/Wait | B | PBC4A | GPIO port/Wait Input |
| 207 | GPIO1 | B | PBC4A | GPIO port |
| 208 | GPIO0 | B | PBC4A | GPIO port |

Note: IO classification (weak pull-up and weak pull-down is pull-up/down with over

50 Kohm.)

B: Bidirectional,        I: Input,        O:Output,        BU:Bidirectional with weak pull-up

IU: Input with weak pull-up,        OU: Output with weak pull-up,

ID: Input with weak pull-down,        BD: Bidirectional with weak pull-down

**Version 1.5**

# 1.4. Package Pin Diagram (PQ208)

HMS30C7110

PQ208

XXXX
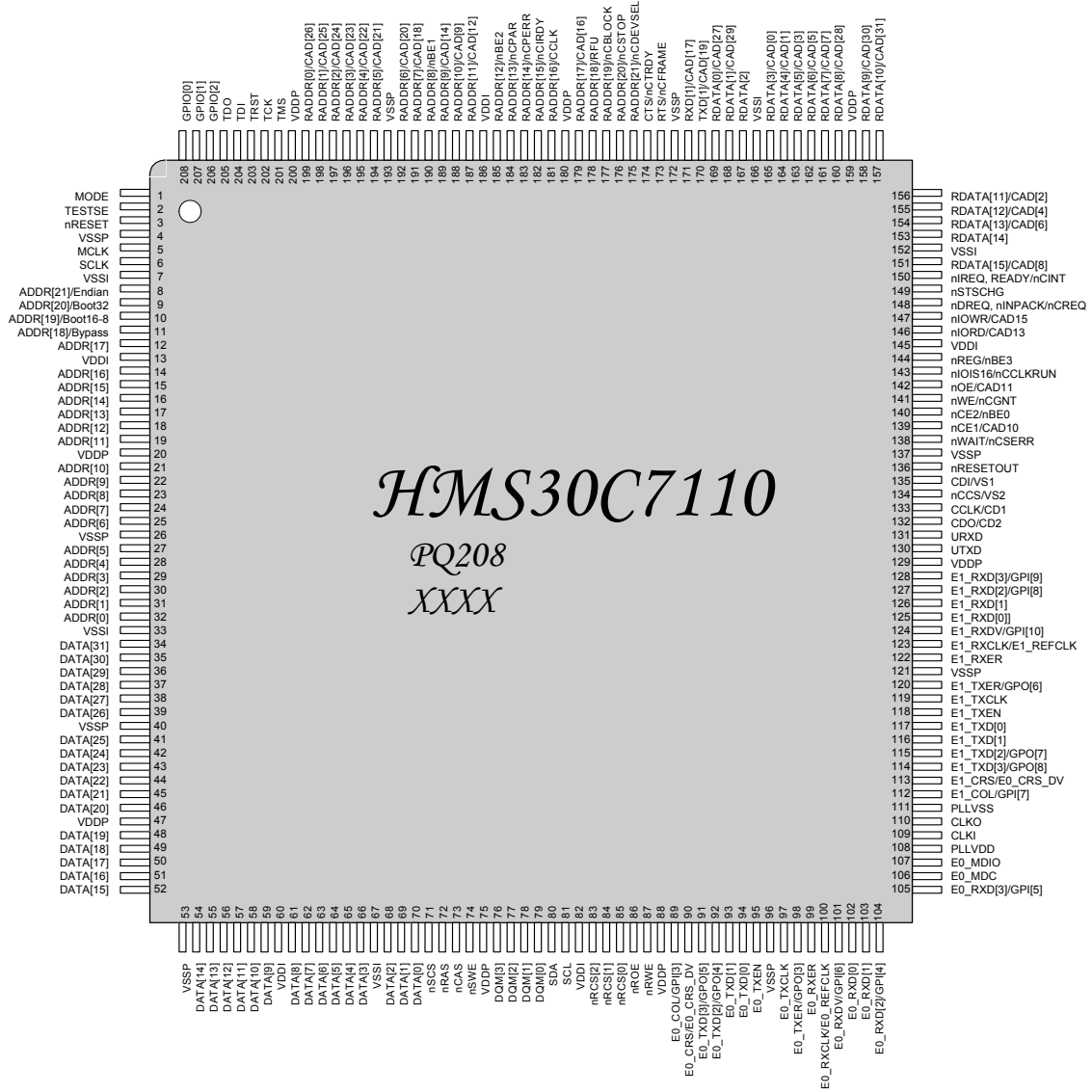
**Figure 1.2 HMS30C7110® 208-Pin PQFP Assignment (top view)**

# 1.5. Pin Description

**Table 1.2 Pin Description**

| Block | Port | Direction | Description |
|---|---|---|---|
| System Configuration (5) | MODE | I | Operation Mode |
| | TESTSE | I | Test Scan Enable |
| | nRESET | I | System Reset, active low |
| | CLKI | I | Main Clock In for crystal (10~20MHz) |
| | CLKO | O | Main Clock Out for crystal (10~20MHz) |
| Ethernet MAC (53) | MTXCLK | I | Transmit Nibble or Symbol Clock from the PHY |
| | MTXD[3:0] | O | Transmit Data Nibble |
| | MTXEN | O | Transmit Enable |
| | MTXERR | O | Transmit Coding Error |
| | MRXCLK | I | Receive Nibble or Symbol Clock from the PHY |
| | MRXDV | I | Receive Data Valid from the PHY |
| | MRXD[3:0] | I | Receive Data Nibble |
| | MRXERR | I | Receive Error from the PHY |
| | MCOLL | I | Collision Detected from the PHY |
| | MCRS | I | Carrier Sense from the PHY |
| | MDC | I | Management Data Clock |
| | MDIO | I/O | Management Data Input/Output |
| | E0_COL | I | Collision from the Ethernet PHY0 |
| | E0_CRS | I | Carrier Sense from the Ethernet PHY0 |
| | E0_TXD[3-0] | O | Transmit Data from the Ethernet PHY0 |
| | E0_TXEN | O | Transmit Data Enable to the Ethernet PHY0 |
| | E0_TXCLK | I | Transmit Clock from the Ethernet PHY0 |
| | E0_TXER | O | Transmit Data Error to the Ethernet PHY0 |
| | E0_RXER | I | Receive Data Error from the Ethernet PHY0 |
| | E0_RXCLK | I | Receive Clock from the Ethernet PHY0 |
| | E0_RXDV | I | Receive Data Valid from the Ethernet PHY0 |
| | E0_RXD[3-0] | I | Receive Data from the Ethernet PHY0 |

**Version 1.5**

| | | | |
|---|---|---|---|
| | E0_MDC | O | Receive Data from the Ethernet PHY0 |
| | E0_MDIO | B | Receive Data from the Ethernet PHY0 |
| | E1_COL | I | Collision from the Ethernet PHY1 |
| | E1_CRS | I | Carrier Sense from the Ethernet PHY |
| | E1_TXD[3-0] | O | Transmit Data from the Ethernet PHY1 |
| | E1_TXEN | O | Transmit Data Enable to the Ethernet PHY1 |
| | E1_TXCLK | I | Transmit Clock from the Ethernet PHY1 |
| | E1_TXER | O | Transmit Data Error to the Ethernet PHY1 |
| | E1_RXER | I | Receive Data Error from the Ethernet PHY1 |
| | E1_RXCLK | I | Receive Clock from the Ethernet PHY1 |
| | E1_RXDV | I | Receive Data Valid from the Ethernet PHY1 |
| | E1_RXD[3-0] | I | Receive Data from the Ethernet PHY1 |
| SDRAM Controller (64) | MCLK | I | SDRAM Memory Feedback Clock |
| | SCLK | O | SDRAM clock |
| | nSCS | O | SDRAM Chip Select |
| | nRAS | O | Row Address Strobe |
| | nCAS | O | Column Address Strobe |
| | NSWE | O | SDRAM Write Enable |
| | DQM3 | O | SDRAM Data Mask 3 |
| | DQM2 | O | SDRAM Data Mask 2 |
| | DQM1 | O | SDRAM Data Mask 1 |
| | DQM0 | O | SDRAM Data Mask 0 |
| | ADDR[21-0] | B | Address Bus |
| | DATA[31-0] | B | Data Bus |
| Flash/ROM Controller (59) | nRCS[2-0] | O | ROM/FLASH Chip Select |
| | nROE | O | ROM/FLASH Output Enable |
| | nRWE | O | ROM/FLASH Write Enable |
| | ADDR[21-0] | B | Address Bus |
| | DATA[31-0] | B | Data Bus |
| PCMCIA Controller (53) | nWAIT_nCSERR | I | Wait from 16-bit PC Card / CardBus CSERR |
| | nCE2_CAD10 | O | Chip enable 0 for 16-bit PC Card/CardBus Address Data |

**Version 1.5**

| | nCE1_nCBE0 | I | Chip enable 1 for 16-bit PC Card/CardBus CBE0 |
|---|---|---|---|
| | nWE_nCGNT | O | Write enable/CardBus CGNT |
| | nOE_CAD11 | B | Read enable/CardBus Address Data |
| | WP/nIOIS16_nCCLKRUN | I | Write Protect / IO Size 16 (*DMA Request for IO mode 3) |
| | nREG_nCBE3 / DACK | O | Attribute Memory Select / DMA Acknowledge |
| | nIORD_CAD13 | B | I/O Read for 16-bit PC Card/CardBus Address Data |
| | nIOWR_CAD15 | B | I/O Write for 16-bit PC Card/CardBus Address Data |
| | nINPACK_nCREQ | I | Input Acknowledge (*DMA Request for IO mode 2) |
| | nSTSCHG_CSTSCHG | I | Status Change from 16-bit PC Card / CTS for UART 1 |
| | READY/nIREQ_nCINT | I | Initialization Ready from 16-bit PC Card / CardBus Interrupt Request |
| | D15_CAD8 | B | 16-bit PC CARD Data/CardBus Address Data |
| | D14 | B | 16-bit PC CARD Data |
| | D13_CAD6 | B | 16-bit PC CARD Data/CardBus Address Data |
| | D12_CAD4 | B | 16-bit PC CARD Data/CardBus Address Data |
| | D11_CAD2 | B | 16-bit PC CARD Data/CardBus Address Data |
| | D10_CAD31 | B | 16-bit PC CARD Data/CardBus Address Data |
| | D9_CAD30 | B | 16-bit PC CARD Data/CardBus Address Data |
| | D8_CAD28 | B | 16-bit PC CARD Data/CardBus Address Data |
| | D7_CAD7 | B | 16-bit PC CARD Data/CardBus Address Data |
| | D6_CAD5 | B | 16-bit PC CARD Data/CardBus Address Data |
| | D5_CAD3 | B | 16-bit PC CARD Data/CardBus Address Data |
| | D4_CAD1 | B | 16-bit PC CARD Data/CardBus Address Data |
| | D3_CAD0 | B | 16-bit PC CARD Data/CardBus Address Data |
| | D2 | B | 16-bit PC CARD Data |
| | D1_CAD29 | B | 16-bit PC CARD Data/CardBus Address Data |
| | D0_CAD27 | B | 16-bit PC CARD Data/CardBus Address Data |
| | A25_CAD19/TXD1 | B | 16-bit PC Card Address/CardBus Address Data / TxData for UART 1 |
| | A24_CAD17 | B | 16-bit PC Card Address/CardBus Address Data |
| | A23_nCFRAME | B | 16-bit PC Card Address/CardBus Frame |

**Version 1.5**
28

| | A22_nCTRDY | B | 16-bit PC Card Address/CardBus TRDY |
|---|---|---|---|
| | A21_nCDEVSEL | B | 16-bit PC Card Address/CardBus Dev Sel |
| | A20_nCSTOP | B | 16-bit PC Card Address/CardBus STOP |
| | A19_nCBLOCK | B | 16-bit PC Card Address/CardBus CBLOCK |
| | A18 | B | 16-bit PC Card Address |
| | A17_CAD16 | B | 16-bit PC Card Address/CardBus Address Data |
| | A16_CCLK | B | 16-bit PC Card Address/ CardBus CCLK |
| | A15_nCIRDY | B | 16-bit PC Card Address/ CardBus CIRDY |
| | A14_nCPERR | B | 16-bit PC Card Address/ CardBus CPERR |
| | A13_CPAR | B | 16-bit PC Card Address/ CardBus CPAR |
| | A12_nCCBE2 | B | 16-bit PC Card Address/ CardBus CCBE2 |
| | A11_CAD12 | B | 16-bit PC Card Address/CardBus Address Data |
| | A10_CAD9 | B | 16-bit PC Card Address/CardBus Address Data |
| | A9_CAD14 | B | 16-bit PC Card Address/CardBus Address Data |
| | A8_nCCBE1 | B | 16-bit PC Card Address/ CardBus CCBE1 |
| | A7_CAD18 | B | 16-bit PC Card Address/CardBus Address Data |
| | A6_CAD20 | B | 16-bit PC Card Address/CardBus Address Data |
| | A5_CAD21 | B | 16-bit PC Card Address/CardBus Address Data |
| | A4_CAD22 | B | 16-bit PC Card Address/CardBus Address Data |
| | A3_CAD23 | B | 16-bit PC Card Address/CardBus Address Data |
| | A2_CAD24 | B | 16-bit PC Card Address/CardBus Address Data |
| | A1_CAD25 | B | 16-bit PC Card Address/CardBus Address Data |
| | A0_CAD26 | B | 16-bit PC Card Address/CardBus Address Data |
| UART (6) | UTXD | O | Tx Data for UART channel0 |
| | URXD | I | Rx Data for UART channel0 |
| | A25_CAD19/TXD1 | O | Tx Data for UART channel1 |
| | nWAIT_nCSERR/RxD | I | Rx Data for UART channel1 |
| | nINPACK_nCREQ | I | CTS for UART 1 |
| | nSTSCHG_CSTSCHG | O | RTS for UART 1 |
| GPIO (21) | GPI12-3 | B | GPIO input |
| | GPO12-3 | B | GPIO output |
| | GPIO2-0 | B | GPIO inout |

| SPI | CCLK | O | SPI Clock Output |
|---|---|---|---|
| (4) | CDO | O | SPI Data Output |
| | CDI | O | SPI Data Input |
| | nCCS | I | SPI Chip Select |

# 2. Functional Description

The following internal functions will be covered in this section:

- System Configuration

- ARM7TDMI Core

- Cache

- Clock/Watch-dog Timer

- Memory Controller (FLASH/SRAM/PCMCIA)

- Memory Controller (SDRAM)

- Ethernet

- UART

- TIMER

- GPIO

- SPI

- DMA

- INTC

- PCMCIA Controller

**Version 1.5**

# 2.1. System Configuration

This section describes system memory map and power-up configuration of HMS30C7110®.

## 2.1.1. Power-up Configuration

The following table shows power-up configuration mapping based on mode configuration pins (MODE, TESTSE).

**Table 2.1 Power-up Configuration**

| MODE | TESTSE | Contents |
|------|--------|----------|
| 0 | 0 | Normal Operation |
| 0 | 1 | NANDTREE/BIST/PLL Test |
| 1 | 0 | Parallel Capture for ATPG |
| 1 | 1 | Scan Shift for ATPG |

## 2.1.2. System Memory Map

The system memory map can be one of the following two configurations:
1. Memory Maps
    A. SDRAM_REMAP = 0
    B. SDRAM_REMAP = 1

Two different memory maps can be selected by one of MEMC (SDRAM) registers. Refer to MEMC for details.

Note 1: This address range is repeated in 0x4000_0000 ~ 0x7FFF_FFFF, 0x8000_0000 ~ 0xBFFF_FFFF, and 0xC000_0000 ~ 0xFFFF_FFFF.

Note 2: 0x0000_0000 ~ 0x7FFF_FFFF is cacheable area and 0x8000_0000 ~ 0xFFFF_FFFF is non-

cacheable area.

**Table 2.2 System Memory Map : SDRAM_REMAP = 0**

| Base Address | Size | Function |
|---|---|---|
| 0x3e00_0000 | 32MB | CardBus special cycle |
| 0x3c00_0000 | 32MB | CardBus configuration |
| 0x3800_0000 | 64MB | CardBus I/O |
| 0x3000_0000 | 128MB | CardBus memory |
| 0x2c00_0000 | 64MB | Reserved |
| 0x2800_0000 | 64MB | Attribute memory of PCMCIA |
| 0x2400_0000 | 64MB | Common memory of PCMCIA |
| 0x2000_0000 | 64MB | I/O area of PCMCIA |
| 0x1960_0000 | 106MB | Reserved |
| 0x1950_0000 | 1MB | CACHE registers |
| 0x1948_0000 | 0.5MB | CardBus Bridge registers |
| 0x1940_0000 | 0.5MB | PCMCIA Interface registers |
| 0x1930_0000 | 1MB | INTC registers |
| 0x1920_1000 | 996KB | ENET MAC1 registers |
| 0x1920_0000 | 4KB | ENET MAC0 registers |
| 0x1910_0000 | 1MB | DMA registers |
| 0x1908_0000 | 0.5MB | SDRAMC registers |
| 0x1900_0000 | 0.5MB | ROMC registers |
| 0x1860_0000 | 10MB | Reserved |
| 0x1850_0000 | 1MB | Reserved |
| 0x1840_0000 | 1MB | SPI registers |
| 0x1830_0000 | 1MB | WDT/CLKRST registers |
| 0x1820_0000 | 1MB | GPIO registers |
| 0x1810_0000 | 1MB | TIMER registers |
| 0x1808_0000 | 0.5MB | UART1 registers |
| 0x1800_0000 | 0.5MB | UART0 registers |
| 0x1000_0000 | 128MB | SDRAM memory |
| 0x0800_0000 | 128MB | Reserved |
| 0x0200_0000 | 96MB | Reserved |
| 0x0100_0000 | 16MB | Flash/IO/SRAM bank-1 |
| 0x0000_0000 | 16MB | Flash/IO/SRAM bank-0 |

**Version 1.5**

**Table 2.3 System Memory Map : SDRAM_REMAP = 1**

| Base Address | Size | Function |
| --- | --- | --- |
| 0x3e00_0000 | 32MB | CardBus special cycle |
| 0x3c00_0000 | 32MB | CardBus configuration |
| 0x3800_0000 | 64MB | CardBus I/O |
| 0x3000_0000 | 128MB | CardBus memory |
| 0x2c00_0000 | 64MB | Reserved |
| 0x2800_0000 | 64MB | Attribute memory of PCMCIA |
| 0x2400_0000 | 64MB | Common memory of PCMCIA |
| 0x2000_0000 | 64MB | I/O area of PCMCIA |
| 0x1960_0000 | 106MB | Reserved |
| 0x1950_0000 | 1MB | CACHE registers |
| 0x1940_0000 | 1MB | CARD Interface registers |
| 0x1930_0000 | 1MB | INTC registers |
| 0x1920_0000 | 1MB | ENET MAC registers |
| 0x1910_0000 | 1MB | DMA registers |
| 0x1900_0000 | 1MB | MEMC registers |
| 0x1860_0000 | 10MB | Reserved |
| 0x1850_0000 | 1MB | Reserved |
| 0x1840_0000 | 1MB | SPI registers |
| 0x1830_0000 | 1MB | WDT/CLKRST registers |
| 0x1820_0000 | 1MB | GPIO registers |
| 0x1810_0000 | 1MB | TIMER registers |
| 0x1808_0000 | 0.5MB | UART1 registers |
| 0x1800_0000 | 0.5MB | UART0 registers |
| 0x1200_0000 | 96MB | Reserved |
| 0x1100_0000 | 16MB | Flash/IO/SRAM bank-1 |
| 0x1000_0000 | 16MB | Flash/IO/SRAM bank-0 |
| 0x0800_0000 | 128MB | Reserved |
| 0x0000_0000 | 128MB | SDRAM memory |

## 2.1.3. Registers Map

**Table 2.4 Register Map at AMBA Peri-Bus**

| BLOCK | ADDRESS | NAME | R/W | Reset Value |
|---|---|---|---|---|
| UART0 | 1800_0000 | DATA FIFO | R/W | xxxx_xxxx |
| | _0004 | Interrupt enable register | R/W | 0000_0000 |
| | _0008 | Interrupt Identification register | R/W | 0000_0001 |
| | _000c | Line control register | R/W | 0000_0000 |
| | _0010 | Modem control Register | R/W | 0000_0000 |
| | _0014 | Line Status Register | R/W | 0000_0000 |
| | _0018 | Modem Status Register | R/W | 0000_0000 |
| | 0001c – 7fffc | RESERVED | | 0000_0000 |
| UART1 | 1808_0000 | DATA FIFO | R/W | xxxx_xxxx |
| | _0004 | Interrupt enable register | R/W | 0000_0000 |
| | _0008 | Interrupt Identification register | R/W | 0000_0001 |
| | _000c | Line control register | R/W | 0000_0000 |
| | _0010 | Modem control Register | R/W | 0000_0000 |
| | _0014 | Line Status Register | R/W | 0000_0000 |
| | _0018 | Modem Status Register | R/W | 0000_0000 |
| | 8001c – ffffc | RESERVED | | 0000_0000 |
| Timer | 1810_0000 | Input clock selection for 3 channels | R/W | 0000_0000 |
| | _0004 | Channel enable | R/W | 0000_0000 |
| | _0008 | Timer Interval For ch0 | R/W | 0000_0000 |
| | _000c | Timer Interval For ch1 | R/W | 0000_0000 |
| | _0010 | Timer Interval For ch2 | R/W | 0000_0000 |
| | _0014 | Current Timer value For ch0 | R | 0000_0000 |
| | _0018 | Current Timer value For ch1 | R | 0000_0000 |
| | _001c | Current Timer value For ch2 | R | 0000_0000 |
| | _0020 | Interrupt Pending Register | R/W | 0000_0000 |
| | _0024 | Interrupt Enable | R/W | 0000_0000 |
| | 00028 – ffffc | RESERVED | | Interrupt Enable |
| GPIO | 1820_0000 | General purpose output | R/W | 0000_0000 |
| | _0004 | General purpose input | R/W | 0000_0000 |
| | _0008 | GPIO direction (Output enable) | R/W | 0000_0000 |

**Version 1.5**

| | | | | |
|---|---|---|---|---|
| | _000c | Interrupt pending register | R/W | 0000_0000 |
| | _0010 | Interrupt enable register | R/W | 0000_0000 |
| | _0014 | Interrupt mode register | R/W | 0000_0000 |
| | _0018 | Interrupt level register | R/W | 0000_0000 |
| | 0001c – ffffc | RESERVED | | 0000_0000 |
| CLOCKRST | 1830_0000 | PLL control | R/W | 0000_0008 |
| | _0004 | WDT control | R/W | 0000_0000 |
| | _0008 | WDT counter | R/W | 0000_0000 |
| | _000c | PLL Control | R/W | a006_0080 |
| | _0010 | PLL Status | R | 0000_0000 |
| | 00014 – ffffc | RESERVED | | 0011_0000 |
| SPI | 1840_0000 | SPI Control | R/W | 0000_0000 |
| | _0004 | SPI Mode 0 | R/W | 0000_0000 |
| | _0008 | Reserved | R/W | |
| | _000c | Reserved | R/W | |
| | _0010 | Reserved | R/W | |
| | _0014 | Reserved | R/W | |
| | _0018 | SPI Tx Data | W | N/a |
| | _001c | SPI Rx Data (R) / Rx Flush (W) | R/W | 0000_00xx |
| | _0020 | SPI Interrupt Pending Register | R/W | 0000_0000 |
| | _0024 | SPI Interrupt Enable Register | R/W | 0000_0000 |
| | _0028 | SPI Status Register | R | 0000_0001 |
| | 0002c – ffffc | RESERVED | | 0000_0000 |
| IIC | 1850_0000 | IIC Control Register | R/W | 0000_0000 |
| | _0004 | IIC Buffer Register | R/W | Xxxx_xxxx |
| | _0008 | IIC Baud Rate Register | R/W | 0000_0000 |
| | _000c | IIC Data Length (in Byte) | R | 0000_0000 |
| | _0010 | IIC Device Address | R/W | 0000_0000 |
| | 00014 – ffffc | RESERVED | | 0000_0000 |

**Table 2.5 Register Map at AMBA Host-Bus**

| BLOCK | ADDRESS | NAME | R/W | Reset Value |
|---|---|---|---|---|
| ROM Controller | 1900_0000 | ROM Timing Control for Bank 0 | R/W | 0333_00ff |
| | _0004 | ROM Timing Control for Bank 1 | R/W | 03ff_ffff |
| | _0008 | ROM Timing Control for Bank 2 | R/W | 03ff_ffff |
| | 0000c – 7fffc | RESERVED | | 0000_0000 |
| SDRAM Controller | 1908_0000 | SDRAM Configuration | R/W | 0000_003f |
| | _0004 | Reserved | | |
| | _0008 | Reserved | | |
| | _000c | Reserved | | |
| | _0010 | SDRAM timing parameters | R/W | 7777_ff71 |
| | _0014 | Refresh interval value & MCLK timing | R/W | 1303_0000 |
| | _0018 | Initialize start register | R/W | 0000_0000 |
| | _001c | Booting Control Register | R/W | 0000_0000 |
| | 80020 – 7fffc | RESERVED | | 0000_0000 |
| DMA Controller | 1910_0000 | DMA Initial Source Register for channel 0 | R/W | 0000_0000 |
| | _0004 | DMA Initial Destination Register for CH0 | R/W | 0000_0000 |
| | _0008 | DMA Control Register for CH0 | R/W | 0000_0000 |
| | _000c | DMA Status Register for CH0 | R | 0000_0000 |
| | _0010 | DMA Current Source Register for CH0 | R | 0000_0000 |
| | _0014 | DMA Current Destination Register for CH0 | R | 0000_0000 |
| | _0018 | DMA Mask Trigger Register for CH0 | R/W | 0000_0000 |
| | _001c | Reserved | | |
| | _0020 | DMA Initial Source Register for channel 1 | R/W | 0000_0000 |
| | _0024 | DMA Initial Destination Register for CH 1 | R/W | 0000_0000 |
| | _0028 | DMA Control Register for CH 1 | R/W | 0000_0000 |
| | _002c | DMA Status Register for CH 1 | R | 0000_0000 |
| | _0030 | DMA Current Source Register for CH 1 | R | 0000_0000 |
| | _0034 | DMA Current Destination Register for CH 1 | R | 0000_0000 |
| | _0038 | DMA Mask Trigger Register for CH 1 | R/W | 0000_0000 |
| | 0003c – ffffc | RESERVED | | 0000_0000 |

**Version 1.5**

| Enet Mac 0 Controller | 1920_0000 | MAC mode register | R/W | 0002_0630 |
|---|---|---|---|---|
| | _0004 | Interrupt Source register | R/W | 0000_0000 |
| | _0008 | Interrupt Mask register | R/W | 0000_0000 |
| | _000c | Inter-frame gap register | R/W | 0000_0018 |
| | _0010 | Collision control register | R/W | 000f_2010 |
| | _0014 | Transmit buffer base address Low | R/W | 0000_0000 |
| | _0018 | Transmit buffer length Low | R/W | 0000_0000 |
| | _001c | Receive buffer base address | R/W | 0000_0000 |
| | _0020 | Receive buffer status | R | 0000_0000 |
| | _0024 | Receive buffer level | R/W | 0000_0000 |
| | _0028 | Receive buffer base address return | R/W | 0000_0000 |
| | _002c | Control mode register | R/W | 0000_0000 |
| | _0030 | MII mode register | R/W | 00f3_6400 |
| | _0034 | MII command register | R/W | 0000_0000 |
| | _0038 | MII transmit data | R/W | 0000_0000 |
| | _003c | MII receive data | R/W | 0000_0000 |
| | _0040 | Reserved | R/W | 0000_0000 |
| | _0044 | Max, Min, burst length register | R/W | 05dc_2e90 |
| | _0048 | Multicast Address (Most significant) | R/W | 0000_0000 |
| | _004c | Multicast Address (Least significant) | R/W | 0000_0000 |
| | _0050 | MAC address 0 (Higher 2 bytes) | R/W | 0000_0000 |
| | _0054 | MAC address 0 (Lower 4 bytes) | R/W | 0000_0000 |
| | _0058 | MAC address 1 (Higher 2 bytes) | R/W | 0000_0000 |
| | _005c | MAC address 1 (Lower 4 bytes) | R/W | 0000_0000 |
| | _0060 | MAC address 2 (Higher 2 bytes) | R/W | 0000_0000 |
| | _0064 | MAC address 2 (Lower 4 bytes) | R/W | 0000_0000 |
| | _0068 | MAC address 3 (Higher 2 bytes) | R/W | 0000_0000 |
| | _006c | MAC address 3 (Lower 4 bytes) | R/W | 0000_0000 |
| | _0070 | MAC address 4 (Higher 2 bytes) | R/W | 0000_0000 |
| | _0074 | MAC address 4 (Lower 4 bytes) | R/W | 0000_0000 |
| | _0078 | MAC address 5 (Higher 2 bytes) | R/W | 0000_0000 |
| | _007c | MAC address 5 (Lower 4 bytes) | R/W | 0000_0000 |
| | _0080 | MAC address 6 (Higher 2 bytes) | R/W | 0000_0000 |
| | _0084 | MAC address 6 (Lower 4 bytes) | R/W | 0000_0000 |
| | _0088 | MAC address 7 (Higher 2 bytes) | R/W | 0000_0000 |

| | | | | |
|---|---|---|---|---|
| | _008c | MAC address 7 (Lower 4 bytes) | R/W | 0000_0000 |
| | _0090 | Pause frame address (Higher 2 bytes) | R/W | 0000_0180 |
| | _0094 | Pause frame address (Lower 4 bytes) | R/W | c200_0001 |
| | _0098 | Pause frame Type / Op. code | R/W | 8808_0001 |
| | _009c | Pause frame delay value | R/W | 0000_0018 |
| | _00a0 | Transmit buffer base address (High priority) | R/W | 0000_0000 |
| | _00a4 | Transmit buffer length (High priority) | R/W | 0000_0000 |
| | _00a8 | TX queue buffer level (Low) | R | 0000_0000 |
| | _00ac | TX address return (Low) | R | 0000_0000 |
| | _00b0 | TX queue buffer level (High) | R | 0000_0000 |
| | _00b4 | TX address return (High) | R | 0000_0000 |
| | 000b8 – 00ffc | RESERVED | | 0000_0000 |
| Enet Mac 1 Controller | 1920_1000 . . . _10c8 | Same Configuration as Enet Mac 0 Controller | | |
| | 010cc – ffffc | RESERVED | | 0000_0000 |
| Interrupt Controller | 1930_0000 | Source Pending Register | R/W | 0000_0000 |
| | _0004 | Interrupt Mode Register | W | 0000_0000 |
| | _0008 | **Interrupt Enable Mask Register** | R/W | ffff_ffff |
| | _000c | Interrupt Priority Control Register | W | 0000_003f |
| | _0010 | Interrupt Pending Register | R/W | 0000_0000 |
| | _0014 | Interrupt Offset Register | R | 0000_0020 |
| | 00018 – ffffc | RESERVED | | Prev_Read_value |
| PCMCIA Controller | 1940_0000 | Identification Register | R | 0000_0082 |
| | _0004 | Interface Status Register (VS_in) | R | 0000_0000 |
| | _0008 | Interface Control Register (VS_out) | R/W | 0000_0000 |
| | _000c | General Control Register | R/W | 0000_0001 |
| | _0010 | Card Status Change Register | R | |
| | _0014 | Card Status Change Enable Register | R/W | 0000_0000 |
| | _0018 | Setup Timing 0 Register for Bank 0 | R/W | 0000_000f |
| | _001c | Command Timing 0 Register for Bank 0 | R/W | 0000_000f |
| | _0020 | Recovery Timing 0 Register for Bank 0 | R/W | 0000_000f |
| | _0024 | Setup Timing 1 Register for Bank 1 | R/W | 0000_000f |
| | _0028 | Command Timing 1 Register for Bank 1 | R/W | 0000_000f |
| | _002c | Recovery Timing 1 Register for Bank 1 | R/W | 0000_000f |

| | _0030 | Setup Timing 2 Register for Bank 2 | R/W | 0000_000f |
|---|---|---|---|---|
| | _0034 | Command Timing 2 Register for Bank 2 | R/W | 0000_000f |
| | _0038 | Recovery Timing 2 Register for Bank 2 | R/W | 0000_000f |
| | _003c | Reserved | | |
| | _0040 | I/O Start Address 0 | R/W | |
| | _0044 | I/O End Address 0 | R/W | N/a |
| | _0048 | I/O Start Address 1 | R/W | 0000_0000 |
| | _004c | I/O End Address 1 | R/W | 0000_0000 |
| | _0050 | Direction control for GPIO-0 (Low) | R/W | 0000_0000 |
| | _0054 | Direction control for GPIO-1 (High) | R/W | 0000_0000 |
| | _0058 | Output value for GPIO-0 (Low) | R/W | 0000_0000 |
| | _005c | Output value for GPIO-1 (High) | R/W | 0000_0000 |
| | _0060 | Input value of GPI-0 (Low) | R/W | 0000_0000 |
| | _0064 | Input value of GPI-1 (High) | R/W | 0000_0000 |
| | 00068 – 7fffc | RESERVED | | 0000_0000 |
| CardBus Controller | 1948_0000 | **CardBus Bridge Command Register** | R/W | 0000_0000 |
| | _0004 | CardBus Bridge Status Register | R/W | 0000_0000 |
| | _0008 | Retry Time Control Register | R/W | 0000_0000 |
| | _000c | Clock Speed Selection Register | R/W | 0000_0000 |
| | 80020 – ffffc | RESERVED | | 0000_0000 |
| Cache | 1950_0000 | Cache Control Register | R/W | 0000_0000 |
| | 00004 – ffffc | RESERVED | | Cache_con_reg |

**Version 1.5**

## 2.1.4.    ARM7TDMI Core

Please refer to ARM7TDMI manual.

**Version 1.5**

41

## 2.2. Cache

This section describes the cache system of HMS30C7110®.

### 2.2.1.    Architecture

The cache of HMS30C7110® has the following characteristics.

1.  4K byte unified (data + instruction)
2.  4-way set associative
3.  64 sets
4.  Write back policy
5.  Read miss line fill up
6.  8 words depth write buffer

This cache uses "write back policy" for high performance in write accesses to cacheable area. To ensure the memory consistency between cache and main memory it supports user controllable line flush mechanism. The 8 word-depth write-buffer is used to further boost up the performance.

The bit position 31 of address (most significant bit) determines whether the access is cacheable or non-cacheable. Value of 0 at the $31^{st}$ address bit (0x00000000H to 0x0fffffffH) is for cacheable area, while 1 (0x10000000H to 0x1fffffffH) is for non-cacheable area.

The write buffer receives and keeps write accesses to cacheable/non-cacheable area and performs actual memory accesses when the bus is idle, program flow (dependency) forces, or user forces. Accesses to cacheable/non-cacheable area originally do not go into write buffer, i.e., directly go to main memory. However, user can change this by programming user controllable registers.

### 2.2.2.    User Accessible Registers (Base = 0x1950_0000)

There is one 32bit register to control the cache and write buffer operations (offset = 0x0).

**Table 2.6 Cache and Write Buffer Control Register**

Address： **1950_0000**

| Bits | Access | Default | Description |
|------|--------|---------|-------------|
| 31:20 | RW | 0x0 | Reserved and should be set to 0. |
| 19 | RW | 0x0 | This bit controls Burst Selection. |

**Version 1.5**

| | | | |
|---|---|---|---|
| | | | 0 – Out HBURST as 3'b011 (Increment 4) |
| | | | 1 – Out HBURST as 3'b001 (Increment) |
| 18 | RW | 0x0 | This bit controls the use of cache.<br><br>0 – Cache is used for Data and Instruction (Unified)<br><br>1 – Cache is used only for Instruction (I-Cache) |
| 17 | RW | 0x0 | This bit controls the "bufferability" of write accesses to cacheable areas.<br><br>0 – accesses to cacheable area are not stored into the write buffer.<br><br>1 – accesses to cacheable area are stored into the write buffer.<br><br>Buffering write accesses to cacheable area can increase the performance, but in some cases it may cause problems. So it should be used with care. We recommend using the default value (non-bufferable) if user is not sure how to handle the consistency problem. |
| 16 | W | 0x0 | This bit flushes the current contents of the write buffer to the main memory (1 – flush). This is recommended to be used only when the cache is off. Normally it is better to read the same address of the previous write to flush the write buffer. |
| 15:13 | RW | 0x0 | These 3 bits control the burst length of the write buffer's write operation. Value 0-7 means 1-8 burst length. As an example, burst length 1 means that write buffer tries to write the content of the buffer as soon as at least one access comes into the buffer, and only one write access will be performed at a time. On the other hand, burst length 8 means that the write buffer will wait till there comes 8 write accesses into write buffer, and those accesses will be written to main memory in a single burst access.<br><br>Generally, the larger the burst length, the higher the bus efficiency. But the latency (from the time CPU writes a data to the actual writing to main memory) will increase as you use larger burst length. |
| 12 | RW | 0x0 | This bit controls the "bufferability" of write accesses to non-cacheable areas.<br><br>0 – accesses to non-cacheable area are not stored into the write buffer.<br><br>1 – accesses to non-cacheable area are stored into the write buffer.<br><br>Buffering write accesses to non-cacheable area can increase the performance, but in some cases it may cause problems. So it should be |

**Version 1.5**

| | | | | used with care. For example, write to control registers may require the "read again" the same address to flush the write buffer. |
|---|---|---|---|
| 11:6 | RW | 0x0 | Specify the set to invalid or flush (write back) – 0 to 63 |
| 5:4 | RW | 0x0 | Specify the way to invalid or flush (write back) – 0 to 3 |
| 3 | W | 0x0 | Line flush (write back) This bit flushes (write back) a cache line (4 words) corresponding to a specific way and a set specified in bit [5:4] and [11:6], respectively. It checks the dirty bit of a given line, write back if necessary, and then clears the valid bit. This should be activated only when the cache is off. This bit is automatically cleared to 0 as soon as the write back finishes. (Actually when cache is off, it finishes immediately even before to execute the next following instruction.) This bit is used for the consistency between the cache and the main memory after the cache is off – i.e., writes the contents of the cache if they are different (new) from those of main memory (old). |
| 2 | W | 0x0 | Line invalidate This bit invalidates a cache line (4 words) corresponding to a specific way and a set specified in bit [5:4] and [11:6], respectively. This clears the valid bit of a given line without writing the contents of the line to main memory. This should be activated only when the cache is off. This bit is automatically cleared to 0 as soon as the invalidation finishes. (Actually when cache is off, invalidation finishes immediately even before to execute the next following instruction.) |
| 1 | RW | 0x0 | Parallel/Sequential control (0 – parallel, 1 – sequential) This bit controls the access sequence of tag RAM and data RAM. Parallel access of tag RAM and data RAM is for high performance, while sequential access is for low power operations. |
| 0 | RW | 0x0 | Cache on/off (0 – off, 1 – on) This bit controls the on and off the cache. After reset, all the valid bits of cache are cleared to 0. So there are no required procedures to turn on the cache in the initial state. However, there should be some caution when turn off the cache since even if the cache is turned off, all the operations of the cache except the line fill is performed as usual. For example, a |

**Version 1.5**

| | | | |
|---|---|---|---|
| | | | cache read hit can occur, and the corresponding data/inst from the cache goes to CPU. Similarly, a write hit can occur, and the corresponding data write to cache follows.<br><br>So, after turn off the cache, user should flush all the cache lines (preferably using codes in a non-cacheable area) if he/she wants to get(put) data/inst from(to) main memory directly. |

**Version 1.5**

# 2.3. Clock/Watchdog Timer

The HMS30C7110® integrates a clock module, which is composed of the clock generator, the reset de-bouncing circuit, and watchdog timer (WDT). The clock generator provides the system clock.

The main PLL multiplies the incoming external crystal clock input by 7(default). Assuming 10 MHz external clock is used and 70MHz of the CPU speed is intended, the register setting is to be "multiplying by 7". An internal register also provides a method to determine the ratio between CPU clock and bus clock. It can be either 1:1 or 2:1. To run CPU with maximum performance, set the PLL output frequency to 70MHz and set the ratio to 1:1, then CPU runs in 70MHz and all others run at 70 MHz.

Watchdog Timer generates the reset signal internally when the timer reaches the end value. So, the controlling software must reload the preset value to the timer before the timer expires during normal operation. Complete descriptions of these registers are given in the Register Section.

## 2.3.1.    Block Diagram

The Clock Module consists of several blocks such as a register file, a reset de-bouncer, a clock generator, and a watchdog timer.
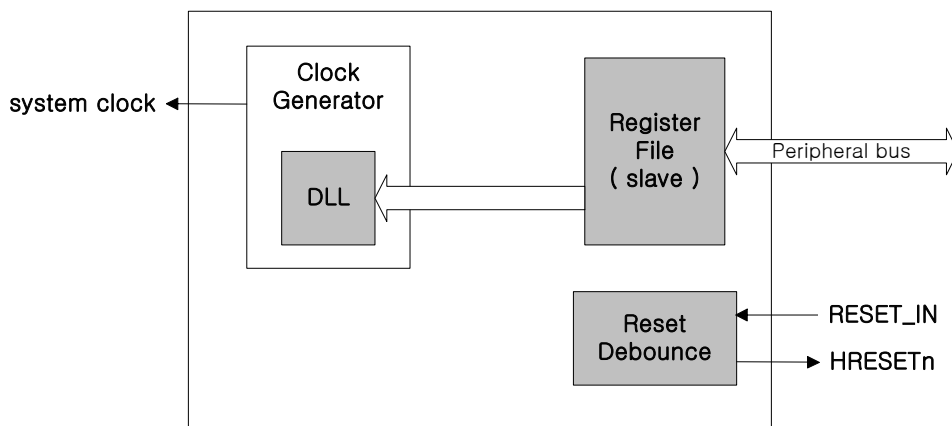


**Figure 2.1 Block Diagram of Clock Module**

## 2.3.2. User Accessible Registers (Base = 0x1830_0000)

This section describes the registers in the Clock Module. The address value on the table shows the relative address in hexadecimal. The width describes the number of bits in the register. The access field specifies the valid access type for the register, where 'RW' stands for read and writes access, and 'RO' for read only access, respectively. 'C' following 'RW' or 'RO' means the corresponding bit is cleared after writing "1" in the matching position.

**Table 2.7 Registers for PLL & Watchdog Timer**

| Name | Address | Width | Access | Description |
|---|---|---|---|---|
| PLL Enable | 0x00 | 32 | RW | Enable for PLL clocks |
| WDT control | 0x04 | 32 | RW | Control for watch dog timer |
| WDT interval | 0x08 | 32 | RW | Time interval for watch dog timer |
| Main PLL Control | 0x0c | 32 | RW | Control for main PLL |
| PLL Reset | 0x10 | 32 | R | Status of PLL |

### 2.3.2.1. PLL Enable

This register is used to select internal three clocks between external low frequency clock and internal PLL output clock.

**Table 2.8 PLL control**

Address：**1830_0000**

| Bits | Access | Default | Description |
|---|---|---|---|
| 31:4 | | 0x00 | Reserved |
| 3 | RW | 0x1 | Bus Clock Ratio vs. CPU clock. SCLK is from the Bus Clock. 0 = Bus Clock is Divided by 2 from CPU clock 1 = Bus Clock is delibered directly from CPU clock |
| 2:1 | | 0x0 | Reserved |

| 0 | RW | 0x0 | Main PLL enable<br>When this bit is set to logic 1, internal CPU and system clock is made from PLL output. If this is set to logic 0, the internal CPU and system clock is made directly from external clock pin. The system clock is same to the Bus Clock or System Bus Clock. |
|---|----|-----|--------|

## 2.3.2.2. Watch Dog Timer (WDT) Control

The Watch Dog control register selects the operation mode of watchdog timer.

**Table 2.9 Watch Dog Timer control**

Address：1830_0004

| Bits | Access | Default | Description |
|------|--------|---------|-------------|
| 31:5 |        | 0x00    | Reserved    |
| 4    | RW     | 0x0     | Enable<br>0 = Watch dog timer disable<br>1 = Watch dog timer enable |
| 3:2  |        | 0x0     | Reserved    |
| 1: 0 | RW     | 0x0     | Pre-scaler<br>00 = System Bus Clock<br>01 = System Bus Clock/ 4<br>10 = System Bus Clock/ 8<br>11 = System Bus Clock/ 256 |

## 2.3.2.3. Watch Dog Timer (WDT) Interval

This register contains the watchdog timer interval value. The input clock is the output clock of the pre-scaler described in *Watch Dog Timer (WDT) Control*

**Table 2.10 Watch Dog Timer Interval**

Address：1830_0008

| Bits | Access | Default | Description |
|------|--------|---------|-------------|
| 31: 0 | RW | 0x0 | Interval (Write) or Current WDT counter value (Read)<br>This 32-bit field contains the interval value that will be loaded to down counter periodically. |

## 2.3.2.4. Main PLL Control

The main PLL control register selects the operating mode of PLL. Refer to H25PL12S Data sheet for more detailed information about PLL control.

**Table 2.11 Main PLL Control**

Address : **1830_000C**

| Bits | Access | Default | Description |
|------|--------|---------|-------------|
| 31:29 | RW | 0x5 | lfm : loop filter mode selector.<br>** Use default value for normal operation, set 0x7 only if external filter logic is used for lower frequency (< 150KHz). |
| 28:15 | RW | 0x0c | m : Feedback divisor |
| 14:7 | RW | 0x1 | n : Reference divisor<br>VCO clock frequency will be determined with M and N based on following equation: $F_{vco} = F_{ref} \times (m+2) / (n+1)$<br>As a default, $F_{vco} = 10MHz \times (12+2) / (1+1) = 70MHz$. |
| 6:5 | RW | 0x0 | p : Post divisor<br>This field defines division factor after VCO. For example, if p=1 and vco output frequency is 20MHz, final PLL output frequency will be 10MHz ($F_{pll} = F_{vco} / (p+1)$ ) |
| 4 | RW | 0x0 | pd : PLL power down mode except VCO<br>If set to "1", PLL will not generate clock though VCO is still working. |
| 3:2 | RW | 0x1 | vc : VCO range control vector<br>This field defines operation range of VCO.<br>00 : 40MHz – 100MHz<br>01 : 60MHz – 120MHz<br>10 : 80MHz – 140MHz |

| | | | 11 : 100MHz – 160MHz |
|---|---|---|---|
| 1 | RW | 0x0 | vcoinit : VCO initialize signal<br>During power-up sequence, vcoinit is recommended to be activated for more than 100ns just after deactivation of the vcopd signal. |
| 0 | RW | 0x0 | vcopd : VCO power down mode<br>If set to "1", PLL will not generate clock and VCO will stop. |

$$Fck = Fref \times (m+ 2) / (n + 1)$$

### 2.3.2.5. PLL Status

This register controls Software reset of PLL and some of control signals of PLL module and indicates PLL locking status.

**Table 2.12 PLL Status**

Address：**1830_0010**

| Bits | Access | Default | Description |
|---|---|---|---|
| 31 | RW | 0x0 | BYPASS, PLL bypass mode "active high" |
| 30 | RW | 0x0 | Cnttest, PLL counter toggle test "active high" |
| 29 | RW | 0x0 | Lfo, PLL External loop filter port " analog" |
| 28:25 | RW | 0xa | ICP, PLL charge pump bias current control vector |
| 24 | RW | 0x0 | Tdm, PLL digital part test mode "active high" |
| 23:22 | RW | 0x0 | Tpdud, PLL charge pump test mode (Normal mode '00') |
| 21:5 | | 0x0 | Reserved |
| 4 | RW | 0x0 | **PLL Software RESET "active high"** |
| 3:1 | | 0x0 | Reserved |
| 0 | R | 0x0 | **Main PLL locking end: high active** |

Refer MAGNACHIP PLL User guide

# 2.4. Memory Controller (Flash/ROM)

The HMS30C7110® has integrated External Bus controller that supports 8-bit/16-bit/32-bit data bus width. The space is divided into 3 static memory (FLASH/ROM) banks. Each bank has fixed size of 4MB. The operating clock of external bus controller is the internal system bus clock. The bus timing exclusively depends on the register programming. The external bus controller is fully configurable through a set of control register. Complete descriptions of these registers are given in the Register Section.

The twenty-two address lines, RA [21:0], allow HMS30C7110® to support up to 1M×32 bit, 2Mx16 bit, or 4Mx8 bit space. Three nRCS lines make three times bigger space.
Each block can be configured independently for the bus style, the wait enable, the shaping of bus control signal.

## 2.4.1.    Block Diagram

The external bus controller of HMS30C7110® consists of several blocks such as main state machine, register file, data-path, signal generator and address generator.

Commands fed via system bus are interpreted in the main state machine and the main state machine generates control signals to all of the signal generation blocks (address generator, signal generator and data-path) using the state signal.

The data-path block makes DATAOUT signal using HWDATA and makes HRDATA using DATAIN signal according to state signal. In write operation, the data-path block asserts the direction signal to enable output path of data pads. The signal generator makes nRCS, nOE, and nWE.

**Figure 2.2 Block Diagram of External controller**

## 2.4.2.    User Accessible Registers (Base = 0x1900_0000)

This section describes the bit assignment of the configuration registers of the ROM controller. The address field indicates a relative address in hexadecimal. Width specifies the number of bits in the register and access specifies the valid access types of the register. Where RW stands for read access and write access, 'RO' for read only access. A 'C' appended to 'RW' or 'RO', indicates that some or all of the bits can be cleared after writing '1' in corresponding bit.

**Table 2.13 Registers for ROM controller**

| Name | Address | Width | Access | Description |
|---|---|---|---|---|
| Configuration 0 | 0x00 | 32 | RW | Timing configuration of bank 0 |
| Configuration 1 | 0x04 | 32 | RW | Timing configuration of bank 1 |
| Configuration 2 | 0x08 | 32 | RW | Timing configuration of bank 2 |

## 2.4.2.1. Configuration 0~3

These registers include timing information for all banks.

**Table 2.14 Configuration Registers Bit Definition**

Address：**1900_0000, 1900_0004, 1900_0008**

| Bits | Access | Default | Description |
|---|---|---|---|
| 31:30 | | 0 | Reserved |
| 29:28 | RW | 0 | BUS MODE<br>00 = A/D non-mux style<br>01 = A/D mux Intel style<br>10 = A/D mux, Motorola style<br>11 = Reserved |
| 27 | RW | 0 | WAIT ENABLE<br>This 1 bit field accepts/denies the external wait (GPIO2/Wait) signal pin expending the access time.<br>0 = disable<br>1 = enable |
| 26 | RW | 0 | This bit selects the active level of wait signal, which would be connected to GPIO2/Wait on pin 206. Each setting for each Configuration register affect only on assigned address range.<br>0 = active low input<br>1 = active high input |
| 25:24 | RW | 0x3 | DATA BUS WIDTH<br>00 = 8 bit wide bus<br>01 = 16 bit wide bus |

| | | | |
|---|---|---|---|
| | | | 10 = 32 bit wide bus<br>11 = Reserved<br>External pull-up/downs on ADDR19 and ADDR18 applies for booting configuration. Boot program should be appeared on nRCS0. Can be changed by Configuration 0 afterward. |
| 23:20 | RW | 0x3<br>0xf | ASTB TIME<br>This 4-bit field selects the pulse width of the address strobe signal when A/D mux bus mode is on. |
| 19:16 | RW | 0x3<br>0xf | ADDR TIME<br>This 4-bit field selects the address phase interval when A/D mux bus mode is on. |
| 15:12 | RW | 0x0<br>0xf | RECOVERY TIME<br>This 4-bit field selects the recovery time delay, this value guarantee the interval from the end of one chip select assertion to the start of another chip select assertion. |
| 11: 8 | RW | 0x0<br>0xf | SETUP TIME<br>This 4-bit field selects the setup time duration, this value guarantees the interval from chip select assertion to read enable or write enable assertion. |
| 7: 4 | RW | 0xf | ACCESS TIME<br>This 4-bit field selects the access time and this value guarantees the asserted pulse width of read enable or write enable signal. |
| 3: 0 | RW | 0xf | HOLD TIME<br>This 4-bit field selects the hold time duration and this value guarantees the interval from read enable or write enable de-assertion to chip select de-assertion. |

Note: The first value in the boxes with two defaults values show the default for ROM Bank0 and the second value for the remaining two Banks, Bank1 and Bank2.

Note: All units are based on the main clock period.

## 2.4.3. Timing Diagram

### 2.4.3.1. Read Access
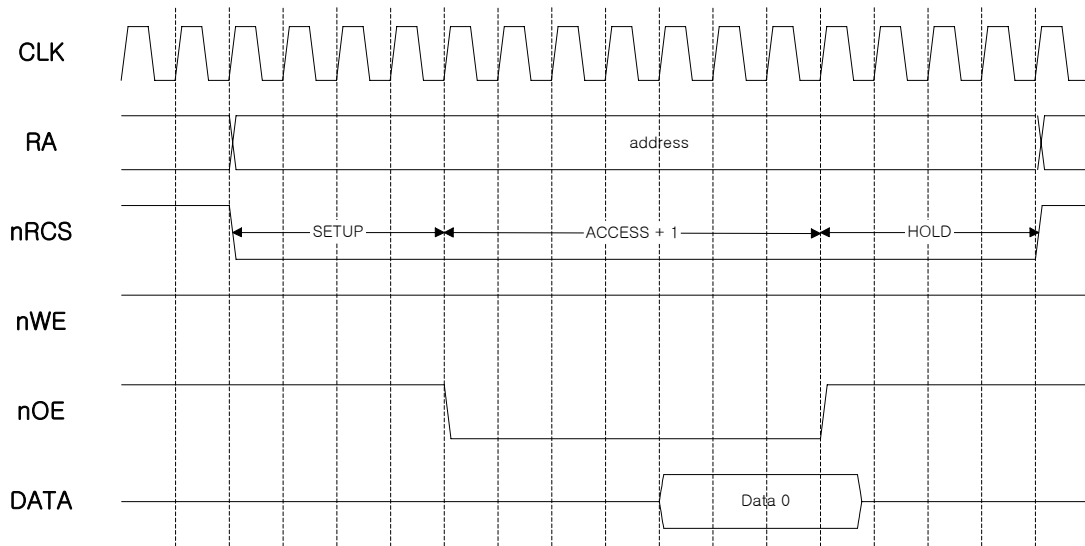


**Figure 2.3 Read cycle timing (setup = 4, access = 6, hold = 4)**

### 2.4.3.2. Write Access
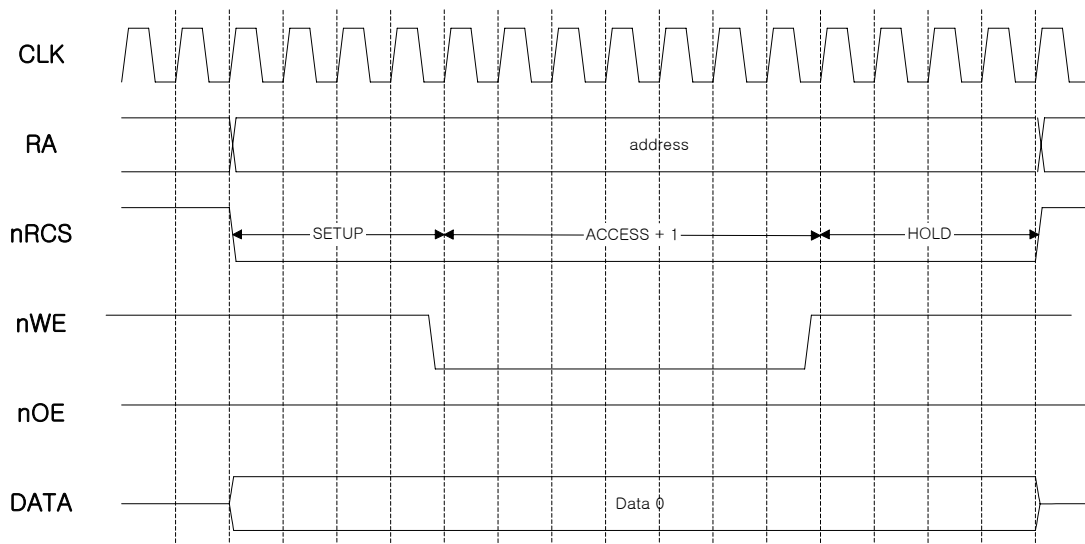


**Figure 2.4 Write timing (setup = 4, access = 6, hold = 4)**

# 2.5. Memory Controller (SDRAM)

The HMS30C7110® integrates a SDRAM controller (SDRAMC) that supports a 32-bit SDRAM array. EDO or FPM type DRAM is not supported. The operating clock is directly related to the internal system bus speed. When the system bus clock runs at 70MHz, the SDRAMC runs at 70MHz. When the system bus clock runs at 35MHz, the SDRAMC runs at 35MHz. The system clock can be selected by PLL output clock or half of PLL output clock. The SDRAMC is fully configurable through a set of control register. Complete descriptions of these registers are given in the Register Section. The operating clock of SDRAMC is shown on SCLK.

Note: SDRAM space works either in 32-bit wide or in 16-bit wide. So, any design using the SDRAM space as program memory space, it can be either 32-bit wide memory structure or 16-bit wide structure. Being used as data memory space, SDRAM may be connected to 8-bit wide data bus sacrificing the address corresponding to unconnected data bus.

The thirteen multiplexed address lines, MA[12:0], and two bank select signals, A13 and A14, allow the HMS30C7110® to support 1M, 2M, 4M, 8M, 16M and 32M×32 bit arrays or 1M, 2M, 4M, 8M, 16M, 32M and 64M×16 bit arrays. Both symmetric (Number of row addresses and number of column addresses are same) and asymmetric addressing (Number of row addresses and number of column addresses are different) is supported. The HMS30C7110® has one nSCS pin. The maximum block size is 128Mbytes. For write operations of less than word in size, the HMS30C7110® supports byte-wise write. The HMS30C7110® provides refresh functionality with programmable rate (normally SDRAM refresh rate is 1 / 64ms).

## 2.5.1. Block Diagram

The SDRAM controller of HMS30C7110® consists of several blocks such as main state machine, register file, refresh block, data-path, signal generator and address generator.

Commands come in via system bus will be interpreted in the main state machine and the main state machine controls all of the signal generation blocks (address generator, signal generator and data-

path) using state information of main state machine.

The refresh block makes bus request signal to system bus arbiter to start refresh operation and when the system bus arbiter gives bus grant signal to SDRAM controller, the main control logic detects this signal and makes refresh command to signal generator.

The data-path block makes DATAOUT signal using system write data bus and makes system read data bus using DATAIN signal according to state signal. In write operation, the data-path block asserts the direction signal to enable output path of data pads.
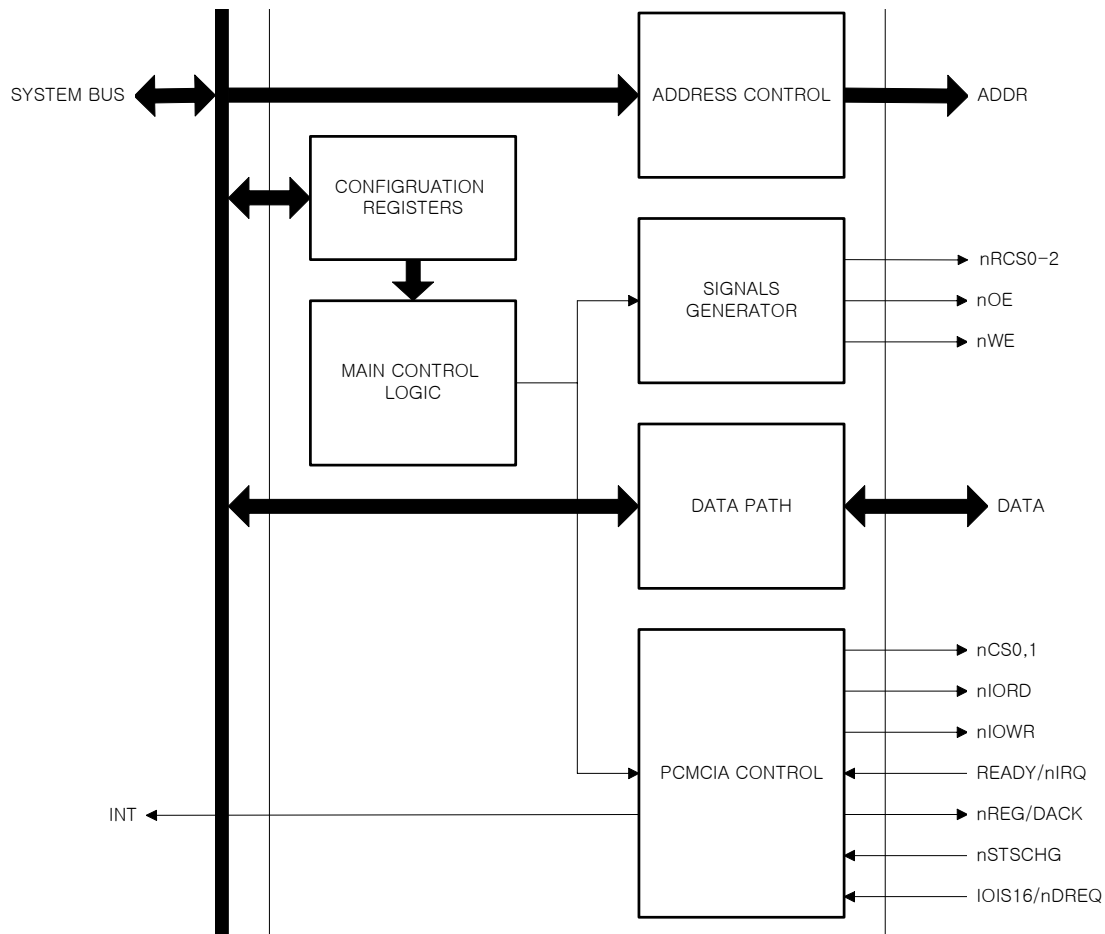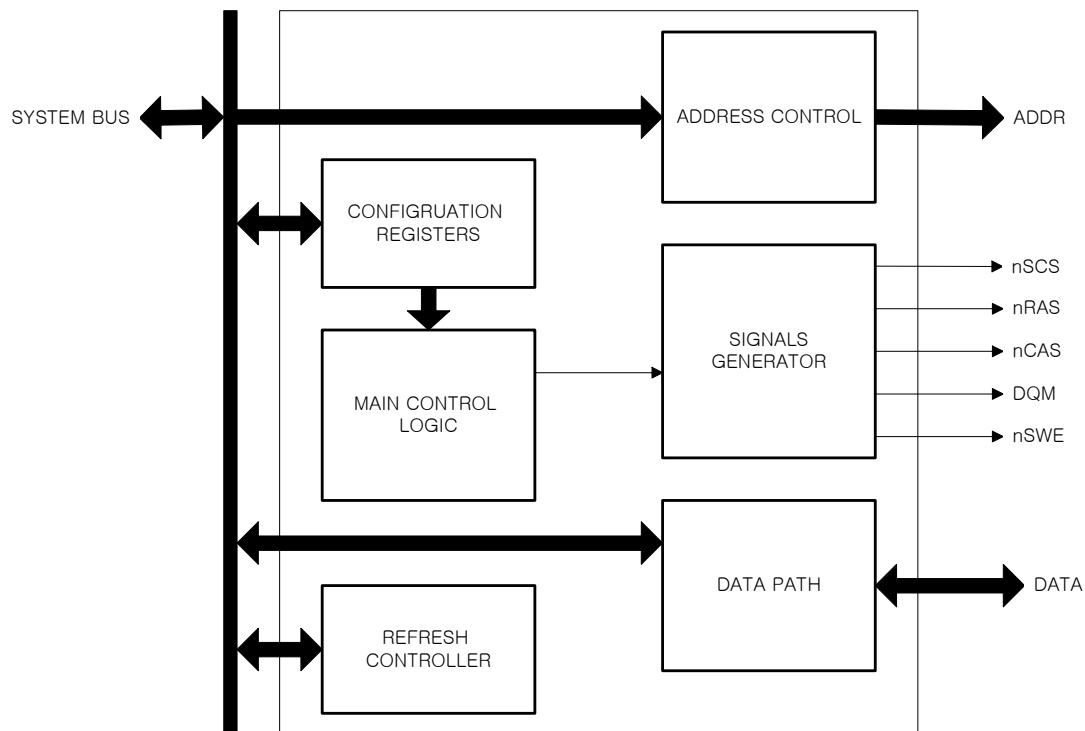


**Figure 2.5 Block Diagram of SDRAM controller**

## 2.5.2. User Accessible Registers (Base = 0x1908_0000)

This section describes all base, control and status register inside the SDRAM controller. The address field indicates a relative address in hexadecimal. Width specifies the number of bits in the register and access specifies the valid access types that register. Where 'RW' stands for read and

write access, 'RO' for read only access. A 'C' appended to 'RW' or 'RO', indicates that some or all of the bits can be cleared after a write '1' in corresponding bit.

**Table 2.15 Registers for SDRAM controller**

| Name | Address | Width | Access | Description |
|------|---------|-------|--------|-------------|
| Configuration | 0x00 | 32 | RW | SDRAM Configuration |
| Reserved | 0x04 | 32 | | Reserved |
| Reserved | 0x08 | 32 | | Reserved |
| Reserved | 0x0C | 32 | | Reserved |
| Timing | 0x10 | 32 | RW | SDRAM timing parameters |
| Refresh Interval | 0x14 | 32 | RW | Refresh interval value & MCLK Timing |
| Init control | 0x18 | 32 | RW | Initialize start register |
| Address Map | 0x1c | 32 | RW | Select address area (Flash or SDRAM) |

## 2.5.2.1. Configuration

These registers include the information of memory size of each bank.

**Table 2.16 Configuration register**

Address：**1908_0000**

| Bits | Access | Default | Description |
|------|--------|---------|-------------|
| 31:22 | | 0x0 | Reserved |
| 21:20 | RW | 0x0 | ROW WIDTH<br>00 = 10-bit row width<br>01 = 11-bit row width<br>10 = 12-bit row width<br>11 = 13-bit row width |
| 19:18 | | 0x0 | Reserved |
| 17:16 | RW | 0x0 | COLUMN WIDTH<br>00 = 8-bit column width<br>01 = 9-bit column width |

| Bits | Access | Default | Description |
|---|---|---|---|
| | | | 10 = 10-bit column width |
| | | | 11 = Reserved |
| 15:14 | | 0x0 | Reserved |
| 13: 8 | RW | 0x0 | START ADDRESS (Reserved for future usage) |
| 7: 6 | | 0x0 | Reserved |
| 5: 0 | RW | 0x3f | END ADDRESS (Reserved for future usage) |

### 2.5.2.2. Timing

This register includes the timing information for all banks.

**Table 2.17 Timing register**

Address：1908_0010

| Bits | Access | Default | Description |
|---|---|---|---|
| 31 | WO | 0x0 | NOP<br>When writing 1 at this bit, it will make NOP command to SDRAM |
| 30:28 | RW | 0x7 | TMODE (Mode Register Set to Active Delay)<br>This 3-bit field specifies the number of cycles guaranteed between mode register set command at the end of the initialization sequence and the first activate command. |
| 27 | | 0x0 | Reserved. |
| 26:24 | RW | 0x7 | DPL (Write to Pre-charge Delay)<br>This 3-bit field specifies the number of cycles guaranteed between write command and a pre-charge command. |
| 23 | | 0x0 | Reserved. |
| 22:20 | RW | 0x7 | RCD (RAS to CAS Delay)<br>This 3-bit field specifies the number of cycles between RAS and CAS. |
| 19 | | 0x0 | Reserved |
| 18:16 | RW | 0x7 | RP (Row Pre-charge Time)<br>This 3-bit field specifies the number of cycles needed for the pre-charge command. A RP value of zero results in a RAS pre-charge of two clocks, as if RP was selected to 1. |

| 15:12 | RW | 0xf | RAS (Row Active Time) <br> This 4-bit field specifies the number of cycles guaranteed between an activate command and a pre-charge command. |
|---|---|---|---|
| 11: 8 | RW | 0xf | RC (Row Cycle Time) <br> This 4-bit field specifies the number of cycles for a refresh command. It also specifies the number of cycles between bank activation commands to the same bank. |
| 7: 6 | | 0x0 | Reserved |
| 5: 4 | RW | 0x7 | CL (CAS Latency) <br> This value is used to program the SDRAM devices during the Initialization sequence, and internally by the SDRAM controller. <br> 0 = 1 clock <br> 1 = 2 clocks <br> 2 = 3 clocks <br> 3 = Reserved |
| 3: 2 | | 0x0 | Reserved |
| 0 | RW | 0x1 | Row Hit Enable <br> This value is used to enable the checking of the row address from internal logic matched with row address of destination SDRAM. It may cause one or two clock delay in accessing SDRAM but will make sure that correct operation will be happening. <br> 0 : Do not check whether row address is HIT or not. <br> 1 : Check whether row address is HIT or not. |

### 2.5.2.3. Refresh Interval

This register includes refresh interval value and MCLK tuning control reginter.

**Table 2.18 Refresh Interval**

Address：**1908_0014**

| Bits | Access | Default | Description |
|---|---|---|---|
| 31:29 | | 0x0 | Reserved. |

| 28 | | 0x1 | Clock Select for SDRAM Read. |
| | | | 1 : Clock Source = SCLK output |
| | | | 0 : Clock Source = MCLK input |
| 27:24 | | 0x3 | Delay for SDRAM Read Clock (0-15ns). |
| 23:21 | | 0x0 | Reserved. |
| 20 | | 0x0 | Clock Select for SDRAM Write. |
| | | | 1 : Clock Source = SCLK output |
| | | | 0 : Clock Source = MCLK input |
| 19:16 | | 0x3 | Delay for SDRAM Write Clock (0-15ns). |
| 15:13 | | 0x0 | Reserved |
| 12: 0 | | 0x0 | Refresh Interval |
| | | | This 13-bit field is used to enable refresh and set the refresh period. When set to zero, refresh cycles are disabled. When set to non-zero, refresh cycles are enabled, and are requested internally based on this programmed value. RP and RC control the timing waveform of refresh cycle. This field is set to zero on the hard reset, so refreshes are disabled after a hard reset. This field is a one-based number, and is programmed in CLK period.(1/CLK) |
| | | | $t_{REFINT} = (REFINT + 1) /CLK$ |

### 2.5.2.4. INIT Control

This register includes init bit.

**Table 2.19 INIT Control register**

Address：**1908_0018**

| Bits | Access | Default | Description |
|---|---|---|---|
| 31:1 | | | Reserved. |
| 0 | WO | 0 | INIT |
| | | | This bit when set invokes an SDRAM power up initialization sequence. All other fields in this register must be programmed with valid values before or at the same time init is set. Once the initialization sequence is |

| | | | complete, hardware clears this bit, so it may be polled by software to determine when the SDRAM(s) is (are) available for use. |
|---|---|---|---|

### 2.5.2.5. Address Map Control

This register includes data bus width selection/memory re-map bits.

**Table 2.20 Address Control register**

Address：**1908_001C**

| Bits | Access | Default | Description |
|---|---|---|---|
| 31:2 | | | Reserved. |
| 1 | RW | 0 | 32/16Bit Data bus width selection<br>0 = 32 bit<br>1 = 16 bit |
| 0 | RW | 0 | SDRAM_REMAP<br>0 = Address 0x0000_0000 is located at Flash memory area<br>    (SDRAM will be 0x1000_0000)<br>1 = Address 0x0000_0000 is located at SDRAM area |

## 2.5.3.    Timing Diagram

### 2.5.3.1. Read/Write Access
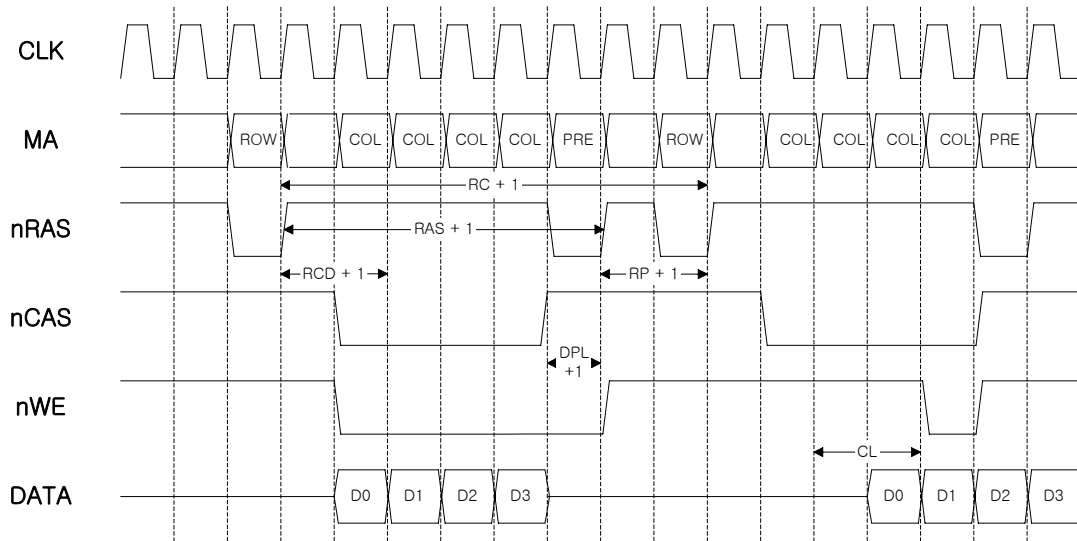
**Figure 2.6 Read/Write cycle**

## 2.5.3.2. Refresh Access
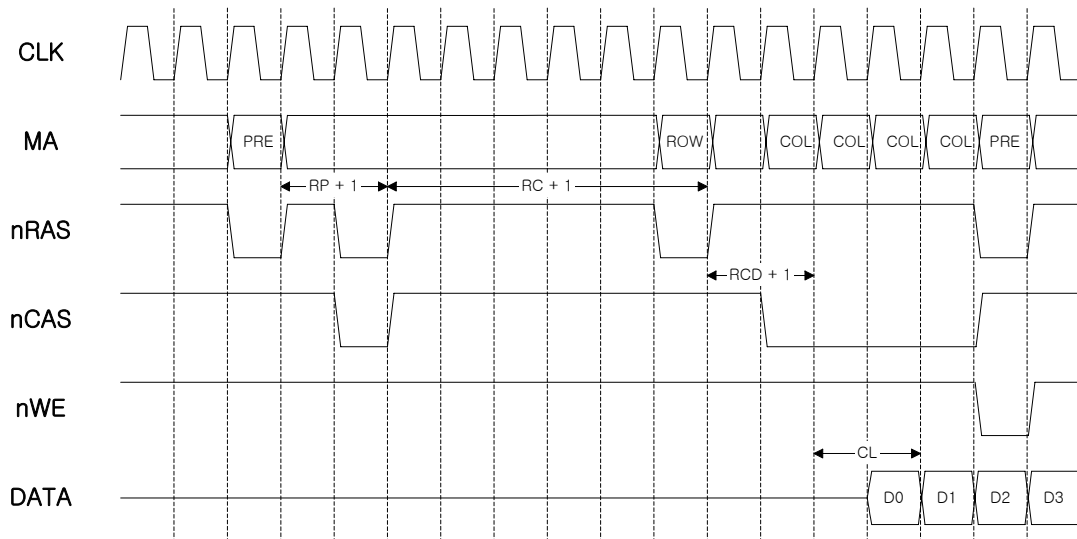


**Figure 2.7 Refresh cycle**

**Version 1.5**

## 2.5.3.3. Initialization

**Table 2.21 Initialization Sequence**

| Sequence | SDRAM requirement | Comments |
|---|---|---|
| 1 | Apply power and start clock | MCLK operates during RESET period |
| 2 | Maintain stable power and clock for at least 1 ms | Power on reset normally takes longer than 1 ms |
| 3 | NOP command (optional) | Software asserts NOP command using command register |
| 4 | Waiting for min 200 us | Software waits for 200 us using for-loop or while-loop |
| 5 | | Software sets valid values in Timing registers |
| 6 | | Software sets the refresh interval |
| 7 | Pre-charge all banks of all blocks | Software sets the init bit of command register |
| 8 | Perform 20 times refresh cycles | |
| 9 | Program SDRAM mode register | |
| 10 | | Polling the init bit and if init bit is cleared, then initialization is finish |
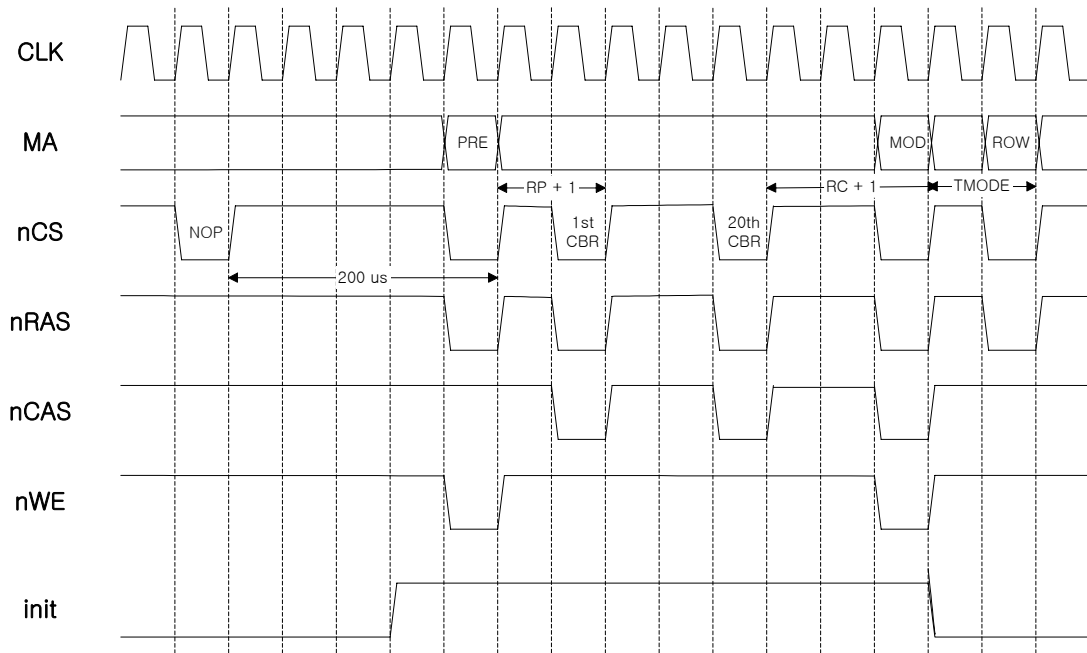
**Figure 2.8 Initialization timing**

# 2.6. Ethernet MAC

HMS30C7110® provides an Ethernet Media Access Controller (MAC) that operates at either 10 Mbps or 100 Mbps in half-duplex or full-duplex mode. In half-duplex mode, the controller supports the IEEE 802.3 Carrier Sense Multiple Access with Collision Detection (CSMA/CD) protocol. It supports the IEEE 802.3 MAC control layer, including the pause operation for flow control for full-duplex mode.

The Ethernet MAC layer supports both Media Independent Interface (MII) / Reduced MII (RMII) and 7-wire interface. The host side supports DMA interface to the system bus. The MAC layer itself consists of the Receive and the Transmit blocks, a flow control block, multiple network address storage, and a number of commands, and status registers.

The Transmit and Receive clocks will be supplied either from Physical layer devices or external clock sources. They are either 2.5MHz at the 10 Mbps or 25MHz at the 100 Mbps. The external clock for RMII is 50MHz regardless of data rate. The MII conforms to the ISO/IEC 802-3 standard for a media-independent layer which separates physical layer issues form the MAC layer.

It also provides MAC address filtering, which is to drop frames with designated MAC addresses while all other frames are received.

## 2.6.1.     Block Diagram

HMS30C7110® Ethernet MAC module consists of several sub-modules such as MII management, physical layer interface, Receive, Transmit, Control, and Host Interface. Frames come in via either 7-wire interface (bit-wide) or MII (nibble-wide) / RMII (di-bit-wide) and Receive module will detect Start of Frame.

Once it verifies a valid sync pattern, a frame will be stored in the MAC Receive FIFO starting from the destination address field. The state machine also monitors total number of bytes in a frame by looking at data length field. When the level of the MAC Receive FIFO is more than the threshold value, the stored frame will be read to the System Receive FIFO at system clock speed. Another state machine running at the system clock speed will detect destination address, data length, and CRC for validity of the frame. The destination address is sent to Address Compare Block for comparison with the stored MAC addresses. If the destination address is found to be a valid address, then the frame will be written into the System Receive FIFO. If the received address is a multicast

**Version 1.5**

address, then CRC will be calculated over 48-bit destination address, and 5 most significant bits are used to locate one of 64 bits of the multicast address register. If the bit to be pointed by the 5 most significant bits is set, then the frame will be received. With sufficient number of bytes held in the System Receive FIFO, the DMA engine will start transfer the stored frame to the system memory space that is pre-programmed by the CPU. An interrupt will be asserted to notify that a frame transfer is completed and the CPU should write the start address of the next buffer for the next incoming frame.

In transmit operation, the CPU prepares a frame and notify the MAC that a frame is ready to transfer. The MAC DMA engine moves the frame from the system memory to System Transmit FIFO. When the state machine detects there is a frame being stored in the System Transmit FIFO and that carrier sense is not detected, the frame will be transferred. Pad byte (0x00) may be inserted if required, and CRC will be appended at the end of the frame. Preamble will be added before the frame, jam signaling will be asserted when collision occurs. The data stream will be either serialized for the 7-wire interface or converted to nibbles/bi-bits for MII/RMII.
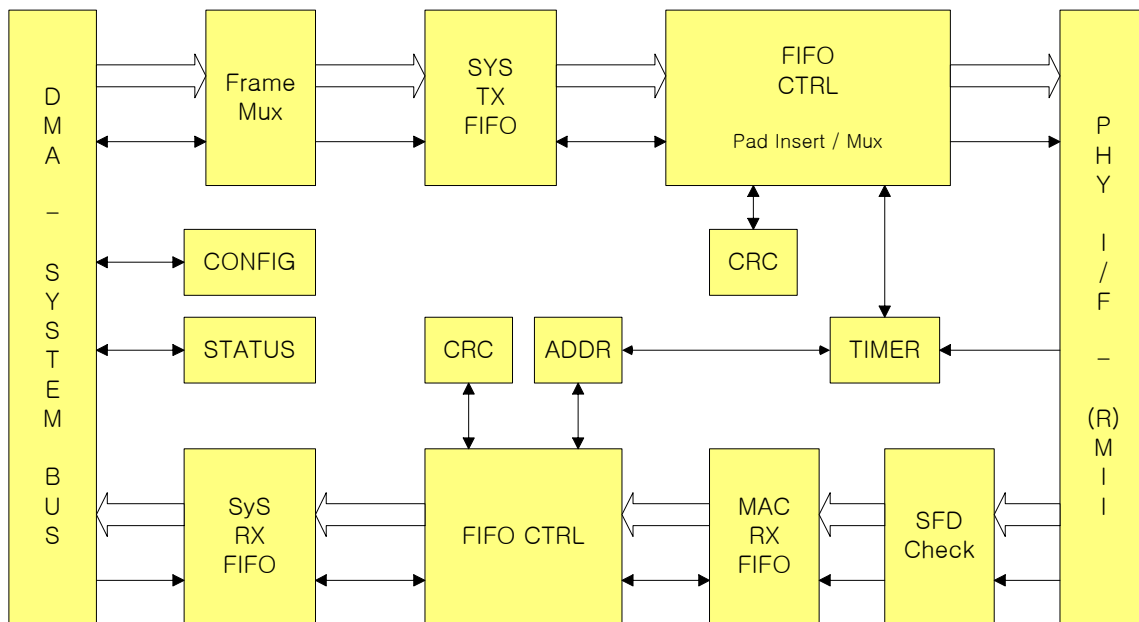
**Figure 2.9 Block Diagram of Ethernet MAC**

**Version 1.5**

Below Table show the PHY interface I/O signals

| Port | Width | Direction | Description |
|---|---|---|---|
| MTXCLK | 1 | I | Transmit Nibble or Symbol Clock. The PHY provides the MTXCLK signal. It operates at the frequency of 25MHz (100Mbps) or 2.5MHz (10Mbps). The clock is used as a timing reference for the transfer of MTXD[3:0], MTXEN, and MTXERR. |
| MTXD[3:0] | 4 | O | Transmit Data Nibble. Signals are the transmit data nibble. They are synchronized to the rising edge of MTXCLK. When MTXEN is asserted, PHY accepts the MTXD. If 7-wire interface is selected, MTXD[0] will carry data bits, all others will be tri-stated. |
| MTXEN | 1 | O | Transmit Enable. When asserted, signal indicates to the PHY that the data MTXD[3:0] is valid and the transmission can start. Then transmission starts with the first nibble of the preamble. Signal remains asserted until all nibbles to be transmitted are presented to the PHY. It is deasserted prior to the first MTXCLK following the final nibble of a frame. |
| MTXERR | 1 | O | Transmit Coding Error. When asserted for on e MTXCLK clock period while MTXEN is also asserted, it causes the PHY to transmit one or more symbols that are not part of the valid data or delimiter set somewhere in the frame being transmitted to indicate that there has been a transmit coding error. |
| MRXCLK | 1 | I | Receive Nibble or Symbol Clock. The PHY provides the MTXCLK signal. It operates at the frequency of 25MHz (100Mbps) or 2.5MHz (10Mbps). The clock is used as a timing reference for the reception of MRXD[3:0], MRXDV and MRXERR. |
| MRXDV | 1 | I | Receive Data Valid. The PHY asserts this signal to indicate to the RX MAC that it is presenting the valid nibbles on the MRXD[3:0] signals. Signal is asserted synchronously to the MRXCLK. MRXDV is asserted from the first recovered nibble of the frame to the final recovered nibble. It is then deasserted |

**Version 1.5**

| | | | prior to the first MRXCLK that follows the final nibble. |
|---|---|---|---|
| MRXD[3:0] | 4 | I | Receive Data Nibble. Signals are the receive data nibble. They are synchronized to the rising edge of MRXCLK. When MRXDV is asserted, the PHY sends a data nibble to the RX MAC. For a correctly interpreted frame, seven bytes of a preamble and a completely formed SFD must be passed across the interface. If 7-wire interface is selected, MRXD[0] will carry data bits, all others will be unused. |
| MRXERR | 1 | I | Receive Error. PHY asserts this signal to indicate to the RX MAC that a media error was detected during the transmission of the current frame. MRXERR is synchronous to the MRXCLK and is asserted for one or more MRXCLK clock periods and then deasserted. |
| MCOLL | 1 | I | Collision Detected. The PHY asynchronously asserts the collision signal MCOLL after the collision is detected on the media. When deasserted, no collision is detected on the media. |
| MCRS | 1 | I | Carrier Sense. The PHY asynchronously asserts the carrier sense MCRS signal after the medium is detected in a non-idle state. When deasserted, signal indicates that the media is in the idle state (and the transmission can start). |
| MDC | 1 | I | Management Data Clock. Clock for the MDIO serial data channel. |
| MDIO | 1 | I/O | Management Data Input/Output. Bi-directional serial data channel for PHY/STA communication. |

PHY Interface Signals

## 2.6.2. User Accessible Registers (Base = 0x1920_0000)

This section describes registers inside the Ethernet MAC. The address field in the following table indicates an address offset in hexadecimal from Ethernet Base Address defined in the CPU section. Width specifies the number of bits in the register and access specifies the valid access types of the register. Where 'RW' stands for read and write access, 'RO' for read only access. A 'C' indicates

that some or all of the bits are auto cleared. Base address of Ethernet MAC 1 is 0x1920_1000.

**Table 2.22 Registers for Ethernet MAC**

| Name | Address | Width | Access | Description |
|---|---|---|---|---|
| MAC_MODE | 0x00 | 32 | RW | MAC mode register |
| INT_SRC | 0x04 | 32 | RW | Interrupt source register |
| INT_ENABLE | 0x08 | 32 | RW | Interrupt enable register |
| IF_GAP | 0x0C | 32 | RW | Inter-frame gap register |
| COLL_CFG | 0x10 | 32 | RW | Collision control register |
| TX_BADDR | 0x14 | 32 | RW | Transmit buffer base address |
| TX_LENGTH | 0x18 | 32 | RW | Transmit buffer length |
| RX_BADDR | 0x1C | 32 | RW | Receive buffer base address |
| RX_BSTAT | 0x20 | 32 | RO | Receive buffer status |
| RX_BUFLVL | 0x24 | 32 | RO | Address/Status buffer level |
| RX_ADDR_BACK | 0x28 | 32 | RO | Receive buffer base address return |
| CTRL_MODE | 0x2C | 32 | RWC | Control mode register |
| MII_MODE | 0x30 | 32 | RW | MII mode register |
| MII_CMD | 0x34 | 32 | RWC | MII command register |
| MII_TXDATA | 0x38 | 32 | RW | MII transmit data |
| MII_RXDATA | 0x3C | 32 | RW | MII receive data |
| RESERVED | 0x40 | 32 | | Reserved |
| LENGTH | 0x44 | 32 | RW | Max, Min, burst length register |
| MCAST_ADDR_0 | 0x48 | 32 | RW | Multicast Address (Most significant) |
| MCAST_ADDR_1 | 0x4C | 32 | RW | Multicast Address (Least significant) |
| MAC_ADDR_00 | 0x50 | 32 | RW | MAC address 0 (Most significant 2 bytes) |
| MAC_ADDR_01 | 0x54 | 32 | RW | MAC address 0 (Least significant 4 bytes) |
| MAC_ADDR_10 | 0x58 | 32 | RW | MAC address 1 (Most significant 2 bytes) |
| MAC_ADDR_11 | 0x5C | 32 | RW | MAC address 1 (Least significant 4 bytes) |
| MAC_ADDR_20 | 0x60 | 32 | RW | MAC address 2 (Most significant 2 bytes) |
| MAC_ADDR_21 | 0x64 | 32 | RW | MAC address 2 (Least significant 4 bytes) |
| MAC_ADDR_30 | 0x68 | 32 | RW | MAC address 3 (Most significant 2 bytes) |

**Version 1.5**

| MAC_ADDR_31 | 0x6C | 32 | RW | MAC address 3 (Least significant 4 bytes) |
|---|---|---|---|---|
| MAC_ADDR_40 | 0x70 | 32 | RW | MAC address 4 (Most significant 2 bytes) |
| MAC_ADDR_41 | 0x74 | 32 | RW | MAC address 4 (Least significant 4 bytes) |
| MAC_ADDR_50 | 0x78 | 32 | RW | MAC address 5 (Most significant 2 bytes) |
| MAC_ADDR_51 | 0x7C | 32 | RW | MAC address 5 (Least significant 4 bytes) |
| MAC_ADDR_60 | 0x80 | 32 | RW | MAC address 6 (Most significant 2 bytes) |
| MAC_ADDR_61 | 0x84 | 32 | RW | MAC address 6 (Least significant 4 bytes) |
| MAC_ADDR_70 | 0x88 | 32 | RW | MAC address 7 (Most significant 2 bytes) |
| MAC_ADDR_71 | 0x8C | 32 | RW | MAC address 7 (Least significant 4 bytes) |
| P_FRM_ADDR_0 | 0x90 | 32 | RW | Pause frame address (Most significant 2 bytes) |
| P_FRM_ADDR_1 | 0x94 | 32 | RW | Pause frame address (Least significant 4 bytes) |
| P_FRM_ID | 0x98 | 32 | RW | Pause frame Type / Op. code |
| P_FRM_VALUE | 0x9C | 32 | RW | Pause frame delay value |
| H_TX_BADDR | 0xA0 | 32 | RW | High priority queue TX address |
| H_TX_LENGTH | 0xA4 | 32 | RW | High priority queue TX length |
| TX_BUF_LVL_L | 0xA8 | 32 | RO | TX queue buffer level |
| TX_ADDR_BK_L | 0xAC | 32 | RO | TX address return |
| TX_BUF_LVL_H | 0xB0 | 32 | RO | TX high priority queue buffer level |
| TX_ADDR_BK_H | 0xB4 | 32 | RO | TX high priority queue address back buffer level |
| RX_TIMER | 0xB8 | 32 | RW | RX timer register |
| RX_LEVEL | 0xBC | 32 | RW | RX level register |
| TX_TIMER | 0xC0 | 32 | RW | TX timer register |
| TX_LEVEL_H | 0xC4 | 32 | RW | TX level register High |
| TX_LEVEL_L | 0xC8 | 32 | RO | TX level register Low |

### 2.6.2.1. MAC_MODE (offset = 0x00)

This register defines general operation mode of the Ethernet MAC.

**Table 2.23 MAC Mode Register Bit Definition**

MAC0 Address：**1920_0000**

MAC1 Address：**1920_1000**

| Bits | Access | Default | Description |
|---|---|---|---|
| 31 | RW | 0 | UNICAST_ENABLE<br>0 = Disable receiving unicast frames<br>1 = Enable receiving unicast frames |
| 30 | RW | 0 | MULTICAST_ENABLE<br>0 = Disable receiving multicast frames<br>1 = Enable receiving multicast frames |
| 29 | RW | 0 | BROADCAST_ENABLE<br>0 = Disable receiving broadcast frames<br>1 = Enable receiving broadcast frames |
| 28 | RW | 0 | RECEIVE_ALL_ENABLE<br>0 = Frames will be received based on Bits [31:29].<br>1 = Enable receiving all frames.<br>When "1", if any MAC address is set to be valid, the frame with valid MAC address will be dropped (see valid bit of MAC address registers). |
| 27:19 | | 0 | Reserved |
| 18 | RW | 0 | TX Error signaling<br>0 = LOW output at TX_ER pin for error signaling<br>1 = HIGH output at TX_ER pin for error signaling |
| 17 | RW | 1 | TX data valid polarity<br>0 = active low data valid output to PHY<br>1 = active high data valid output to PHY |
| 16 | RW | 0 | TX_EN<br>0 = Disable TX after the current frame is transmitted<br>1 = Enable TX |
| 15:11 | | 0 | Reserved |
| 10 | RW | 1 | RX Error polarity<br>0 = active low error output to PHY<br>1 = active high error output to PHY |

| 9 | RW | 1 | RX data valid polarity |
|---|---|---|---|
|   |   |   | 0 = active low data valid input from PHY |
|   |   |   | 1 = active high data valid input from PHY |
| 8 | RW | 0 | RX_EN |
|   |   |   | 0 = Receive disabled after the current packet is received |
|   |   |   | 1 = Receive enabled |
| 7 | RW | 0 | Speed |
|   |   |   | 1 = 100Mbps |
|   |   |   | 0 = 10Mbps |
| 6 | RW | 0 | RMII Enable |
|   |   |   | 1 = RMII mode regardless of bit[1] |
|   |   |   | 0 = MII or 7-wire mode according to bit[1] |
| 5 | RW | 1 | Collision polarity |
|   |   |   | 0 = active low collision input from PHY |
|   |   |   | 1 = active high collision input from PHY |
| 4 | RW | 1 | Carrier Sense polarity |
|   |   |   | 0 = active low carrier sense input from PHY |
|   |   |   | 1 = active high carrier sense input from PHY |
| 3 | RW | 0 | Loop-back mode |
|   |   |   | 0 = normal operation |
|   |   |   | 1 = Enable MAC loop-back |
| 2 | RW | 0 | Full / Half Duplex mode |
|   |   |   | 0 = Half duplex mode |
|   |   |   | 1 = Full duplex mode |
| 1 | RW | 0 | Interface Select |
|   |   |   | 0 = 7-wire |
|   |   |   | 1 = MII |
| 0 | RWC | 0 | Reset MAC |
|   |   |   | 0 = Normal operation |
|   |   |   | 1 = Reset active (auto-clear after reset is done) |

## 2.6.2.2. INT_SRC (offset = 0x04)

The Ethernet MAC provides 30 interrupt sources.

**Table 2.24 Interrupt Source Bit Definition**

MAC0 Address：**1920_0004**

MAC1 Address：**1920_1004**

| Bits | Access | Default | Description (Write "1" to clear each bit) |
|---|---|---|---|
| 31 | RW | 0 | 1 = MII PHY interrupt |
| 30 | RW | 0 | 1 = MII Scan interrupt |
| 29 | RW | 0 | 1 = TX error at PHY |
| 28 | RW | 0 | 1 = RX error at PHY |
| 27 | RW | 0 | 1 = TX address return buffer overflow. |
| 26 | RW | 0 | 1 = TX address return buffer underflow |
| 25 | RW | 0 | 1 = Collision Detected |
| 24 | RW | 0 | 1 = Carrier Sense Detected |
| 23 | | 0 | Reserved. |
| 22 | RW | 0 | 1 = TX complete (high priority queue) |
| 21 | RW | 0 | 1 = Late collision detected |
| 20 | RW | 0 | 1 = TX completed (low priority queue) |
| 19 | RW | 0 | 1 = SYS TX FIFO overflow |
| 18 | RW | 0 | 1 = SYS TX FIFO underflow |
| 17 | RW | 0 | 1 = MAC TX FIFO overflow |
| 16 | RW | 0 | 1 = MAC TX FIFO underflow |
| 15 | RW | 0 | 1 = TX length buffer overflow (low priority queue) |
| 14 | RW | 0 | 1 = TX length buffer underflow (low priority queue) |
| 13 | RW | 0 | 1 = TX address buffer overflow (low priority queue) |
| 12 | RW | 0 | 1 = TX address buffer underflow (low priority queue) |
| 11 | RW | 0 | 1 = RX status buffer overflow |
| 10 | RW | 0 | 1 = RX status buffer underflow |
| 9 | RW | 0 | 1 = RX address buffer overflow |
| 8 | RW | 0 | 1 = RX address buffer underflow |

**Version 1.5**

| 7 |    | 0 | Reserved |
|---|----|---|----------|
| 6 | RW | 0 | 1 = RX frame cut due to bus busy & excessive incoming frames |
| 5 | RW | 0 | 1 = RX frame dropped due to no buffer assigned. |
| 4 | RW | 0 | 1 = RX completed. |
| 3 | RW | 0 | 1 = SYS RX FIFO overflow |
| 2 | RW | 0 | 1 = SYS RX FIFO underflow |
| 1 | RW | 0 | 1 = MAC RX FIFO overflow |
| 0 | RW | 0 | 1 = MAC RX FIFO underflow |

## 2.6.2.3. INT_ENABLE (offset = 0x08)

Each bit of this register can be set to enable corresponding interrupt source. Writing '0' disables the source from generating interrupt.

**Table 2.25 Interrupt Enable Bit Definition**

MAC0 **Address：1920_0008**

MAC1 **Address：1920_1008**

| Bits | Access | Default | Description |
|------|--------|---------|-------------|
| 31 | RW | 0 | MII PHY<br>0 = Interrupt disabled<br>1 = Interrupt enabled<br>This bit will be set whenever PHY device sends an interrupt |
| 30 | RW | 0 | MII Scan<br>0 = Interrupt disabled<br>1 = Interrupt enabled<br>This bit will be set when single read is completed in scan mode. Refer to MII_CMD register (offset = 0x34). |
| 29 | RW | 0 | TX error occurred at PHY<br>0 = Interrupt disabled<br>1 = Interrupt enabled |
| 28 | RW | 0 | RX error occurred at PHY |

| | | | 0 = Interrupt disabled |
|---|---|---|---|
| | | | 1 = Interrupt enabled. |
| 27 | RW | 0 | TX address return buffer overflow. |
| | | | 0 = Interrupt disabled |
| | | | 1 = Interrupt enabled. |
| 26 | RW | 0 | TX address return buffer underflow |
| | | | 0 = Interrupt disabled |
| | | | 1 = Interrupt enabled. |
| 25 | RW | 0 | Collision Detected |
| | | | 0 = Interrupt disabled |
| | | | 1 = Interrupt enabled |
| 24 | RW | 0 | Carrier Sense Detected |
| | | | 0 = Interrupt disabled |
| | | | 1 = Interrupt enabled |
| 23 | | 0 | Reserved. |
| 22 | RW | 0 | TX complete (high priority queue) |
| | | | 0 = Interrupt disabled |
| | | | 1 = Interrupt enabled |
| 21 | RW | 0 | Late collision detected (TX frame dropped) |
| | | | 0 = Interrupt disabled |
| | | | 1 = Interrupt enabled |
| 20 | RW | 0 | TX completed |
| | | | 0 = Interrupt disabled |
| | | | 1 = Interrupt enabled |
| 19 | RW | 0 | SYS TX FIFO overflow. |
| | | | 0 = Interrupt disabled |
| | | | 1 = Interrupt enabled |
| 18 | RW | 0 | SYS TX FIFO underflow. |
| | | | 0 = Interrupt disabled |
| | | | 1 = Interrupt enabled |
| 17 | RW | 0 | MAC TX FIFO overflow |
| | | | 0 = Interrupt disabled |
| | | | 1 = Interrupt enabled |

**Version 1.5**

| 16 | RW | 0 | MAC TX FIFO underflow. |
|----|----|---|------------------------|
|    |    |   | 0 = Interrupt disabled |
|    |    |   | 1 = Interrupt enabled |
| 15 | RW | 0 | TX length buffer overflow (low priority queue) |
|    |    |   | 0 = Interrupt disabled |
|    |    |   | 1 = Interrupt enabled |
| 14 | RW | 0 | TX length buffer underflow (low priority queue) |
|    |    |   | 0 = Interrupt disabled |
|    |    |   | 1 = Interrupt enabled |
| 13 | RW | 0 | TX address buffer overflow (low priority queue) |
|    |    |   | 0 = Interrupt disabled |
|    |    |   | 1 = Interrupt enabled |
| 12 | RW | 0 | TX address buffer underflow (low priority queue) |
|    |    |   | 0 = Interrupt disabled |
|    |    |   | 1 = Interrupt enabled |
| 11 | RW | 0 | Receive status buffer overflow |
|    |    |   | 0 = Interrupt disabled |
|    |    |   | 1 = Interrupt enabled |
| 10 | RW | 0 | Receive status buffer underflow |
|    |    |   | 0 = Interrupt disabled |
|    |    |   | 1 = Interrupt enabled |
| 9  | RW | 0 | Receive address buffer overflow |
|    |    |   | 0 = Interrupt disabled |
|    |    |   | 1 = Interrupt enabled |
| 8  | RW | 0 | Receive address buffer underflow |
|    |    |   | 0 = Interrupt disabled |
|    |    |   | 1 = Interrupt enabled |
| 7  |    | 0 | Reserved |
| 6  | RW | 0 | RX frame cut due to bus busy & excessive incoming frames |
|    |    |   | 0 = Interrupt disabled |
|    |    |   | 1 = Interrupt enabled |
| 5  | RW | 0 | Received frame is dropped due to no buffer. |
|    |    |   | 0 = Interrupt disabled |

| | | | 1 = Interrupt enabled |
|---|---|---|---|
| 4 | RW | 0 | RX completed<br>0 = Interrupt disabled<br>1 = Interrupt enabled |
| 3 | RW | 0 | SYS RX FIFO overflow.<br>0 = Interrupt disabled<br>1 = Interrupt enabled |
| 2 | RW | 0 | SYS RX FIFO underflow.<br>0 = Interrupt disabled<br>1 = Interrupt enabled |
| 1 | RW | 0 | MAC RX FIFO overflow<br>0 = Interrupt disabled<br>1 = Interrupt enabled |
| 0 | RW | 0 | MAC RX FIFO underflow.<br>0 = Interrupt disabled<br>1 = Interrupt enabled |

### 2.6.2.4. IF_GAP (offset = 0x0C)

This register defines a gap time between two back-to-back frames.

**Table 2.26 Inter-Frame Gap Register**

MAC0 Address：**1920_000C**
MAC1 Address：**1920_100C**

| Bits | Access | Default | Description |
|---|---|---|---|
| 31:7 | | 0 | Reserved |
| 6:0 | RW | 0x18 | Back to Back Inter Packet Gap<br>The recommended value is 0x0C (24 – 12 = 12) for MII, 0x54 (96 – 12 = 84) for 7-wire interface, which equals to 0.96us (100Mbps) or 9.6us (10Mbps). |

### 2.6.2.5. COLL_CFG (offset = 0x10)

This register defines collision related parameters.

**Table 2.27 Collision Configuration Definition**

MAC0 Address：**1920_0010**

MAC1 Address：**1920_1010**

| Bits | Access | Default | Description |
|------|--------|---------|-------------|
| 31:16 | RW | 0x000F | Maximum Retry<br>This field specifies the maximum number of consequential retransmission attempts after the collision is detected. When the maximum number is reached the MAC reports an error and tops transmitting the current packet. According to the Ethernet standard, the Maximum Retry default value is set to 0xF (15 in decimal) |
| 15:8 | RW | 0x20 | Write Allowance<br>Define the margin in system fifo to prevent overflow. |
| 7:0 | RW | 0x10 | Collision Window<br>Define the number of data from the start of transmission until collision is detected. If collision is detected before the number of data is reached this value, it will be considered as retry able collision. |

### 2.6.2.6. TX_BADDR (offset = 0x14)

The start address of outgoing frame is written in this register. Writing to TX_BADDR and TX_LENGTH will initiate transmission process.

**Table 2.28 Transmit Buffer Address**

MAC0 Address：**1920_0014**

MAC1 Address：**1920_1014**

| Bits | Access | Default | Description |
|------|--------|---------|-------------|
| 31:0 | RW | 0x0 | Transmit buffer start address |

### 2.6.2.7. TX_LENGTH (offset = 0x18)

The total frame length (starting from destination address to the end of frame, excluding CRC field) should be written to this register.

**Table 2.29 Transmit Buffer Length**

MAC0 Address：**1920_0018**

MAC1 Address：**1920_1018**

| Bits | Access | Default | Description |
|------|--------|---------|-------------|
| 31:16 | RW | 0x00 | Reserved |
| 15:0 | RW | 0x00 | Transmit buffer length in bytes |

### 2.6.2.8. RX_BADDR (offset = 0x1C)

The starting address of RX buffer to store incoming frame is written to this register. RX_BADDR can hold 15 entries of the start address of RX buffer. The number of entries not taken for RX processing can be read from bits[10:8] of RX_BUFLVL (offset = 0x24). Not assigning a buffer address in time will result in an interrupt (if enabled) with INT_SRC bit[5] and the frame will be dropped.

**Table 2.30 Receive Buffer Address**

MAC0 Address：**1920_001C**

MAC1 Address：**1920_101C**

| Bits | Access | Default | Description |
|------|--------|---------|-------------|
| 31:0 | RW | 0x00 | Receive buffer start address |

### 2.6.2.9. RX_BSTAT (offset = 0x20)

This address can hold 32 entries of the status of RX frames. The number of status of frames not read can be obtained from bits[5:0] of RX_BUFLVL (offset = 0x24).

**Table 2.31 Receive Frame Status**

MAC0 Address：**1920_0020**

MAC1 Address：**1920_1020**

| Bits | Access | Default | Description |
|------|--------|---------|-------------|
| 31:16 | RO | 0x0 | RX frame length |
| 15:3 | | 0x0 | Reserved |
| 2 | RO | 0x0 | Control frame |
| 1 | RO | 0x0 | CRC error |
| 0 | RO | 0x0 | Length error |

### 2.6.2.10. RX_BUFLVL (offset = 0x24)

This read only register provides buffer levels for both RX_BADDR and RX_BSTAT.

**Table 2.32 Receive Buffer Level**

MAC0 Address：**1920_0024**

MAC1 Address：**1920_1024**

| Bits | Access | Default | Description |
|------|--------|---------|-------------|
| 31:16 | | 0 | Reserved |
| 23:22 | | 0 | Reserved |
| 21:16 | RO | 0x0 | The number of buffer address taken for RX processing |
| 15:13 | | 0 | Reserved |
| 13:8 | RO | 0x0 | The number of buffer address not taken for RX processing |
| 7:6 | | 0 | Reserved |

**Version 1.5**

| 5:0 | RO | 0x0 | The number of RX_BSTAT available for read |
|-----|-----|------|--------------------------------------------|

## 2.6.2.11.  RX_ADDR_RETURN (offset = 0x28)

This read only register provides RX address used.

**Table 2.33 RX Address Return**

MAC0 **Address：1920_0028**

MAC1 **Address：1920_1028**

| Bits | Access | Default | Description |
|------|--------|---------|-------------|
| 31:0 | RO | 0x00 | RX address return |

## 2.6.2.12.  CTRL_MODE (offset = 0x2C)

This register defines two operation modes: Rx preamble and control frames. For transmit pause frame format, refer to P_FRM_ADDR_0, P_FRM_ADDR_1, P_FRM_ID, and P_FRM_VALUE starting at address offset 0x90.

**Table 2.34 Control Mode Register Bit Definition**

MAC0 **Address：1920_002C**

MAC1 **Address：1920_102C**

| Bits | Access | Default | Description |
|-------|--------|---------|-------------|
| 31:24 | RW | 0 | Reserved |
| 23:20 | RW | 0 | Reserved |
| 19:16 | RW | 0 | Minimum number of valid preambles for a valid RX frame<br>0 = No preamble byte: when SFD found, it is considered as a valid frame<br>1 = One preamble byte is required to be considered as a valid frame.<br>…<br>7 = Seven preamble bytes are required to be considered as a valid frame.<br>Set to 0 for RMII mode. |

**Version 1.5**

| | | | Preamble of RX packet should be byte alligned. All packets with any number of byte alligned preambles can be received as valid packets. |
|---|---|---|---|
| 15:8 | RW | 0 | Reserved |
| 7:5 | RW | 0 | Pause Frame Slot Request ( [7] & ([6] \| [5]) ) <br> Need more information to describe this field. |
| 4 | RW | 0 | Reserved |
| 3 | RW | 0 | TX_PAUSE Enable <br> 0 = Disable TX Pause frame <br> 1 = Enable TX Pause frame. |
| 2:0 | RW | 0 | Reserved |

## 2.6.2.13. MII_MODE (offset = 0x30)

MII_MODE register defines various operation modes of Media Independent Interface.

**Table 2.35 MII Mode Register Bit Definition**

MAC0 Address：**1920_0030**
MAC1 Address：**1920_1030**

| Bits | Access | Default | Description |
|---|---|---|---|
| 31:28 | | 0 | Reserved |
| 27:24 | RW | 0 | Additional output delay on TX_EN, TX_ER, and TX_D[3:0]. <br> 0x0 = 0ns <br> 0x1 = 1ns <br> …. <br> 0xE = 14ns <br> 0xF = 15ns |
| 23:20 | RW | 0xF | Inter-Transaction gap <br> This specifies time gap (in MDC cycles) between two consecutive MII transactions when SCAN mode. Ignored when any mode other than scan. <br> 0x0 = Reserved |

| | | | 0x1 = (number of gap cycle = 1) |
| | | | 0x2 = (number of gap cycles = 2) |
| | | | … |
| | | | 0xE = (number of gap cycles = 14), |
| | | | 0xF= (number of gap cycles = 15) |
| 19:18 | | 0 | Reserved |
| 17 | RW | 1 | Preamble<br>0 = No MII preamble will be used.<br>1 = 32-bit preamble will lead MII transaction. |
| 16 | RW | 1 | MDC Mode<br>0 = Gated MDC<br>1 = Continuous MDC |
| 15:8 | RW | 0x64 | Clock Divider<br>This field defines a host cock divider factor. The host clock can be divided by an even number, greater than 1. The default value is 0x64 (100 in decimal). |
| 7:1 | | 0 | Reserved |
| 0 | RWC | 0 | MII Reset<br>0 = Normal operation<br>1 = Reset active (auto-clear after reset is done) |

## 2.6.2.14.  MII_CMD (offset = 0x34)

This register defines command operation of MII

**Table 2.36 MII Command Register Definition**

MAC0 **Address：1920_0034**

MAC1 **Address：1920_1034**

| Bits | Access | Default | Description |
|------|--------|---------|-------------|
| 31:24 | | 0 | Reserved |
| 23:19 | | 0 | Reserved |

| 18 | RWC | 0 | READ<br>0 = no operation or read complete<br>1 = read operation is in progress<br>Setting '1' will initiate read operation. It will be auto cleared if read operation is completed. Data read from PHY will be stored at MII_RXDATA (offset = 0x3C) |
|---|---|---|---|
| 17 | RWC | 0 | WRITE<br>0 = no operation or write complete<br>1 = write operation is in progress<br>Setting '1' will initiate write operation. It will be auto cleared if write operation is completed. Transmit data should be written in MII_TXDATA (offset = 0x38) before setting this bit. |
| 16 | RW | 0 | Scan / setting '1' will initiate a scan operation<br>0 = no operation or scan complete<br>1 = SCAN operation is in progress<br>Scan operation will perform read operation throughout the entire range of PHY ID and register offsets. Interrupt will be generated (if enabled) whenever single read is completed. The first PHY ID and register offsets to start scan operation should be written in bits[12:8] and bits[4:0] before setting this bit. The CPU needs to clear this bit to stop scan operation. |
| 15:13 | | 0 | Reserved |
| 12:8 | RW | 0 | PHY address |
| 7:5 | | 0 | Reserved |
| 4:0 | RW | 0 | Register address in a PHY |

## 2.6.2.15. MII_TXDATA (offset = 0x38)

The CPU should write 16-bit TX data prior to send write command via MII_CMD register.

**Version 1.5**

**Table 2.37 MII Transmit Data Register**

MAC0 Address：**1920_0038**

MAC1 Address：**1920_1038**

| Bits | Access | Default | Description |
|------|--------|---------|-------------|
| 31:16 | | 0 | Reserved |
| 15:0 | RW | 0 | Data to be written to the PHY |

## 2.6.2.16.  MII_RXDATA (offset = 0x3C)

The CPU can read contents of registers from PHY by issuing read command via MII_CMD register.

**Table 2.38 MII Receive Data Register**

MAC0 Address：**1920_003C**

MAC1 Address：**1920_103C**

| Bits | Access | Default | Description |
|------|--------|---------|-------------|
| 31:16 | | 0 | Reserved |
| 15:0 | RW | 0 | Data received from PHY |

## 2.6.2.17.  LENGTH (offset = 0x44)

This register defines upper and lower bound of payload length of a MAC frame. It also defines maximum burst transfer length between Ethernet MAC and system memory.

**Table 2.39 Length Register**

MAC0 Address：**1920_0044**

MAC1 Address：**1920_1044**

| Bits | Access | Default | Description |
|------|--------|---------|-------------|
| 31:16 | RW | 0x05dc | Maximum payload length, default = 1500 in decimal |

| | | | Frames with payload of more than this value will be truncated. |
|---|---|---|---|
| 15:8 | RW | 0x2e | Minimum payload length, default = 46 in decimal |
| 7:0 | RW | 0x10 | Maximum burst transfer length in word on the system bus.<br>default = 16 in decimal |

## 2.6.2.18.  MCAST_ADDR_0 (offset = 0x48)

This register combined with MCAST_ADDR_1 (offset = 0x4C) provides 64-bit field to qualify multicast frame destined to this Ethernet MAC. Only one bit out of 64-bits should be set to determine valid multicast address. The bit position to be set is calculated by applying CRC over destination address field of incoming multicast frame.

**Table 2.40 Multicast Address (Most Significant)**

MAC0 **Address：1920_0048**

MAC1 **Address：1920_1048**

| Bits | Access | Default | Description |
|---|---|---|---|
| 31:0 | RW | 0 | Only one bit position should be set to determine that the current multicast frame is destined to this MAC |

## 2.6.2.19.  MCAST_ADDR_1 (offset = 0x4c)

**Table 2.41 Multicast Address (Least Significant)**

MAC0 **Address：1920_004C**

MAC1 **Address：1920_104C**

| Bits | Access | Default | Description |
|---|---|---|---|
| 31:0 | RW | 0 | See Multicast Address 0 (offset = 0x48) |

## 2.6.2.20.  MAC_ADDR_x0 (x = 0..7) (offset = 0x50, 0x58, …, 0x88)

This register along with the MAC_ADDR_x1 register forms a 6-byte address field of an Ethernet frame. It also provides byte ordering and valid bits for configuration. Total of eight MAC addresses are available.

**Table 2.42 MAC Address 0 (Control & Byte 5, 4)**

MAC0 Address：**1920_0050, 1920_0058, 1920_0060, 1920_0068, 1920_0070, 1920_0078, 1920_0080, 1920_0088**

MAC1 Address：**1920_1050, 1920_1058, 1920_1060, 1920_1068, 1920_1070, 1920_1078, 1920_1080, 1920_1088**

| Bits | Access | Default | Description |
|---|---|---|---|
| 31:24 | | 0 | Reserved |
| 23:21 | | 0 | Reserved |
| 20 | RW | 0 | Byte ordering<br>0 = Byte 5 is the most significant byte of the address<br>1 = Byte 0 is the most significant byte of the address |
| 19:17 | | 0 | Reserved |
| 16 | RW | 0 | VALID<br>0 = The address will not be compared with incoming destination address<br>1 = The address will be compared with incoming destination address |
| 15:8 | RW | 0 | Byte 5 of the Ethernet MAC address |
| 7:0 | RW | 0 | Byte 4 of the Ethernet MAC address |

## 2.6.2.21.  MAC_ADDR_x1 (x = 0..7) (offset = 0x54, 0x5C, …, 0x8C)

**Table 2.43 MAC Address 1 (Byte 3, 2, 1, 0)**

MAC0 Address：**1920_0054, 1920_005C, 1920_0064, 1920_006C, 1920_0074, 1920_007C, 1920_0084, 1920_008C**

MAC1 Address：**1920_1054, 1920_105C, 1920_1064, 1920_106C, 1920_1074, 1920_107C, 1920_1084,**

**1920_108C**

| Bits | Access | Default | Description |
|------|--------|---------|-------------|
| 31:24 | RW | 0 | Byte 3 of the Ethernet MAC address |
| 23:16 | RW | 0 | Byte 2 of the Ethernet MAC address |
| 15:8 | RW | 0 | Byte 1 of the Ethernet MAC address |
| 7:0 | RW | 0 | Byte 0 of the Ethernet MAC address |

## 2.6.2.22. P_FRM_ADDR_0 (offset = 0x90)

This register along with the P_FRM_ADDR 1 register (offset 0x94) forms a 6-byte address field of an Ethernet Pause frame. It also provides byte ordering and valid bits for configuration.

**Table 2.44 Pause Frame Address 0**

MAC0 Address：**1920_0090**

MAC1 Address：**1920_1090**

| Bits | Access | Default | Description |
|------|--------|---------|-------------|
| 31:24 | RW | 0 | Reserved |
| 23:21 | RW | 0 | Reserved |
| 20 | RW | 0 | Byte ordering<br>0 = Byte 5 is the most significant byte of the address<br>1 = Byte 0 is the most significant byte of the address |
| 19:17 | RW | 0 | Reserved |
| 16 | RW | 0 | VALID<br>0 = The address will not be compared with incoming destination address<br>1 = The address will be compared with incoming destination address |
| 15:8 | RW | 0x01 | Byte 5 of the Pause Frame Multicast address |
| 7:0 | RW | 0x80 | Byte 4 of the Pause Frame Multicast address |

## 2.6.2.23. P_FRM_ADDR_1 (offset = 0x94)

**Table 2.45 Pause Frame Address 1**

MAC0 Address：**1920_0094**

MAC1 Address：**1920_1094**

| Bits | Access | Default | Description |
|------|--------|---------|-------------|
| 31:24 | RW | 0xC2 | Byte 3 of the Pause Frame Multicast address |
| 23:16 | RW | 0x00 | Byte 2 of the Pause Frame Multicast address |
| 15:8 | RW | 0x01 | Byte 1 of the Pause Frame Multicast address |
| 7:0 | RW | 0x80 | Byte 0 of the Pause Frame Multicast address |

## 2.6.2.24. P_FRM_ID (offset = 0x98)

This register holds frame type identifier and operation code of Pause frame.

**Table 2.46 Pause Frame Type ID and OP Code**

MAC0 Address：**1920_0098**

MAC1 Address：**1920_1098**

| Bits | Access | Default | Description |
|------|--------|---------|-------------|
| 31:16 | RW | 0x8808 | Type identifier for pause frame |
| 15:0 | RW | 0x0001 | Operation code for pause frame |

## 2.6.2.25. P_FRM_VALUE (offset = 0x9C)

This register holds delay information of Pause frame. This field specifies the number of delay cycles to pause TX based on MAC_CLK, i.e., 25MHz for 100Mbps and 2.5 MHz for 10Mbps.

**Table 2.47 Pause frame delay value**

| Bits | Access | Default | Description |
|------|--------|---------|-------------|
| 31:16 | | 0 | Reserved |
| 15:0 | RW | 0x0018 | Delay value |

## 2.6.2.26. H_TX_BADDR (offset = 0xA0)

This register holds base address of a frame to be transmitted via high priority TX queue (single entry).

**Table 2.48 TX High Priority Queue Base Address**

| Bits | Access | Default | Description |
|------|--------|---------|-------------|
| 31:0 | RW | 0 | Base address of TX frame for high priority queue |

## 2.6.2.27. H_TX_LENGTH (offset = 0xA4)

This register holds length of a frame to be transmitted via high priority TX queue (single entry).

**Table 2.49 TX High Priority Queue Length**

| Bits | Access | Default | Description |
|------|--------|---------|-------------|
| 31:0 | R/W | 0 | Length of TX frame for high priority queue |

### 2.6.2.28. TX_BUF_LVL_L (offset = 0xA8)

This register displays buffer levels of TX address buffer, TX length buffer, and TX address-back buffer of low priority TX queue.

**Table 2.50 TX Low Priority Queue Level**

MAC0 Address：**1920_00A8**

MAC1 Address：**1920_10A8**

| Bits | Access | Default | Description |
|------|--------|---------|-------------|
| 31:21 | | 0 | Reserved |
| 20:16 | RO | 0 | TX address-back buffer level |
| 15:12 | | 0 | Reserved |
| 12:8 | RO | 0 | TX address buffer level |
| 7:5 | | 0 | Reserved |
| 4:0 | RO | 0 | TX length buffer level |

### 2.6.2.29. TX_ADDR_BK_L (offset = 0xAC)

This register holds up to 15 TX addresses of low priority used in transmission.

**Table 2.51 TX Low Priority Queue Address Return**

MAC0 Address：**1920_00AC**

MAC1 Address：**1920_10AC**

| Bits | Access | Default | Description |
|------|--------|---------|-------------|
| 31:0 | RO | 0 | TX address back |

### 2.6.2.30. TX_BUF_LVL_H (offset = 0xB0)

This register displays buffer levels of TX address buffer, TX length buffer, and TX address-back

buffer of high priority TX queue.

**Table 2.52 TX High Priority Queue Level**

MAC0 Address：**1920_00B0**

MAC1 Address：**1920_10B0**

| Bits | Access | Default | Description |
|---|---|---|---|
| 31:21 | | 0 | Reserved |
| 20:16 | RO | 0 | TX address-back buffer level |
| 15:12 | | 0 | Reserved |
| 12:8 | RO | 0 | TX address buffer level |
| 7:5 | | 0 | Reserved |
| 4:0 | RO | 0 | TX length buffer level |

### 2.6.2.31. TX_ADDR_BK_H (offset = 0xB4)

This register holds up to 15 TX addresses of high priority used in transmission.

**Table 2.53 TX High Priority Queue Address Return**

MAC0 Address：**1920_00B4**

MAC1 Address：**1920_10B4**

| Bits | Access | Default | Description |
|---|---|---|---|
| 31:0 | RO | 0 | TX address back |

# 2.7. UART

The HMS30C7110® integrates a serial interface which supports Universal Asynchronous Receiver / Transmitter (UART) function. It is compatible with the NS16550A.

The universal asynchronous receiver / transmitter supports independent 1 channel Tx/Rx functionality. The UART supports full duplex and it has 16 bytes internal FIFOs to overcome interrupt latency for receiving and sending data.

The serial information exchanged between host PC and serial device has a synchronization bit (start of frame), stop bit (end of frame) and error detection bit (parity bit). The supported parity is even, odd, forced one, and forced zero.

The UART is designed to clear the cumulative clock jitter between host PC and UART device. When the UART detects the start bit, the clock generator synchronize internal counter to the falling edge of start bit. For this reason, the maximum tolerable jitter rate is 5 % of BAUD rate.

The UART contains following features.
- Full duplex
- Receive FIFO of 16 bytes
- 6, 7 or 8 bits per character
- 1 or 2 stop bits
- Odd parity, even parity, forced parity or none
- Break detection and generation
- Parity, overrun and framing error detection
- Receiver timeout interrupt
- Programmable BAUD rate generator
- Two modem control signals
- Auto-flow control

## 2.7.1. Block Diagram
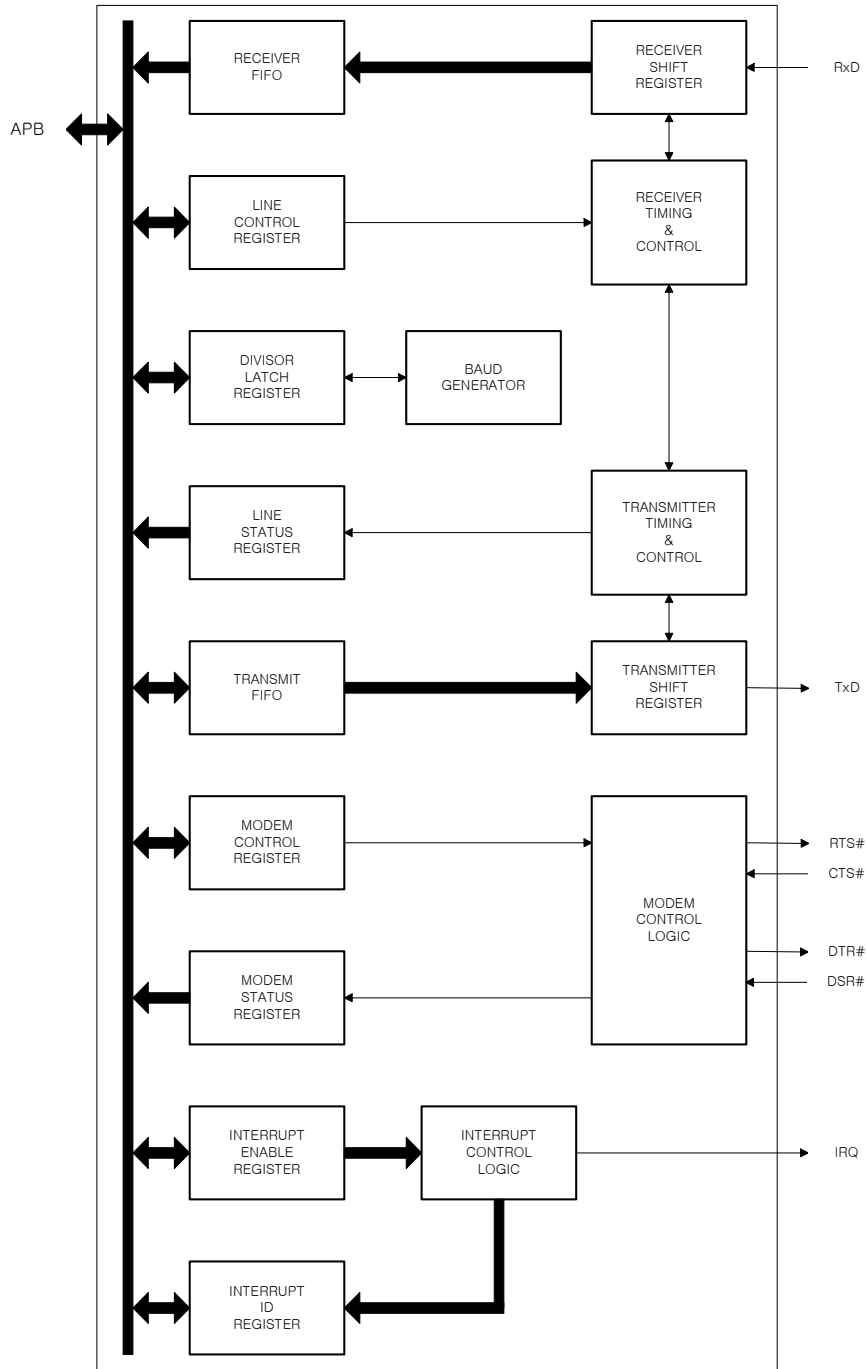


**Figure 2.10 Block Diagram of UART Device**

## 2.7.2.    User Accessible Registers (Base = 0x1800_0000)

This section describes all base, control and status register inside the UART. The address field indicates a relative address in hexadecimal. The base address of UART0 is 0x18000000 and UART1 is 0x18080000. Only UART1 has a hardware flow control. Width specifies the number of bits in the register and access specifies the valid access types that register. Where RW stands for read and write access, RO for read only access. A "C" appended to RW or RO, indicates that some or all of the bits can be cleared after a write '1' in corresponding bit. Base address for UART 1 is 0x1808_0000.

**Table 2.54 Registers for UART**

| Name | Address | Width | Access | Description |
|------|---------|-------|--------|-------------|
| RHR | 0x00 | 8 | RO | Receiver Holding Register |
| THR | 0x00 | 8 | WO | Transmitter Holding Register |
| IER | 0x04 | 4 | RW | Interrupt Enable Register |
| IIR | 0x08 | 4 | RO | Interrupt Identification Register |
| FCR | 0x08 | 2 | WO | FIFO Control Register |
| LCR | 0x0c | 7 | RW | Line Control Register |
| MCR | 0x10 | 4 | RW | MODEM Control Register |
| LSR | 0x14 | 8 | RO | Line Status Register |
| MSR | 0x18 | 8 | RO | MODEM Status Register |
| DLL | 0x00 | 8 | WO | Divisor Latch LSB Register |
| DLM | 0x04 | 8 | WO | Divisor Latch MSB Register |

*DLL and DLM are accessible ONLY when LCR bit-7 is set to "1"

### 2.7.2.1.    Receiver Holding Register (RHR)

This register holds the received incoming data byte. Bit 0 is the least significant bit, which is transmitted and received first. Received data is double buffered; this uses an additional shift register to receive the serial data stream and convert it to a parallel 8 bit word which is transferred to the Receive Buffer register. The shift register is not accessible.

**Table 2.55 RHR Bit Definition**

Address：**1800_0000**

| Bits | Access | Default | Description |
|------|--------|---------|-------------|
| 31:8 |  | 0x00 | Reserved. |
| 7: 0 | RO | 0x00 | RECEIVE DATA<br>This 8 bit field is loaded with receive data. The receive data is stored in FIFO. RHR has the byte data first loaded in the FIFO. |

## 2.7.2.2.　Transmitter Buffer Register (THR)

This register contains the data byte to be transmitted. The transmit buffer is double buffered, utilizing an additional shift register (not accessible) to convert the 8 bit data word to a serial format. This shift register is loaded from the Transmit Buffer when the transmission of the previous byte is complete.

**Table 2.56 THR Bit Definition**

Address：**1800_0000**

| Bits | Access | Default | Description |
|------|--------|---------|-------------|
| 31:8 |  | 0x00 | Reserved. |
| 7: 0 | WO | 0x00 | TRANSMIT DATA<br>This 8 bit field is loaded with transmit data. The transmit data is double buffered. THR is copied to transmit shift register to convert serial data. |

## 2.7.2.3.　Interrupt Enable Register (IER)

This register enables five types of interrupts for the associated serial channel. Each interrupt can activate the interrupt request signal. It is possible to totally disable the interrupt system by resetting bits 0 through 3 of the Interrupt Enable Register (IER). Similarly, setting bits of the IER register to logic 1, enables the selected interrupt. Disabling an interrupt prevents it from being indicated as active in the IIR and from activating IRQ signal.

**Version 1.5**

**Table 2.57 IER Bit Definition**

Address：**1800_0004**

| Bits | Access | Default | Description |
|---|---|---|---|
| 31:4 | | 0x00 | Reserved |
| 3 | RW | 0x0 | MODEM Status Interrupt Enable<br>When set to logic 1 this bit enables the MODEM Status Interrupt |
| 2 | RW | 0x0 | Receiver Line Status Interrupt Enable<br>When set to logic 1 this bit enables the Receiver Line Status Interrupt. |
| 1 | RW | 0x0 | Transmitter Holing Register Empty Interrupt Enable<br>When set to logic 1 this bit enables the Transmitter Holding Register Empty Interrupt. |
| 0 | RW | 0x0 | Receive Data Available Interrupt Enable<br>When set to logic 1 this bit enables the Receive Data Available Interrupt and Timeout Interrupt in the FIFO Mode. |

## 2.7.2.4. Interrupt Identification Register (IIR)

In order to provide minimum software overhead during data character transfers, each serial channel of the UART prioritizes interrupts into four levels and records these in the Interrupt Identification Register. The four levels of interrupt conditions in order of priority are Receiver Line Status; Received Data Ready; Transmitter Holding Register Empty; and MODEM Status.

**Table 2.58 IIR Bit Definition**

Address：**1800_0008**

| Bits | Access | Default | Description |
|---|---|---|---|
| 31:4 | | 0x00 | Reserved |
| 3 | RO | 0x0 | This bit is set along with bit 2 when a timeout interrupt is pending. |
| 2:1 | RO | 0x00 | These two bits of the IIR identify the highest priority interrupt pending from those shown in Table 2.64. |

**Version 1.5**

| 0 | RO | 0x0 | This bit can be used in a prioritized interrupt environment to indicate whether an interrupt is pending. When bit 0 is logic 0, an interrupt is pending and the IIR contents may be used as a pointer to the appropriate interrupt service routine. When bit 0 is logic 1, no interrupt is pending. |
|---|---|---|---|

**Table 2.59 Interrupt Control Functions**

| FIFO Mode Only | Interrupt Identification Register | | | Interrupt Set and Reset Functions | | | |
|---|---|---|---|---|---|---|---|
| Bit 3 | Bit 2 | Bit 1 | Bit 0 | Priority Level | Interrupt Type | Interrupt Source | Interrupt Reset Control |
| 0 | 0 | 0 | 1 | - | None | None | - |
| 0 | 1 | 1 | 0 | Highest | Receiver Line Status | Overrun Error or Parity Error or Framing Error or Break Interrupt | Reading the Line Status Register |
| 0 | 1 | 0 | 0 | Second | Received Data Available | Receiver Data Available or Trigger Level Reached | Reading the Receiver Buffer Register or the FIFO Drops below the Trigger Level |
| 1 | 1 | 0 | 0 | Second | Character Timeout Indication | No Characters Have Been Removed from or Input to the RCVR FIFO During the Last 4 Char. Times and There is at Least 1 Char. In it During This Time | Reading the Receiver Buffer Register |
| 0 | 0 | 1 | 0 | Third | Transmitter Holding Register Empty | Transmitter Holding Register Empty | Reading the IIR Register (if Source of Interrupt) or Writing into the Transmitter Holding Register |
| 0 | 0 | 0 | 0 | Fourth | MODEM Status | Clear To Send or Data Set Ready or Ring Indicator or Data Carrier Detect | Reading the MODEM Status Register |

**Version 1.5**

## 2.7.2.5. FIFO Control Register (FCR)

This is a write only register at the same location as the IIR (the IIR is a read only register). This register is used to enable the FIFOs, clear the FIFOs, set the RCVR FIFO trigger level, and select the type of DMA signaling.

**Table 2.60 FCR Bit Definition**

Address：**1800_0008**

| Bits | Access | Default | Description |
|------|--------|---------|-------------|
| 31:8 | | 0x00 | Reserved |
| 7:6 | WO | 0x00 | These two bits are used to designate the interrupt trigger level. When the number of bytes in the RCVR FIFO is equivalent to the designated interrupt trigger level, a Received Data Available Interrupt is activated. This interrupt must be enabled by setting IER0<br>0 = RCVR FIFO Trigger Level is 1<br>1 = RCVR FIFO Trigger Level is 4<br>2 = RCVR FIFO Trigger Level is 8<br>3 = RCVR FIFO Trigger Level is 14 |
| 5:3 | | 0x00 | Reserved |
| 2 | WO | 0x0 | Writing a 1 to FCR2 clears all bytes in the XMIT FIFO and resets its counter logic to 0. The shift register is not cleared. The 1 that is written to this bit position is self-clearing |
| 1 | WO | 0x0 | Writing a 1 to FCR1 clears all bytes in the RCVR FIFO and resets its counter logic to 0. The shift register is not cleared. The 1 that is written to this bit position is self-clearing. |
| 0 | | 0x0 | Reserved |

## 2.7.2.6. Line Control Register (LCR)

The system programmer specifies the format of the asynchronous data communications exchange and sets the Divisor Latch Access bit via the Line Control Register (LCR). This is a read and write register. Details on each bit follow:

**Table 2.61 LCR Bit Definition**

<div align="right">Address：**1800_000C**</div>

| Bits | Access | Default | Description |
|------|--------|---------|-------------|
| 31:8 |  | 0x00 | Reserved |
| 7 | RW | 0x0 | Divisor latch enable<br>0 = disable<br>1 = enable |
| 6 | RW | 0x0 | BREAK CONTROL<br>It causes a break condition to be transmitted to the receiving UART. When it is set to logic 1, the serial output is forced to the Spacing state (logic 0). The break is disabled by setting this bit to logic 0. |
| 5 | RW | 0x0 | PARITY ENABLE<br>When this bit is logic 1, a Parity bit is generated (transmit data) or checked (receive data) between the last data bit and Stop bit of the serial data. |
| 4:3 | RW | 0x00 | PARITY SELECT<br>0 = odd parity<br>1 = even parity<br>2 = forced one (Mark parity)<br>3 = forced zero (Space parity) |
| 2 | RW | 0x0 | STOP BIT LENGTH<br>This bit specifies the number of Stop bits transmitted or received serial character.<br>0 = 1 stop bit<br>1 = 2 stop bit |
| 1: 0 | RW | 0x00 | WORD LENGTH<br>These two bits specify the number of data bits in each transmitted or received serial character. The encoding of bits 0 and 1 is as follows:<br>0 = 5 bits / character<br>1 = 6 bits / character<br>2 = 7 bits / character<br>3 = 8 bits / character |

## 2.7.2.7. MODEM Control Register (MCR)

This register controls the interface with the MODEM or data set. Details on each bit follow:

**Table 2.62 MCR Bit Definition**

Address：**1800_0010**

| Bits | Access | Default | Description |
|------|--------|---------|-------------|
| 31:6 | | 0x00 | Reserved |
| 5 | RW | 0x0 | AUTOFLOW<br>When this bit is set, the auto-flow control is enabled. The auto-flow control can be configured by setting bit 1 and 5 of MCR as shown in Table 2.68. |
| 4 | RW | 0x0 | LOOPBACK Enable<br>This bit provides a local loop-back feature for diagnostic testing of the associated serial channel. |
| 3:2 | | 0x00 | Reserved |
| 1 | RW | 0x0 | RTS Control<br>This bit controls the Request to Send (RTS#) output. Bit 1 affects the RTS# output in a manner identical to that described below for bit 0. |
| 0 | RW | 0x0 | DTR Control<br>This bit controls the Data Terminal Ready (DTR#) output. When bit 0 is set to logic 1, the DTR# output is forced to a logic 0. When bit 0 is reset to logic 0, the DTR# output is forced to logic 1. |

**Table 2.63 Autoflow Control Configuration**

| Bit5 | Bit1 | Description |
|------|------|-------------|
| 1 | 1 | Auto-rts and auto-cts enabled |
| 1 | 0 | Auto-cts only enabled |
| 0 | X | Auto-rts and auto-cts disabled |

### 2.7.2.8.   Line Status Register (LSR)

The Register provides status information to the CPU concerning the data transfer. Details on each bit follow:

**Table 2.64 LSR Bit Definition**

<div align="right">Address：**1800_0014**</div>

| Bits | Access | Default | Description |
|------|--------|---------|-------------|
| 31:8 |        | 0x00    | Reserved |
| 7    | RO     | 0x0     | FIFO error<br>This bit is set when there is at least one parity error, framing error or break indication if the FIFO. It is cleared when the CPU reads the LSR, if there are no subsequent errors in the FIFO. |
| 6    | RO     | 0x0     | Transmitter Empty (TEMT)<br>This bit is the Transmitter Empty indicator. Bit 6 is set to logic 1 whenever the transmitter FIFO and shift register are both empty. |
| 5    | RO     | 0x0     | Transmitter Holding Register Empty (THRE)<br>This bit is the Transmitter Holding Register Empty indicator. this bit is set when the XMIT FIFO is empty; it is cleared when at least 1 byte is written to the XMIT FIFO. |
| 4    | RO     | 0x0     | Break Interrupt (BI)<br>This bit is the Break Interrupt indicator. Bit 4 is set to logic 1 whenever the received data input is held in the Spacing (logic 0) state for longer than a full word transmission time. The BI indicator is reset whenever the CPU reads the contents of the Line Status Register or when the next valid character is loaded into the Receiver FIFO. This error is associated with the particular character in the FIFO it applies to. This error is revealed to the CPU when its associated character is at the top of the FIFO. When break occurs only one zero character is loaded into the FIFO. |
| 3    | RO     | 0x0     | Framing Error (FE)<br>This bit is the Framing Error indicator. Bit 3 indicates that the |

| | | | received character did not have a valid Stop bit. The FE bit is set to logic 1 when the serial channel detects logic 0 during the first Stop bit time. The FE indicator is reset whenever the CPU reads the contents of the Line Status Register. This error is associated with the particular character in the FIFO it applies to. This error is revealed to the CPU when its associated character is at the top of the FIFO. |
|---|---|---|---|
| 2 | RO | 0x0 | Parity Error (PE)<br><br>This bit is the Parity Error (PE) indicator. Bit 2 indicates that the received data character does not have the correct even or odd parity. This error is associated with the particular character in the FIFO it applies to. This error is revealed to the CPU when its associated character is at the top of the FIFO. |
| 1 | RO | 0x0 | Overrun Error (OE)<br><br>This bit is the Overrun Error indicator. Bit 1 indicates that the next character received was transferred into the Receiver Buffer Register before the CPU could read the previously received character. It is reset whenever the CPU reads the contents of the Line Status Register. |
| 0 | RO | 0x0 | Data Ready (DR)<br><br>This bit is the receive Data Ready indicator. Bit 0 is set to logic 1 whenever a complete incoming character has been received and transferred into the Receiver FIFO. Bit 0 is reset to logic 0 by reading all of the data in the Receive FIFO. |

## 2.7.2.9.    MODEM Status Register (MSR)

This register provides the current stat of the control lines from the MODEM. Details on each bit follow:

**Table 2.65 MSR Bit Definition**

Address：**1800_0018**

| Bits | Access | Default | Description |
|---|---|---|---|

| 31:6 | | 0x00 | Reserved |
|---|---|---|---|
| 5 | RO | 0x0 | DSR#<br>This bit is the complement of the Data Set Ready input. In the loopback mode, this bit is equivalent to DTR in MCR. |
| 4 | RO | 0x0 | CTS#<br>This bit is the complement of the Clear to Send input. In the loopback mode, this bit is equivalent to RTS in MCR. |
| 3:2 | | 0x00 | Reserved |
| 1 | RO | 0x0 | DDSR<br>This bit is the Delta Data Set Ready indicator. It indicates that the DSR# input to the chip has changed state since the last time it was read by the CPU. |
| 0 | RO | 0x0 | DCTS<br>This bit is the Delta Clear to Send indicator. It indicates that the CTS# input to the chip has changed stat since the last time it was read by the CPU. |

## 2.7.2.10.  Divisor Latch LSB Register (DLL)

The UART contains a programmable Baud Generator. The output frequency of the Baud Generator is $16 \times$ the divisor number. The output of Baud Generator drives the transmitter and receiver sections of the associated serial channel. This register contains the lower byte of the divisor number. DLL is accessible only when LCR bit-7 is set to 1.

**Table 2.66 DLL Bit Definition**

Address：**1800_0000**

| Bits | Access | Default | Description |
|---|---|---|---|
| 31:8 | | 0x00 | Reserved. |
| 7: 0 | WO | 0x00 | Lower byte of Divisor number |

## 2.7.2.11.  Divisor Latch MSB Register (DLM)

This register contains the higher byte of the divisor number. DLM is accessible only when LCR bit-

7 is set to 1.

**Table 2.67 DLM Bit Definition**

Address：**1800_0004**

| Bits | Access | Default | Description |
|------|--------|---------|-------------|
| 31:8 |        | 0x00    | Reserved. |
| 7: 0 | WO     | 0x00    | Higher byte of Divisor number |

*Baud rate = clock frequency ÷ (16 × (Divisor# + 1))

# 2.8. TIMER

The HMS30C7110® integrates 3 channels of Timers and each channel has 32-bit down counter. The input clock of each channel is selectable among one of the 4 clock speeds, System Bus Clock/1, System Bus Clock/4, System Bus Clock/16, or System Bus Clock/256. When the clock of the system bus is 100MHz, the Timer input options are 100/25/6.25/0.4MHz. When the clock of the system is 80MHz, the Timer input options are 80/20/5/0.3125MHz. The Timers are fully configurable through a set of control registers. Complete descriptions of these registers are given in the Register Section.

## 2.8.1. User Accessible Registers (Base = 0x1810_0000)

This section describes all base, control and status registers inside the Timer. The address field indicates a relative address in hexadecimal. Width specifies the number of bits in the register and access specifies the valid access types that register. Where 'RW' stands for read and write access, 'RO' for read only access. A 'C' appended to 'RW' or 'RO', indicates that some or all of the bits can be cleared after a write '1' in corresponding bit.

**Table 2.68 Registers for TIMER**

| Name | Address | Width | Access | Description |
|------|---------|-------|--------|-------------|
| Clock Selection | 0x00 | 3x2 | RW | Input clock selection for 3 channels |
| Timer Control | 0x04 | 3x2 | RW | Channel enable/disable |
| Timer Interval 0 | 0x08 | 32 | RW | Reload value for Timer Channel 0 |
| Timer Interval 1 | 0x0c | 32 | RW | Reload value for Timer Channel 1 |
| Timer Interval 2 | 0x10 | 32 | RW | Reload value for Timer Channel 2 |
| Timer Value 0 | 0x14 | 32 | RW | Channel 0 current timer value |
| Timer Value 1 | 0x18 | 32 | RW | Channel 1 current timer value |
| Timer Value 2 | 0x1c | 32 | RW | Channel 2 current timer value |
| INT SRC | 0x20 | 3 | RW | Interrupt pending register |
| INT ENABLE | 0x24 | 3 | RW | Interrupt enable register |

## 2.8.1.1. Clock Selection

This register contains clock division values (pre-scalar) for 3 channels.

**Table 2.69 Clock Selection Register Bit Definition**

Address：**1810_0000**

| Bits | Access | Default | Description |
|---|---|---|---|
| 31:10 | | | Reserved |
| 9: 8 | RW | 0 | Clock Select for Timer 2<br>0 = System Bus Clock<br>1 = System Bus Clock ÷ 4<br>2 = System Bus Clock ÷ 8<br>3 = System Bus Clock ÷ 256 |
| 7: 6 | | | Reserved |
| 5: 4 | RW | 0 | Clock Select for Timer 1<br>0 = System Bus Clock<br>1 = System Bus Clock ÷ 4<br>2 = System Bus Clock ÷ 8<br>3 = System Bus Clock ÷ 256 |
| 3: 2 | | | Reserved |
| 1: 0 | RW | 0 | Clock Select for Timer 0<br>0 = System Bus Clock<br>1 = System Bus Clock ÷ 4<br>2 = System Bus Clock ÷ 8<br>3 = System Bus Clock ÷ 256 |

## 2.8.1.2. Timer Control

This register includes timer enable and clear bits for 3 channels.

**Version 1.5**

**Table 2.70 Timer Control Register Bit Definition**

<div align="right">

Address：**1810_0004**
</div>

| Bits | Access | Default | Description |
|---|---|---|---|
| 31:10 | | | Reserved |
| 9 | W | 0 | Timer 2 clear<br>Writing '1' to this bit clears the channel 2 counter. |
| 8 | RW | 0 | Timer 2 enable<br>0 = Timer2 disable<br>1 = Timer2 enable |
| 7: 6 | | | Reserved |
| 5 | W | 0 | Timer 1 clear<br>Writing '1' to this bit clears the channel 1 counter. |
| 4 | RW | 0 | Timer 1 enable<br>0 = Timer1 disable<br>1 = Timer1 enable |
| 3: 2 | | | Reserved |
| 1 | W | 0 | Timer 0 Clear<br>Writing '1' to this bit clears the channel 0 counter. |
| 0 | RW | 0 | Timer 0 enable<br>0 = Timer0 disable<br>1 = Timer0 enable |

### 2.8.1.3. Timer Interval 0~2

These registers include timer values.

**Table 2.71 Timer Interval Register Bit Definition**

<div align="right">

Address：**1810_0008, 1810_000C, 1810_0010**
</div>

| Bits | Access | Default | Description |
|---|---|---|---|
| 31: 0 | RW | 0 | INTERVAL<br>This 32-bit field contains the interval value that is loaded to the down |

**Version 1.5**

| | | | counter to recount. |
|---|---|---|---|

### 2.8.1.4. Timer Value 0~2

These registers show the current values of the down counter.

**Table 2.72 Current Timer Value Register Bit Definition**

Address：**1810_0014, 1810_0018, 1810_001C**

| Bits | Access | Default | Description |
|---|---|---|---|
| 31: 0 | RO | 0 | CTV (Current Timer Value)<br>This 32-bit field indicates the present value of internal down counter. |

### 2.8.1.5. INT SRC

This register has 3 timer interrupts that occur every time the counter reaches zero.

**Table 2.73 Interrupt Source Register Bit Definition**

Address：**1810_0020**

| Bits | Access | Default | Description |
|---|---|---|---|
| 31: 3 | | 0 | Reserved |
| 2 | RW | 0 | Expire timer 2<br>This bit indicates that the channel 2 down counter reaches zero value. |
| 1 | RW | 0 | Expire timer 1<br>This bit indicates that the channel 1 down counter reaches zero value. |
| 0 | RW | 0 | Expire timer 0<br>This bit indicates that the channel 0 down counter reaches zero value. |

### 2.8.1.6. INT ENABLE

This register includes interrupt enable bits.

**Table 2.74 Interrupt Enable**

Address：**1810_0024**

| Bits | Access | Default | Description |
|------|--------|---------|-------------|
| 31: 3 | | | Reserved |
| 2 | RW | 0 | Expire timer 2 enable |
| 1 | RW | 0 | Expire timer 1 enable |
| 0 | RW | 0 | Expire timer 0 enable |

**Version 1.5**

# 2.9. GPIO

The HMS30C7110® integrates a GPIO module that supports 21 GPIO (General Purpose Input / Output) ports. 3 GPIO out of 21 are dedicated to GPIO feature and others are muxed to other features. Each port is configurable with input or output mode. The initial modes of all GPIO pins are set to input, so when to use in output mode, it is highly recommended to pull-up or pull-down the ports to prevent annoying vibration caused by temporary high impedance state during reset and initialization.

The GPIO is capable of interrupting the MCU when input signal triggers, and it supports one of the four kinds of interrupt source signal.  Level trigger high/low and edge trigger rising/falling signal. The GPIO is fully configurable through a set of control registers. Complete descriptions of these registers are given in the Register Section.

## 2.9.1.    User Accessible Registers (Base = 0x1820_0000)

This section describes all base, control and status registers inside the GPIO module. The address field indicates a relative address in hexadecimal. Width specifies the number of bits in the register and access specifies the valid access types to that register. 'RW' stands for read and write access and 'RO' for read only access. A 'C' appended to 'RW' or 'RO' indicates some or all of the bits can be cleared after writing '1' to the corresponding bit.

**Table 2.75 Registers for GPIO**

| Name | Address | Width | Access | Description |
|---|---|---|---|---|
| GPO Data | 0x00 | 9 | RW | GPO data to I/O pins |
| GPI Data | 0x04 | 12 | RO | GPI data from I/O pins |
| GPIO Direction | 0x08 | 9 | RW | Direction for I/O pins |
| INT SRC | 0x0c | 16 | RW | Interrupt Source |
| INT ENABLE | 0x10 | 16 | RW | Interrupt enable |
| INT MODE | 0x14 | 16 | RW | Detection mode of interrupt |

| INT LEVEL | 0x18 | 16 | RW | Detection level of interrupt |
|-----------|------|----|----|-----------------------------|

### 2.9.1.1. GPO Data

The GPO data register has GPO output data that is put on the external pins. This value can be read also. Each GPO has nothing to do with same numbered GPI, which means independent operation.

**Table 2.76 GPO Data Register Bit Definition**

Address：**1820_0000**

| Bits | Access | Default | Description |
|------|--------|---------|-------------|
| 31: 9 | | | Reserved |
| 8: 0 | RW | 0 | GPO DATA<br>This 9-bit field is output to the GPO pins and the GPIOs when set to output mode. |

### 2.9.1.2. GPI Data

The GPI data register shows the GPI input data on the GPIO pins.

**Table 2.77 GPI Data Register Bit Definition**

Address：**1820_0004**

| Bits | Access | Default | Description |
|------|--------|---------|-------------|
| 31:12 | | | Reserved |
| 11: 0 | RO | 0 | GPI DATA<br>This 12-bit field shows the status of GPI pins and the stautus of GPIO pins regardless the direction of GPIOs or the mode of GPI, dedicated mode or GPI mode. |

### 2.9.1.3. GPIO Direction

The GPIO direction register controls the direction of GPIO pins and chooses the function of GPIs and GPOs.

**Table 2.78 GPIO Direction Register Bit Definition**

Address：**1820_0008**

| Bits | Access | Default | Description |
|------|--------|---------|-------------|
| 31:9 |  | 0x00 | Reserved |
| 8:3 | RW | 0x00 | GPO FUNCTION<br>These pins set the coressponding pins to either dedicated function or GPO.<br>0 = Use for specific features<br>1 = Use for GPI or GPO |
| 2:0 | RW | 0x00 | GPIO DIRECTION<br>This 3-bits field controls the direction of GPIO pads.<br>0 = input mode<br>1 = output mode |

## 2.9.1.4. INT SRC

The Interrupt source register indicates which input pin is triggered.

**Table 2.79 Interrupt Source Register Bit Definition**

Address：**1820_000C**

| Bits | Access | Default | Description |
|------|--------|---------|-------------|
| 31: 16 |  | 0x00 | Reserved |
| 15: 0 | ROC | 0x00 | INT SOURCE<br>This 16-bits field indicates which GPI pin(s) generated interrupt(s).<br>Bit 0 through 7 generate external interrupt 0 through 7 respectively.<br>Bit 8 through 15 generate GPIO interrupt. |

## 2.9.1.5. INT ENABLE

The Interrupt enable register chooses the interrupt sources being used.

**Table 2.80 The Bit Definition of the Interrupt Enable Register**

Address：**1820_0010**

| Bits | Access | Default | Description |
|---|---|---|---|
| 31: 16 | | 0x00 | Reserved |
| 15: 0 | RW | 0x00 | INT ENABLE<br>This 16-bits field enables the GPI interrupts.<br>0 = disable<br>1 = enable |

## 2.9.1.6. INT MODE

The interrupt mode register chooses the interrupt mode of either level trigger or edge trigger.

**Table 2.81 Interrupt Mode Register Bit Definition**

Address：**1820_0014**

| Bits | Access | Default | Description |
|---|---|---|---|
| 31: 16 | | 0x00 | Reserved |
| 15: 0 | RW | 0x00 | INT MODE<br>This 16-bits field controls the interrupt mode of each GPI pins.<br>0 = Level trigger<br>1 = Edge trigger |

## 2.9.1.7. INT LEVEL

The interrupt level register selects active level of interrupt sources when in level triggering mode, active low or active high.

**Table 2.82 Interrupt Level Register Bit Definition**

Address：**1820_0018**

| Bits | Access | Default | Description |
|---|---|---|---|

| 31:16 |    | 0x00 | Reserved |
|-------|----|------|----------|
| 15: 0 | RW | 0x00 | INT LEVEL<br><br>This 16-bits field controls the active level of interrupt request on each GPI pins.<br><br>0 = Low active for Level / Falling edge for Edge trigger<br><br>1 = High active for Level / Rising edge for Edge trigger |

# 2.10.  SPI

The HMS30C7110® integrates an Serial Peripheral Interface (SPI) module which is used for interfacing the Serial ROM or Serial Interface Devices such as PCM Audio codec.

The SPI of HMS30C7110® generates serial clock (CCLK), serial data out (CDOUT), serial data in (CDIN) and chip select (nCCS). The clock speed is selectable by programming the internal register. The SPI also supports the 8bit, 16bit and 32bit IO devices. The complete descriptions of these registers are given in the Register Section.

## 2.10.1.  Block Diagram

The SPI of HMS30C7110® consists of several blocks such as register file and serial interface core module.

**Version 1.5**
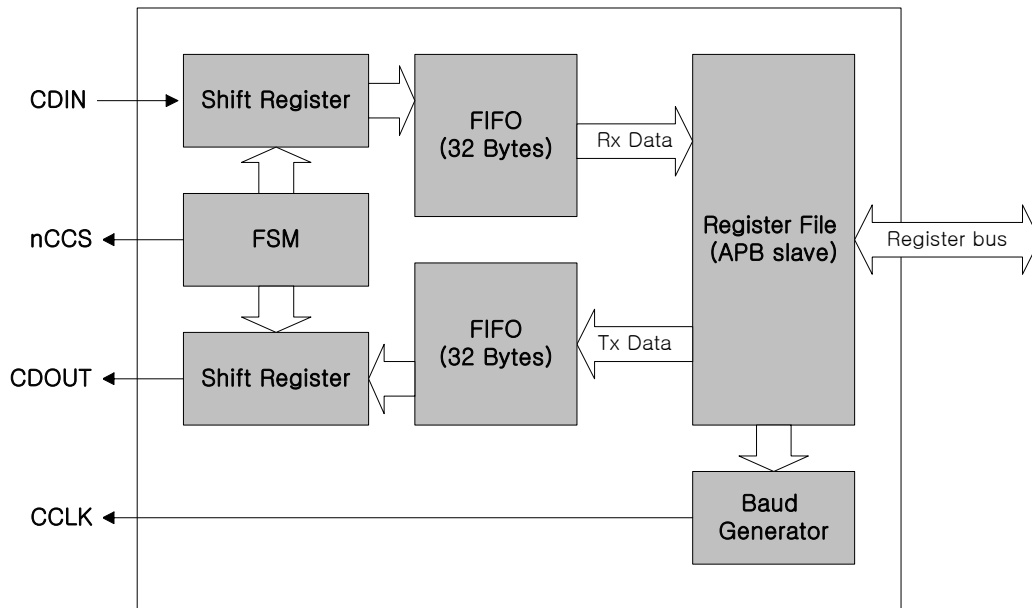
**Figure 2.11 Block Diagram of SPI**

## 2.10.2.   User Accessible Registers (Base = 0x1840_0000)

This section describes all base, control and status register inside SPI. The address field indicates a relative address in hexadecimal. The base address is 0x18500000. Width specifies the number of bits in the register and access specifies the valid access types that register. Where RW stands for read and write access, RO for read only access. A "C" appended to RW or RO, indicates that some or all of the bits can be cleared after a write '1' in corresponding bit.

**Table 2.83 Registers for SPI**

| Name | Address | Width | Access | Description |
|---|---|---|---|---|
| SPI Control | 0x00 | 9 | RW | SPI baud rate and enable control |
| SPI Mode0 | 0x04 | 8 | RW | Recovery timing control & Tx start |
| Reserved | 0x08 | | | Reserved |
| Reserved | 0x0c | | | Reserved |
| Reserved | 0x10 | | | Reserved |
| Reserved | 0x14 | | | Reserved |
| TX Data | 0x18 | 8 | RW | Transmit Data |
| RX Data | 0x1c | 8 | RW | Receive Data |
| INT SRC | 0x20 | 1 | ROC | Interrupt source |
| INT ENABLE | 0x24 | 1 | RW | Interrupt enable |
| STATUS | 0x28 | 1 | R | Rx Empty |

**Version 1.5**

### 2.10.2.1. SPI Control (offset = 0x00)

This register includes SPI enable bit and baud rate setting value.

**Table 2.84 SPI Control Register Bit Definition**

Address：**1840_0000**

| Bits | Access | Default | Description |
|---|---|---|---|
| 31:9 |  | 0 | Reserved |
| 8 | RW | 0 | Enable<br>0 = SPI clock disable<br>1 = SPI clock enable |
| 7:0 | RW | 0 | Baud rate<br>One clock period = (baudrate+1) × CLKIN |

### 2.10.2.2. SPI Mode 0 (offset = 0x04)

These registers include recovery cycle field and data width field.

**Table 2.85 Configuration Register Bit Definition**

Address：**1840_0004**

| Bits | Access | Default | Description |
|---|---|---|---|
| 31:12 |  | 0 | Reserved |
| 7:4 | RW | 0 | Recovery time<br>This 4-bit field is used for making recovery time from one access cycle to the next access cycle. |
| 3:1 |  | 0 | Reserved |
| 0 | RW | 0 | Tx Start<br>This one bit field is used to start transfer of Tx FIFO data. |

### 2.10.2.3. TX Data (offset = 0x18)

This register is writing path to 32 bytes Tx FIFO.

**Table 2.86 Tx Data Register Bit Definition**

Address：**1840_0018**

| Bits | Access | Default | Description |
|------|--------|---------|-------------|
| 31:8 |        | 0       | Reserved    |
| 7:0  | RW     | 0       | Tx Data     |

### 2.10.2.4.  RX Data (offset = 0x20)

This register is reading path of 32 bytes Rx FIFO.

**Table 2.87 Rx Data Register Bit Definition**

Address：**1840_001C**

| Bits | Access | Default | Description |
|------|--------|---------|-------------|
| 31:8 |        | 0       | Reserved    |
| 7: 0 | RW     | 0       | Rx Data     |

### 2.10.2.5.  INT SRC

This register includes SPI interrupt source bits.

**Table 2.88 Interrupt Source Register Bit Definition**

Address：**1840_0020**

| Bits | Access | Default | Description |
|------|--------|---------|-------------|
| 31: 1 |       | 0       | Reserved    |
| 0    | RW     | 0       | TX done     |

### 2.10.2.6.  INT ENABLE

This register includes SPI interrupt enable bits.

**Version 1.5**

**Table 2.89 Interrupt Mask Register Bit Definition**

Address：1840_0024

| Bits | Access | Default | Description |
|------|--------|---------|-------------|
| 31:1 |        | 0       | Reserved    |
| 0    | RW     | 0       | TX done enable |

### 2.10.2.7.  STATUS

This register includes SPI Rx Status bits.

**Table 2.90 SPI Status Register Bit Definition**

Address：1840_0028

| Bits | Access | Default | Description |
|------|--------|---------|-------------|
| 31: 1 |       | 0       | Reserved    |
| 0    | R      | 0       | TX empty    |

# 2.11.  DMA

HMS30C7110® supports two-channel DMA controller that is located between the system bus and the peripheral bus. Each channel of DMA controller can perform data movements between devices in the system bus and/or peripheral bus with no restrictions. In other words, each channel can handle the following four cases: 1) both source and destination are in the system bus, 2) source is in the system bus while destination is in the peripheral bus, 3) sources in the peripheral bus while destination is in the system bus, 4) both source and destination are in the peripheral bus.

The main advantage of DMA is that it can transfer data without any CPU intervention. The operation of DMA can be initiated by S/W, the request from internal peripherals or the external request pins.

## 2.11.1.   User Accessible Registers (Base = 0x1910_0000)

There are seven control registers for each DMA channel (Since there are two channels, the total number of control registers is 14). Four of them are to control the DMA transfer, and other three are to see the status of DMA controller. The details of those registers for a channel are as follows.

### 2.11.1.1.  DMA INITIAL SOURCE REGISTER (DISRC, offset = 0x00, 0x20)

**Table 2.91 DMA Initial Source Register**

DMA0 **Address**：**1910_0000**

DMA1 **Address**：**1910_0020**

| Field Name | Bit | Description |
|---|---|---|
| LOC | [31] | Bit 31 is used to select the location of source. 0: the source is in the system bus, 1: the source is in the peripheral bus |
| INC | [30] | Bit 30 is used to select the address increment. Increment unit is determined by DS of DCON. 0 = Increment 1= Fixed |

**Version 1.5**

| S_ADDR | [29:0] | These bits are the base address (start address) of source data to transfer |
|--------|--------|---------------------------------------------------------------------------|

## 2.11.1.2. DMA INITIAL DESTINATION REGISTER (DIDST, offset = 0x04, 0x24)

**Table 2.92 DMA Initial Destination Register**

DMA0 **Address** : **1910_0004**

DMA1 **Address** : **1910_0024**

| Field Name | Bit | Description |
|------------|-----|-------------|
| LOC | [31] | Bit 31 is used to select the location of destination.<br>0: the destination is in the system bus,<br>1: the destination is in the peripheral bus |
| INC | [30] | Bit 30 is used to select the address increment.<br>0 = Increment 1= Fixed |
| S_ADDR | [29:0] | These bits are the base address (start address) of destination to transfer |

## 2.11.1.3. DMA CONTROL REGISTER (DCON, offset = 0x08, 0x28)

**Table 2.93 DMA Control Register**

DMA0 **Address** : **1910_0008**

DMA1 **Address** : **1910_0028**

| Field Name | Bit | Description |
|------------|-----|-------------|
| DMD_HS | [30] | Select one between demand mode and handshake mode.<br>0 : demand mode is selected<br>1 : handshake mode is selected. |
| SYNC. | [29] | Select DREQ/DACK synchronization<br>0: DREQ and DACK are synchronized to PCLK.<br>1: DREQ and DACK are synchronized to HCLK. |
| INT | [28] | Enable/Disable the interrupt setting for TC(terminal count)<br>0: TC interrupt is disabled.<br>1: Interrupt is generated when all the transfer is done. |

**Version 1.5**

| TS | [27] | Select the transfer size of a transfer<br>0: a single transfer is performed.<br>1: a burst transfer of length four is performed. |
|---|---|---|
| SERVMODE | [26] | Select the service mode between single service mode and whole service mode<br>0: single service mode<br>1: whole service mode |
| HWSRCSEL | [25:24] | Select DMA request source for each DMA. |
| SWHW_SEL | [23] | Select the DMA source between software (S/W request mode) and hardware (H/W request mode).<br>0: S/W request mode<br>1: H/W request mode |
| RELOAD | [22] | Set the reload on/off option.<br>0: auto reload is performed.<br>1: channel is turned off after TC reaches. |
| DS | [21:20] | Data size to be transferred.<br>0: 8-bit (Byte), 1 : 16-bit (Half-Word), 2: 32-bit (Word) |
| TC | [19:0] | Initial transfer count (or transfer beat).<br>Note that the actual # of bytes that are transfered is obtained bt the following equation: DSZ * TSZ * TB, where DSZ, TSZ, and TB represent data size, transfer size, and transfer beats, respectively. |

## 2.11.1.4. DMA STATUS REGISTER (DSTAT, offset = 0x0c, 0x2c)

This register shows the current value of transfer counter(TC) and whether this channel is busy or idle.

**Table 2.94 DMA Status Register**

DMA0 Address：**1910_000C**
DMA1 Address：**1910_002C**

| Field Name | Bit | Description |
|---|---|---|

| RSVD | [30:22] | Reserved |
|---|---|---|
| STATUS | [21:20] | DMA Status. 0: Ready(or Idle), 1: Busy, 2: Error |
| CTC | [19:0] | Current transfer count (or transfer beat). |

## 2.11.1.5.  DMA CURRENT SOURCE REGISTER (DCSRC, offset = 0x10, 0x30)

This register shows the value of current source address of the transfer. So only least 30 bits are valid.

DMA0 **Address：1910_0010**

DMA1 **Address：1910_0030**

| Field Name | Bit | Description |
|---|---|---|
| | [29:00] | Current Source Address |

## 2.11.1.6.  CURRENT DESTINATION REGISTER (DCDST, offset = 0x14, 0x34)

This register shows the value of current destination address of the transfer.

DMA0 **Address：1910_0014**

DMA1 **Address：1910_0034**

| Field Name | Bit | Description |
|---|---|---|
| | [29:00] | Current Destination Address |

## 2.11.1.7.  DMA MASK TRIGGER REGISTER (MASKTRIG, offset = 0x18, 0x38)

**Table 2.95 DMA Mask Trigger Register**

DMA0 **Address：1910_0018**

DMA1 **Address：1910_0038**

| Field Name | Bit | Description |
|---|---|---|

| STOP | [2] | Stop the DMA operation.<br><br>1: DMA stops as soon as the current atomic transfer ends.<br><br>Note : If there is no current atomic transfer, DMA stops immediately. Due to possible current atomic transfer, "stop" may take several cycles. The finish of "stop" operation can be detected by waiting until the channel on/off bit is set to off. This stop is actual "stop". It means that if DMA starts again all the values including TC start from initial values. |
|---|---|---|
| ON_OFF | [1] | DMA channel on/off bit.<br><br>0: DMA channel is turned off. (DMA request to this channel is ignored.)<br><br>1: DMA channel is turned on and the DMA request is handled.<br><br>Note : This bit is automatically set to off if we set the control register (DCON) to "no auto reload" and/or this register (MASKTRIG) to "stop". Note that when "no auto reload" is set, this bit becomes "off" when TC becomes 0. On the other hand, if "stop" is set, this bit becomes "off" as soon as the current automic transfer finishes. This bit should not be changed manually during DMA operation. |
| SW_TRIG | [0] | Trigger the DMA channel in S/W.<br><br>1: it requests a DMA operation to this controller. |

## 2.12. INTC

The interrupt controller of HMS30C7110® receives the request from 21 interrupt sources. Among 21, 8 come from outside of chip, i.e., external interrupt sources, and remaining 13 come from internal sources such as DMA controller, ENET MAC, and etc.

The role of the interrupt controller is to ask for IRQ and/or NMI to the ARM7TDMI after selecting one among 21 sources based on the arbitration rule. The arbitration is performed by the hardware priority logic and the result is written to the interrupt pending register to let users to know which interrupt has been requested.

**Interrupt Mode**

INTC supports 2 types of interrupt modes, NMI and IRQ. The mode of each interrupt source can be set by programming interrupt mode register. Among 21 sources, only one, generally very urgent one such as power failure can be set to operate as NMI.

**Interrupt Pending Register**

There are two interrupt pending registers. One is source pending register (SRCPND) and the other is interrupt pending register (INTPND). These pending registers indicate whether or not an interrupt request is pending. When the interrupt sources request interrupt services the corresponding bits of SRCPND register are set to 1, and at the next clock cycle one bit of INTPND register corresponding to the highest priority one is set to 1 after the arbitration process. If some of interrupts are masked (by programming mask register), the corresponding bits of INTPND register are not set to 1, while the corresponding bits of SRCPND register are set to 1. When a pending bit of INTPND register is set, IRQ is generated and goes to CPU. The SRCPND and INTPND registers can be read and written. The interrupt service routine must clear the pending condition by writing a 1 to the corresponding bit of SRCPND register first and then clear the pending condition in INTPND registers by writing 1.

The NMI shares the SRCPND register with IRQ, but it does not go through the arbitration logic and directly goes to CPU for a fast service. So, in the service routine of NMI, user needs to clear only the SRCPND register.

**Interrupt Mask Register**

This register is used to mask the requests of some interrupt sources. If a bit of the register is set to 1, it indicates that the corresponding interrupt is disabled. If it is set to 0, the corresponding interrupt will be serviced normally. This mask register actually resides after the SRCPND register and affects the effective value of SRCPND register going to arbitration logic and NMI generation logic. Therefore, even for the masked requests, SRCPND register reflects their arrival as usual.

**Interrupt Sources**

Interrupt controller supports 21 interrupt sources as shown in the below table.

**Table 2.96 Interrupt Source Register**

| Sources | Bit | Descriptions | Arbiter Group |
|---|---|---|---|
| INT_UART1 | 29 | UART1 interrupt | ARB5 |
| INT_UART0 | 28 | UART0 interrupt | ARB5 |
| INT_SPI | 27 | SPI interrupt | ARB4 |
| Reserved | 26 | Reserved | ARB4 |
| CardBus | 22 | CardBus error interrupt | ARB4 |
| CARD IRQ | 21 | IRQ from installed Card | ARB3 |
| PCMCIA | 20 | PCMCIA interrupt including card detection | ARB3 |
| INT_GPIO | 19 | GPIO interrupt (from GPI8 ~ GPI10) | ARB3 |
| INT_DMA1 | 18 | DMA channel 1 interrupt | ARB3 |
| INT_DMA0 | 17 | DMA channel 0 interrupt | ARB3 |
| INT_ENET1 | 14 | ENET MAC 1 interrupt | ARB2 |
| INT_ENET0 | 11 | ENET MAC 0 interrupt | ARB2 |
| INT_TIMER | 10 | TIMER interrupt | ARB2 |
| INT_EXT7 | 7 | External interrupt 7 (from GPI7) | ARB1 |
| INT_EXT6 | 6 | External interrupt 6 (from GPI6) | ARB1 |
| INT_EXT5 | 5 | External interrupt 5 (from GPI5) | ARB1 |
| INT_EXT4 | 4 | External interrupt 4 (from GPI4) | ARB1 |
| INT_EXT3 | 3 | External interrupt 3 (from GPI3) | ARB0 |
| INT_EXT2 | 2 | External interrupt 2 (from GPIO2) | ARB0 |
| INT_EXT1 | 1 | External interrupt 1 (from GPIO1) | ARB0 |

| INT_EXT0 | 0 | External interrupt 0 (from GPIO0) | ARB0 |

**Arbitration block**

The priority logic for 32 interrupt requests is composed of seven rotation based arbiters: six first-level arbiters and one second-level arbiter as shown in the following figure.
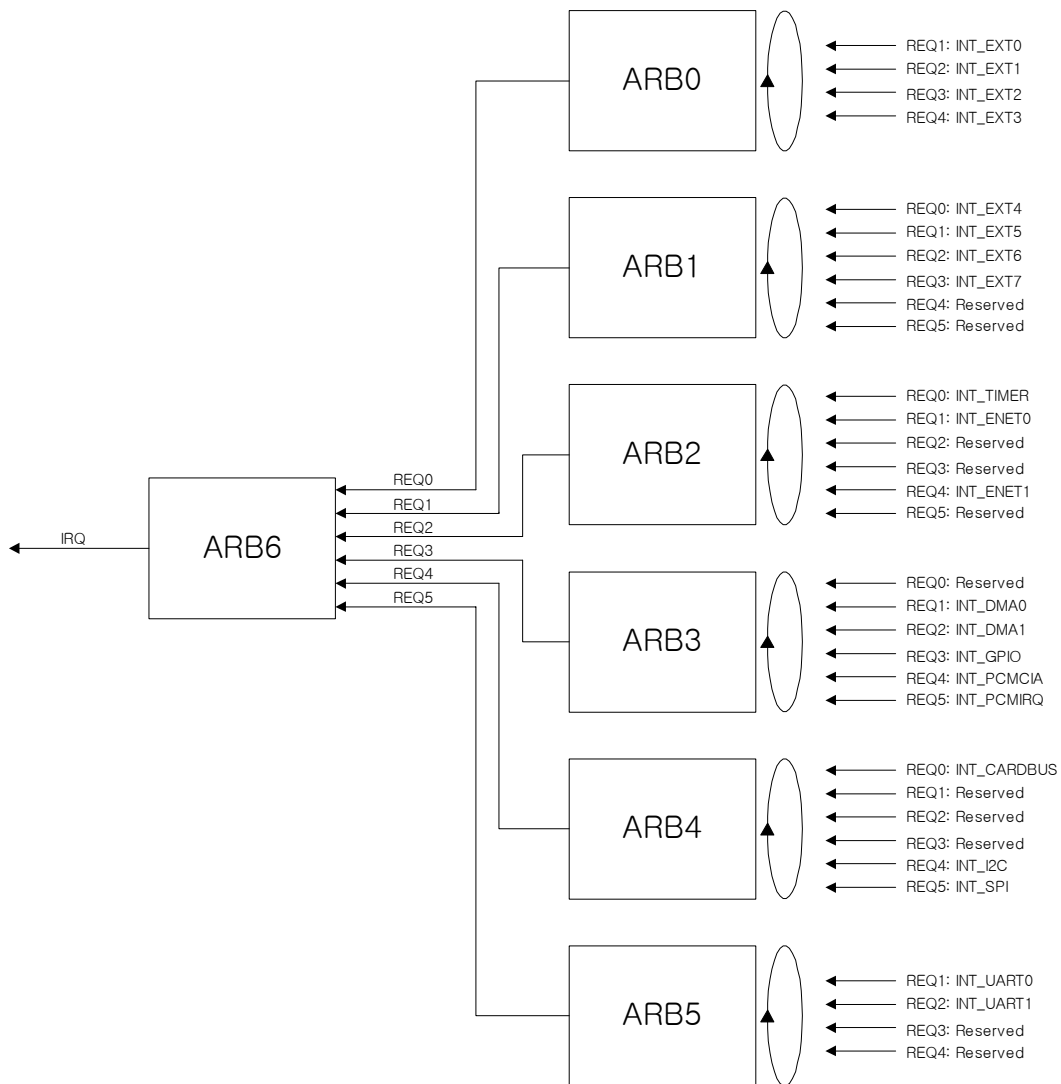


**Figure 2.12 Arbitration Block Diagram**

Each arbiter can handle six interrupt requests based on the one bit arbiter mode control

(ARB_MODE) and two bits of selection control signals (ARB_SEL) as follows:

➤ If ARB_SEL bits are 00b, the priority order is REQ0, REQ1, REQ2, REQ3, REQ4, and REQ5.

➤ If ARB_SEL bits are 01b, the priority order is REQ0, REQ2, REQ3, REQ4, REQ1, and REQ5.

➤ If ARB_SEL bits are 10b, the priority order is REQ0, REQ3, REQ4, REQ1, REQ2, and REQ5.

➤ If ARB_SEL bits are 11b, the priority order is REQ0, REQ4, REQ1, REQ2, REQ3, and REQ5.


Note that REQ0 of an arbiter is always the highest priority, and REQ5 is the lowest one. In addition, by changing the ARB_SEL bits, we can rotate the priority of REQ1 - REQ4.

Here, if ARB_MODE bit is set to 0, ARB_SEL bits are not automatically changed, thus the arbiter operates in the fixed priority mode. (Note that even in this mode, we can change the priority by manually changing the ARB_SEL bits.). On the other hand, if ARB_MODE bit is 1, ARB_SEL bits are changed in rotation fashion, e.g., if REQ1 is serviced, ARB_SEL bits are changed to 01b automatically so as to make REQ1 the lowest priority one. The detailed rule of ARB_SEL change is as follows.

➤ If REQ0 or REQ5 is serviced, ARB_SEL bits are not changed at all.

➤ If REQ1 is serviced, ARB_SEL bits are changed to 01b.

➤ If REQ2 is serviced, ARB_SEL bits are changed to 10b.

➤ If REQ3 is serviced, ARB_SEL bits are changed to 11b.

➤ If REQ4 is serviced, ARB_SEL bits are changed to 00b.


## 2.12.1.   User Accessible Registers (Base = 0x1930_0000)

There are five control registers in the interrupt controller: source pending register, interrupt mode register, mask register, priority register, and interrupt pending register.

All the interrupt requests from the interrupt sources are first registered in the source pending register. They are divided into two groups based on the interrupt mode register, i.e., one NMI request and the remaining IRQ requests. Arbitration process is performed for the multiple IRQ requests based on the priority register.

### 2.12.1.1.  SOURCE PENDING REGISTER (SRCPND, offset = 0x00)

SRCPND register is composed of 32 bits each of which is related to an interrupt source. Each bit is

set to 1 if the corresponding interrupt source generates the interrupt request and waits for the interrupt to be serviced. By reading this register, we can see the interrupt sources waiting for their requests to be serviced. Note that each bit of SRCPND register is automatically set by the interrupt sources regardless of the masking bits in the INTMASK register. In addition, it is not affected by the priority logic of interrupt controller.

In the interrupt service routine for a specific interrupt source, the corresponding bit of SRCPND register has to be cleared to get the interrupt request from the same source correctly. If you return from the ISR (interrupt service routine) without clearing the bit, interrupt controller operates as if another interrupt request comes in from the same source. In other words, if a specific bit of SRCPND register is set to 1, it is always considered as a valid interrupt request waiting to be serviced.

The specific time to clear the corresponding bit depends on the user's requirement. The bottom line is that if you want to receive another valid request from the same source you should clear the corresponding bit first, and then enable the interrupt.

You can clear a specific bit of SRCPND register by writing a data to this register. It clears only the bit positions of SRCPND corresponding to those set to one in the data. The bit positions corresponding to those that are set to 0 in the data remains as they are with no change.

**Table 2.97 Source Pending Register**

Address：**1930_0000**

| Register | Address | R/W | Description | value |
|----------|---------|-----|-------------|-------|
| SRCPND | 0x00 | R/W | 0 = The interrupt has not been requested<br>1 = The interrupt source has asserted the interrupt request | 0x00000000 |

| Bit | Field | Bit | Field | Bit | Field | Bit | Field |
|-----|-------|-----|-------|-----|-------|-----|-------|
| 31 | Reserved | 23 | Reserved | 15 | Reserved | 7 | INT_EXT7 |
| 30 | Reserved | 22 | INT_CARDBUS | 14 | INT_ENET1 | 6 | INT_EXT6 |
| 29 | INT_UART1 | 21 | INT_PCMIRQ | 13 | Reserved | 5 | INT_EXT5 |
| 28 | INT_UART0 | 20 | INT_PCMCIA | 12 | Reserved | 4 | INT_EXT4 |
| 27 | INT_SPI | 19 | INT_GPIO | 11 | INT_ENET0 | 3 | INT_EXT3 |
| 26 | Reserved | 18 | INT_DMA1 | 10 | INT_TIMER | 2 | INT_EXT2 |
| 25 | Reserved | 17 | INT_DMA0 | 9 | Reserved | 1 | INT_EXT1 |

**Version 1.5**

| 24 | Reserved | 16 | Reserved | 8 | Reserved | 0 | INT_EXT0 |
|----|----------|----|----------|---|----------|---|----------|

## 2.12.1.2.  MODE REGISTER (INTMOD, offset = 0x04)

This register is composed of 32 bits each of which is related to an interrupt source. If a specific bit is set to 1, the corresponding interrupt is processed as the NMI request. Otherwise, it is processed as a normal IRQ.

Note that at most only one interrupt source can be serviced in the NMI mode. (You should use the NMI mode only for the urgent interrupt.) Thus, only one bit of INTMOD can be set to 1 at most.

This register is write-only one, thus it cannot be read out.

**Table 2.98 Interrupt Mode Register**

Address：**1930_0004**

| Register | Address | R/W | Description | Reset value |
|----------|---------|-----|-------------|-------------|
| INTMOD | 0x04 | W | 0 = IRQ mode, 1 = NMI mode | 0x00000000 |

Each field and the corresponding bit position are the same as those of SRCPND register.

## 2.12.1.3.  INTERRUPT MASK REGISTER (INTMSK, offset = 0x08)

Each of the 32 bits in the interrupt mask register is related to an interrupt source. If you set a specific bit to 1, the interrupt request from corresponding interrupt source is not serviced by CPU. (Note that even in such a case, the corresponding bit of SRCPND register is set to 1). If the mask bit is 0, the interrupt request can be serviced.

**Table 2.99 Interrupt Mask Register**

Address：**1930_0008**

| Register | Address | R/W | Description | Reset value |
|----------|---------|-----|-------------|-------------|
| INTMSK | 0x08 | R/W | 0 = Interrupt service is available | 0xffffffff |

| | | | 1 = Interrupt service is masked | |
|---|---|---|---|---|

Each field and the corresponding bit position are same to those of SRCPND register.

## 2.12.1.4. PRIORITY REGISTER (PRIORITY, offset = 0x0c)

**Table 2.100 Priority Register**

Address：**1930_000C**

| Register | Address | R/W | Description | Reset value |
|---|---|---|---|---|
| PRIORITY | 0x0c | W | IRQ priority control register | 0x07f |

| Field | Bit | Description | Reset value |
|---|---|---|---|
| ARB_SEL6 | 20:19 | Arbiter 6 priority<br>00 = REQ 0-1-2-3-4-5 01 = REQ 0-2-3-4-1-5<br>10 = REQ 0-3-4-1-2-5 11 = REQ 0-4-1-2-3-5 | 00 |
| ARB_SEL5 | 18:17 | Arbiter 5 priority<br>00 = REQ 1-2-3-4 01 = REQ 2-3-4-1<br>10 = REQ 3-4-1-2 11 = REQ 4-1-2-3 | 00 |
| ARB_SEL4 | 16:15 | Arbiter 4 priority<br>00 = REQ 0-1-2-3-4-5 01 = REQ 0-2-3-4-1-5<br>10 = REQ 0-3-4-1-2-5 11 = REQ 0-4-1-2-3-5 | 00 |
| ARB_SEL3 | 14:13 | Arbiter 3 priority<br>00 = REQ 0-1-2-3-4-5 01 = REQ 0-2-3-4-1-5<br>10 = REQ 0-3-4-1-2-5 11 = REQ 0-4-1-2-3-5 | 00 |
| ARB_SEL2 | 12:11 | Arbiter 2 priority<br>00 = REQ 0-1-2-3-4-5 01 = REQ 0-2-3-4-1-5<br>10 = REQ 0-3-4-1-2-5 11 = REQ 0-4-1-2-3-5 | 00 |
| ARB_SEL1 | 10:9 | Arbiter 1 priority<br>00 = REQ 0-1-2-3-4-5 01 = REQ 0-2-3-4-1-5<br>10 = REQ 0-3-4-1-2-5 11 = REQ 0-4-1-2-3-5 | 00 |
| ARB_SEL0 | 8:7 | Arbiter 0 priority | 00 |

| | | 00 = REQ 1-2-3-4 01 = REQ 2-3-4-1<br>10 = REQ 3-4-1-2 11 = REQ 4-1-2-3 | |
| ARB_MODE6 | 6 | Arbiter 6 rotate enable<br>0 = Priority does not rotate, 1 = Priority rotate enable | 1 |
| ARB_MODE5 | 5 | Arbiter 5 rotate enable<br>0 = Priority does not rotate, 1 = Priority rotate enable | 1 |
| ARB_MODE4 | 4 | Arbiter 4 rotate enable<br>0 = Priority does not rotate, 1 = Priority rotate enable | 1 |
| ARB_MODE3 | 3 | Arbiter 3 rotate enable<br>0 = Priority does not rotate, 1 = Priority rotate enable | 1 |
| ARB_MODE2 | 2 | Arbiter 2 rotate enable<br>0 = Priority does not rotate, 1 = Priority rotate enable | 1 |
| ARB_MODE1 | 1 | Arbiter 1 rotate enable<br>0 = Priority does not rotate, 1 = Priority rotate enable | 1 |
| ARB_MODE0 | 0 | Arbiter 0 rotate enable<br>0 = Priority does not rotate, 1 = Priority rotate enable | 1 |

## 2.12.1.5.  INTERRUPT PENDING REGISTER (INTPND, offset = 0x10)

Each of the 32 bits in the interrupt pending register shows whether the corresponding interrupt request is the highest priority one that is unmasked and waits for the interrupt to be serviced. Since INTPND is located after the priority logic, only one bit can be set to 1 at most, and that is the very interrupt request generating IRQ to CPU. In interrupt service routine for IRQ, you can read this register to determine the interrupt source to be serviced among 23 sources.

Like the SRCPND, this register has to be cleared in the interrupt service routine. We can clear a specific bit of INTPND register by writing a data to this register. It clears only the bit positions of INTPND corresponding to those set to one in the data. The bit positions corresponding to those that are set to 0 in the data remains as they are with no change.

**Table 2.101 Interrupt Pending Register**

Address：**1930_0010**

| Register | Address | R/W | Description | Reset value |
|---|---|---|---|---|
| INTPND | 0x10 | R/W | 0 = The interrupt has not been requested | 0x00000000 |
| | | | 1 = The interrupt source has asserted a request | |

Each field and the corresponding bit position are the same as those of SRCPND register.

### 2.12.1.6. INTERRUPT OFFSET REGISTER (INTOFS, offset = 0x14)

If INTPND[n] is set to 1 (due to interrupt requests from sources), number "n" is shown in this register. This is for the ease of an interrupt target address computation, i.e., target address = INTOFS << 2.

Note that the valid value of this register is 0 – 31 when an interrupt occurs. If it is 32, it means that there happens no interrupts, i.e., INTPND = 0.

**Table 2.102 Interrupt Offset Register**

Address：**1930_0014**

| Register | Address | R/W | Description | Reset value |
|---|---|---|---|---|
| INTOFS | 0x14 | R | "n" in 0 – 31 if INTPND[n] is set to 1. | 0x00000020 |
| | | | 32 if there happens no interrupt. | |

# 2.13. PCMCIA Controller

The HMS30C7110® integrates a PCMCIA controller which supports 16-bits PC card and 32-bits PC card (Cardbus) functionalities.

The PCMCIA Controller supports 1 socket 16-bits PC Card or 1 socket CardBus card.

## 2.13.1. User Accessible Registers (Base = 0x1940_0000)

This section describes all base, control and status register inside the PCMCIA. The address field indicates a relative address in hexadecimal. The base address for 16-bits PC Card is 0x19400000 and the base address for CARDBUS is 0x19480000. Width specifies the number of bits in the register and access specifies the valid access types that register. Where RW stands for read and write access, RO for read only access. A "C" appended to RW or RO, indicates that some or all of the bits can be cleared after a write '1' in corresponding bit.

**Table 2.103 Registers for PCMCIA**

| Name | Address | Width | Access | Description |
|---|---|---|---|---|
| IDR | 0x00 | 32 | RO | Identification Register |
| ISR | 0x04 | 32 | RO | Interface Status Register |
| ICR | 0x08 | 32 | RW | Interface Control Register |
| GCR | 0x0c | 32 | RW | General Control Register |
| CSCR | 0x10 | 32 | ROC | Card Status Change Register |
| IER | 0x14 | 32 | RW | Interrupt Enable Register |
| Setup0 | 0x18 | 32 | RW | Setup Timing 0 Register for IO |
| Command0 | 0x1c | 32 | RW | Command Timing 0 Register for IO |
| Recovery0 | 0x20 | 32 | RW | Recovery Timing 0 Register for IO |
| Setup1 | 0x24 | 32 | RW | Setup Timing 1 Register for common memory |
| Command1 | 0x28 | 32 | RW | Command Timing 1 Register for common memory |

| Recovery1 | 0x2c | 32 | RW | Recovery Timing 1 Register for common memory |
|---|---|---|---|---|
| Setup2 | 0x30 | 32 | RW | Setup Timing 2 Register for attribute memory |
| Command2 | 0x34 | 32 | RW | Command Timing 2 Register for attribute memory |
| Recovery2 | 0x38 | 32 | RW | Recovery Timing 2 Register for attribute memory |
| Start0 | 0x40 | 32 | RW | I/O Start Address 0 |
| End0 | 0x44 | 32 | RW | I/O End Address 0 |
| Start1 | 0x48 | 32 | RW | I/O Start Address 1 |
| End1 | 0x4c | 32 | RW | I/O End Address 1 |
| GPO_enable0 | 0x50 | 32 | RW | Direction control for GPIO-0 |
| GPO_enable1 | 0x54 | 32 | RW | Direction control for GPIO-1 |
| GPO_0 | 0x58 | 32 | RW | Output value for GPIO-0 |
| GPO_1 | 0x5c | 32 | RW | Output value for GPIO-1 |
| GPI_0 | 0x60 | 32 | RW | Input value of GPI-0 |
| GPI_1 | 0x64 | 32 | RW | Input value of GPI-1 |

**Table 2.104 Registers for CardBus**

| Name | Address | Width | Access | Description |
|---|---|---|---|---|
| Command | 0x00 | 32 | RW | CardBus Bridge Command Register |
| Status | 0x04 | 32 | RO | CardBus Bridge Status Register |
| Retry time | 0x08 | 32 | RW | Retry Time Control Register |
| Clk select | 0x0c | 32 | RW | Clock Speed Selection Register |

## 2.13.1.1. Identification Register (IDR)

**Table 2.105 IDR Bit Definition**

Address：**1940_0000**

| Bits | Access | Default | Description |
|---|---|---|---|

**Version 1.5**

| 31:8 | | 0x00 | Reserved. |
| 7: 0 | RO | 0x00 | ID number |
| | | | This 8bit field "0x82". |

## 2.13.1.2.  Interface Status Register (ISR)

**Table 2.106 ISR Bit Definition**

Address：1940_0004

| Bits | Access | Default | Description |
|---|---|---|---|
| 31:8 | | 0x00 | Reserved. |
| 7 | RO | 0 | VS2 |
| | | | This bit reflects the state of the VS2 pin of the PCMCIA. |
| 6 | RO | 0 | VS1 |
| | | | This bit reflects the state of the VS1 pin of the PCMCIA. |
| 5 | RO | 0 | Ready/Busy Signal Status |
| | | | This bit reflects the state of the READY pin of the PCMCIA. |
| 4 | RO | 0 | Memory Write Protect Signal Status |
| | | | This bit reflects the state of the WP pin of the PCMCIA. |
| 3 | RO | 0 | Card Detect 2 Signal Status |
| | | | This bit reflects the state of the CD2 pin of the PCMCIA. |
| 2 | RO | 0 | Card Detect 1 Signal Status |
| | | | This bit reflects the state of the CD1 pin of the PCMCIA. |
| 1 | | 0 | Reserved |
| 0 | RO | 0 | Status Change Signal Status |
| | | | This bit reflects the state of the STSCHG pin of the PCMCIA. |

## 2.13.1.3.  Interface Control Register (ICR)

**Table 2.107 ICR Bit Definition**

Address：**1940_0008**

| Bits | Access | Default | Description |
|---|---|---|---|
| 31:4 | | 0x00 | Reserved. |
| 7 | RW | 0 | VS2 Signal Output<br>0 = drive logic 0 value to VS2 pin<br>1 = input mode or pull-up to VS2 pin |
| 6 | RW | 0 | VS1 Signal Output<br>0 = drive logic 0 value to VS1 pin<br>1 = input mode or pull-up to VS1 pin |
| 5:0 | | 0x0 | Reserved |

## 2.13.1.4. General Control Register (GCR)

**Table 2.108 GCR Bit Definition**

Address：**1940_000C**

| Bits | Access | Default | Description |
|---|---|---|---|
| 31:6 | | 0x00 | Reserved |
| 6:5 | RW | 0x0 | PCMCIA is I/O<br>0 = Socket configured for the memory-only interface<br>1 = Socket configured for I/O and memory interface<br>2 = Socket configured for DMA interface<br>3 = Reserved |
| 4:2 | | 0x0 | Reserved |
| 1 | RW | 0x0 | SPI and UART1 Mode<br>When set to logic 1, this bit converts the CD2~1 pins and VS2~1 pins to the SPI functional pins and CTRDY, CFRAME,CAD19 and CSERR to CTS,RTS,TXD and RXD respectively for UART1 flow control . |
| 0 | RW | 0x0 | CardBus Mode<br>0 = 16-bit PC Card Mode<br>1 = CardBus Mode |

## 2.13.1.5.  Card Status Change Register (CSCR)

**Table 2.109 CSCR Bit Definition**

Address：**1940_0010**

| Bits | Access | Default | Description |
|------|--------|---------|-------------|
| 31:4 |        | 0x00    | Reserved |
| 3    | ROC    | 0x      | Card Detect Change Detected<br>This bit will be set when a transition (low to high or high to low) has occurred on the CD2~1 pins. This bit is reset to 0 when this register is read. |
| 2    | ROC    | 0x0     | Ready Change Detected<br>This bit will be set when a transition (low to high or high to low) has occurred on the READY pin. This bit is reset to 0 when this register is read.<br>This bit has meaning only when the socket is configured for the memory-only interface. In the I/O and memory interface it reads back as 0. |
| 1    |        | 0x0     | Reserved |
| 0    | ROC    | 0x0     | Status Change Detected<br>This bit will be set when a transition (low to high or high to low) has occurred on the STSCHG pin. This bit is reset to 0 when this register is read. |

## 2.13.1.6.  Interrupt Enable Register (IER)

**Table 2.110 IER Bit Definition**

Address：**1940_0014**

| Bits | Access | Default | Description |
|------|--------|---------|-------------|

**Version 1.5**

| 31:4 | | 0x00 | Reserved |
|------|------|------|----------|
| 3 | RW | 0x | Interrupt Enable for Card Detect Change Detected |
| 2 | RW | 0x0 | Interrupt Enable for Ready Change Detected |
| 1 | | 0x0 | Reserved |
| 0 | RW | 0x0 | Interrupt Enable for Status Change Detected |

### 2.13.1.7. Setup Timing Registers (Setup0, Setup1, Setup2)

**Table 2.111 Setup Bit Definition**

Address：**1940_0018**

| Bits | Access | Default | Description |
|------|--------|---------|-------------|
| 31:8 | | 0x00 | Reserved |
| 7:6 | RW | 0x0 | Setup Timing Prescaler Select<br>For this timing set, this field selects prescaler value to prescale the setup value to determine the number of clocks of setup time from valid address before the PCMCIA command goes active.<br>00 = 1<br>01 = 16<br>10 = 256<br>11 = 4096 |
| 5:0 | RW | 0x00 | Setup Timing Multiplier Value<br>For this timing set, this field selects setup value from 0 to 63 to be multiplied with prescaler value to determine the number of clocks of setup time from valid address before the PC card command goes active. |

### 2.13.1.8. Command Timing Registers (Command0, Command1, Command2)

**Table 2.112 Command Bit Definition**

Address：**1940_001C**

Address：**1940_0028**

Address：**1940_0034**

| Bits | Access | Default | Description |
|------|--------|---------|-------------|
| 31:8 | | 0x00 | Reserved |
| 7:6 | RW | 0x0 | Command Timing Prescaler Select<br><br>For this timing set, this field selects prescaler value to prescale the command value to determine the number of clocks of command active time.<br>00 = 1<br>01 = 16<br>10 = 256<br>11 = 4096 |
| 5:0 | RW | 0x00 | Command Timing Multiplier Value<br><br>For this timing set, this field selects command value from 0 to 63 to be multiplied with prescaler value to determine the number of clocks of command active time. |

## 2.13.1.9. Recovery Timing Registers (Recovery0, Recovery 1, Recovery 2)

**Table 2.113 Recovery Bit Definition**

Address：**1940_0020**

Address：**1940_002C**

Address：**1940_0038**

| Bits | Access | Default | Description |
|------|--------|---------|-------------|
| 31:8 | | 0x00 | Reserved |
| 7:6 | RW | 0x0 | Recovery Timing Prescaler Select<br><br>For this timing set, this field selects prescaler value to prescale the command value to determine the number of clocks of recovery time.<br>00 = 1<br>01 = 16 |

| | | | |
|---|---|---|---|
| | | | 10 = 256 |
| | | | 11 = 4096 |
| 5:0 | RW | 0x00 | Recovery Timing Multiplier Value |
| | | | For this timing set, this field selects command value from 0 to 63 to be multiplied with prescaler value to determine the number of clocks of recovery time. |

### 2.13.1.10. Start Address for I/O (Start0, Start1)

**Table 2.114 Start Address Bit Definition**

Address：**1940_0040**

Address：**1940_0048**

| Bits | Access | Default | Description |
|---|---|---|---|
| 31:30 | | 0x00 | Bank width<br>00 = 8-bit<br>01 = 16-bit<br>10 = Auto detect using IOIS16n<br>11 = Reserved |
| 29:26 | | 0x0 | Reserved |
| 25:0 | RW | 0x00 | Start Address<br>This 26-bit field is used for decoding start address for IO bank. |

### 2.13.1.11. End Address for I/O (End0, End1)

**Table 2.115 End Address Bit Definition**

Address：**1940_0044**

Address：**1940_004C**

| Bits | Access | Default | Description |
|---|---|---|---|
| 31:26 | | 0x0 | Reserved |

| Bits | Access | Default | Description |
|------|--------|---------|-------------|
| 25:0 | RW | 0x00 | End Address<br>This 26-bit field is used for decoding end address for IO bank. |

## 2.13.1.12. GPIO Direction Control (GPO_enable0, GPO_enable1)

**Table 2.116 GPIO direction registers Bit Definition**

Address：**1940_0050**

Address：**1940_0054**

| Bits | Access | Default | Description |
|------|--------|---------|-------------|
| 31:0 |  | 0x0 | GPO direction<br>0 = input: All PCMCIA pins are used for PCMCIA feature<br>1 = output: All PCMCIA pins are used for GPIO feature |

## 2.13.1.13. Output value for GPIO (GPO_0, GPO_1)

**Table 2.117 GPO registers Bit Definition**

Address：**1940_0058**

Address：**1940_005C**

| Bits | Access | Default | Description |
|------|--------|---------|-------------|
| 31:0 |  | 0x0 | GPO values |

## 2.13.1.14. Input value of GPIO (GPI_0, GPI_1)

**Table 2.118 GPI registers Bit Definition**

Address：**1940_0060**

Address：**1940_0064**

| Bits | Access | Default | Description |
|------|--------|---------|-------------|

| 31:0 | | 0x0 | GPI values |
|---|---|---|---|

**Table 2.119 GPIO muxing table**

| Pin Name | GPIO | Pin Name | GPIO | Pin Name | GPIO |
|---|---|---|---|---|---|
| CAD0 | GPIO-00 | CAD21 | GPIO-20 | CAD16 | GPIO-40 |
| CAD1 | GPIO-01 | CAD22 | GPIO-21 | RADDR18 | GPIO-41 |
| CAD3 | GPIO-02 | CAD23 | GPIO-22 | nCBLOCK | GPIO-42 |
| CAD5 | GPIO-03 | CAD24 | GPIO-23 | nCSTOP | GPIO-43 |
| CAD7 | GPIO-04 | CAD25 | GPIO-24 | nDEVSEL | GPIO-44 |
| nCBE0 | GPIO-05 | CAD26 | GPIO-25 | nTRDY | GPIO-45 |
| CAD9 | GPIO-06 | CAD27 | GPIO-26 | nFRAME | GPIO-46 |
| CAD11 | GPIO-07 | CAD29 | GPIO-27 | CAD17 | GPIO-47 |
| CAD12 | GPIO-08 | RADDR2 | GPIO-28 | CAD19 | GPIO-48 |
| CAD14 | GPIO-09 | nCCLKRUN | GPIO-29 | VS2 | GPIO-49 |
| nCBE1 | GPIO-10 | CD1 | GPIO-30 | nCSERR | GPIO-50 |
| CPAR | GPIO-11 | CAD2 | GPIO-31 | nCREQ | GPIO-51 |
| nCPERR | GPIO-12 | CAD4 | GPIO-32 | nCBE3 | GPIO-52 |
| nCGNT | GPIO-13 | CAD6 | GPIO-33 | nSTSCHG | GPIO-53 |
| CINT | GPIO-14 | RADDR14 | GPIO-34 | CAD28 | GPIO-54 |
| CCLK | GPIO-15 | CAD8 | GPIO-35 | CAD30 | GPIO-55 |
| nIRDY | GPIO-16 | CAD10 | GPIO-36 | CAD31 | GPIO-56 |
| nCBE2 | GPIO-17 | VS1 | GPIO-37 | CD2 | GPIO-57 |
| CAD18 | GPIO-18 | CAD13 | GPIO-38 | | |
| CAD20 | GPIO-19 | CAD15 | GPIO-39 | | |

## 2.13.1.15. Command for CardBus

**Table 2.120 Command Bit Definition**

Address：**1948_0000**

| Bits | Access | Default | Description |
|------|--------|---------|-------------|
| 31:9 |  | 0x0 | Reserved |
| 9 | RO | 0x0 | Fast back to back enable<br>This 1 bit field is fixed to "0" because the bridge doesn't support fast back to back transaction. |
| 8 | RW | 0x0 | CSERR enable<br>This 1 bit field is used for enabling nCSERR pin interrupt. |
| 7 | RW | 0x0 | Wait cycle control (No connection to actual block)<br>This 1 bit field is fixed to "0" because the bridge doesn't support this function. |
| 6 | RW | 0x1 | Parity error response<br>This 1 bit field is used for enabling nCPERR pin interrupt. |
| 5 | RO | 0x0 | VGA pallet snoop enable<br>This 1 bit field is fixed to logic 0 because the bridge doesn't support this functionality. |
| 4 | RO | 0x0 | Memory write invalidate enable<br>This 1 bit field is used for enabling MWI and MRL command. |
| 3 | RO | 0x0 | Special cycle monitoring enable<br>This 1 bit field is fixed to logic 0 because the bridge doesn't support this functionality. |
| 2 | RO | 0x1 | Bus mastering enable<br>This 1 bit field is fixed to logic 1 because the bridge always supports the master functionality. |
| 1 | RO | 0x1 | Memory access enable<br>This 1 bit field is fixed to logic 1 because the bridge always supports the memory area accessing. |

| 0 | RO | 0x1 | IO access enable |
| | | | This 1 bit field is fixed to logic 1 because the bridge always supports the I/O area accessing. |

### 2.13.1.16. Status for CardBus

This register will be cleared when it is read.

**Table 2.121 Status Bit Definition**

Address：**1948_0004**

| Bits | Access | Default | Description |
|---|---|---|---|
| 31:16 | | 0x0 | Reserved |
| 15 | RO | 0x0 | Detected parity error |
| | | | This 1 bit field is set when nCPERR pin is asserted. |
| 14 | RO | 0x0 | Received system error |
| | | | This 1 bit field is set when nCSERR pin is asserted. |
| 13 | RO | 0x0 | Received master abort |
| | | | This 1 bit field is set when the bridge get the master abort signaling from the PC card. |
| 12 | RO | 0x0 | Received target abort |
| | | | This 1 bit field is set when the bridge get the target abort signaling from the PC card. |
| 11 | RO | 0x0 | Signaled target abort |
| | | | This 1 bit field is set when the bridge make a target abort signaling. |
| 10 | RO | 0x1 | DEVSEL timing |
| | | | This 1 bit field is fixed to 0x1 and that means the bridge operates as a medium slave. |
| 9 | RO | 0x0 | Data parity detected |
| | | | This bit will be set when Bridge detected parity error condition in data phase |
| 8 | RO | 0x0 | Fast back to back capable |
| | | | The CardBus bridge doesn't support fast back to back transactions. |

| Bits | Access | Default | Description |
|------|--------|---------|-------------|
| 7:0 | | 0x00 | Reserved |

### 2.13.1.17. Retry time (Retry)

**Table 2.122 Retry time register Bit Definition**

Address：**1948_0008**

| Bits | Access | Default | Description |
|------|--------|---------|-------------|
| 31:9 | | 0x0 | Reserved |
| 8:4 | RW | 0x0 | Access time<br>This 5-bit field is used for checking the latency of data.<br>If slave failed to make valid data by the access time, then it enter to the target abort state.<br>If this value is zero, then there is no checking for the latency of data. |
| 3:0 | RW | 0x0 | Retry time<br>This 4-bit field is used for limiting the number of retry. If this value is zero, then there is no limit for the retry. |
| 7:0 | | 0x00 | Reserved |

### 2.13.1.18. Clk select (Csel)

**Table 2.123 Clk select register Bit Definition**

Address：**1948_000C**

| Bits | Access | Default | Description |
|------|--------|---------|-------------|
| 31:1 | | 0x0 | Reserved |
| 0 | RW | 0x0 | Divid by 4<br>This 1-bit field controls the CardBus clock speed.<br>0 = System clock/2<br>1 = System clock/4 |

# 3. Electrical Characteristics

## 3.1. Absolute Maximum Ratings

**Table 3.1 Absolute maximum ratings**

| Parameter | Symbol | Rating | Units |
|---|---|---|---|
| Supply Voltage (Core) | $V_{DDI}$ | TBD | V |
| Supply Voltage (I/O) | $V_{DDP}$ | TBD | V |
| DC Input Current | $I_{IN}$ | TBD | MA |
| Operating Temperature | $T_{OPERATING}$ | 0 to 70 | °C |
| Storage Temperature | $T_{STORAGE}$ | | °C |

## 3.2. Recommended Operating Conditions

**Table 3.2 Recommended operating conditions**

| Parameter | Symbol | Rating | Units |
|---|---|---|---|
| Supply Voltage (Core) | $V_{DDI}$ | 2.5 | V |
| Supply Voltage (I/O) | $V_{DDP}$ | 3.3 | V |
| Oscillator Frequency | $f_{OSC}$ | TBD | MHz |

**Version 1.5**

# 3.3. DC Characteristics

TBD

**Table 3.3 DC Characteristics**

| Parameter | Symbol | Condition | Min. | Typ. | Max. | Unit |
|---|---|---|---|---|---|---|
| High Level Input Voltage | $V_{IH}$ | | | | | V |
| Low Level Input Voltage | $V_{IL}$ | | | | | V |
| High Level Input Current | $I_{IH}$ | | | | | μA |
| Low Level Input Current | $I_{IL}$ | | | | | μA |
| High Level Output Voltage | $V_{OH}$ | | | | | V |
| Low Level Output Voltage | $V_{OL}$ | | | | | V |
| Tri-state output leakage current | | | | | | |
| Maximum operating current | | | | | | |

**Version 1.5**

# 3.4. AC Characteristics

## 3.4.1. Clocks
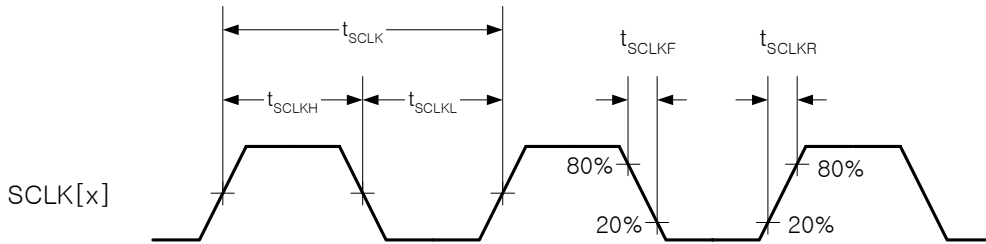


**Figure 3.1 SDRAM Clock Timing**
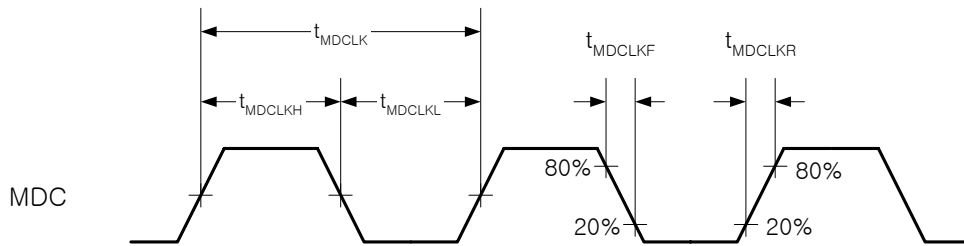


**Figure 3.2 MDC Timing (Ethernet)**



**Figure 3.3 SPI Clock Timing**

**Version 1.5**

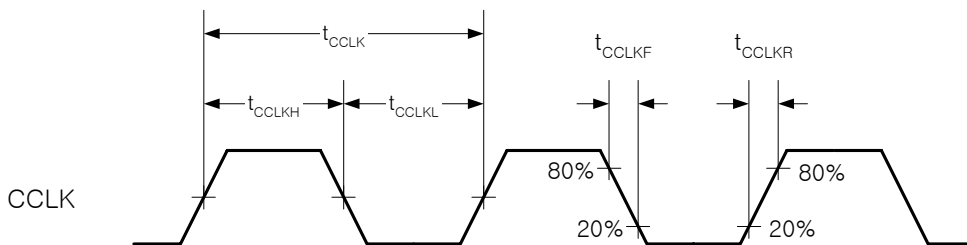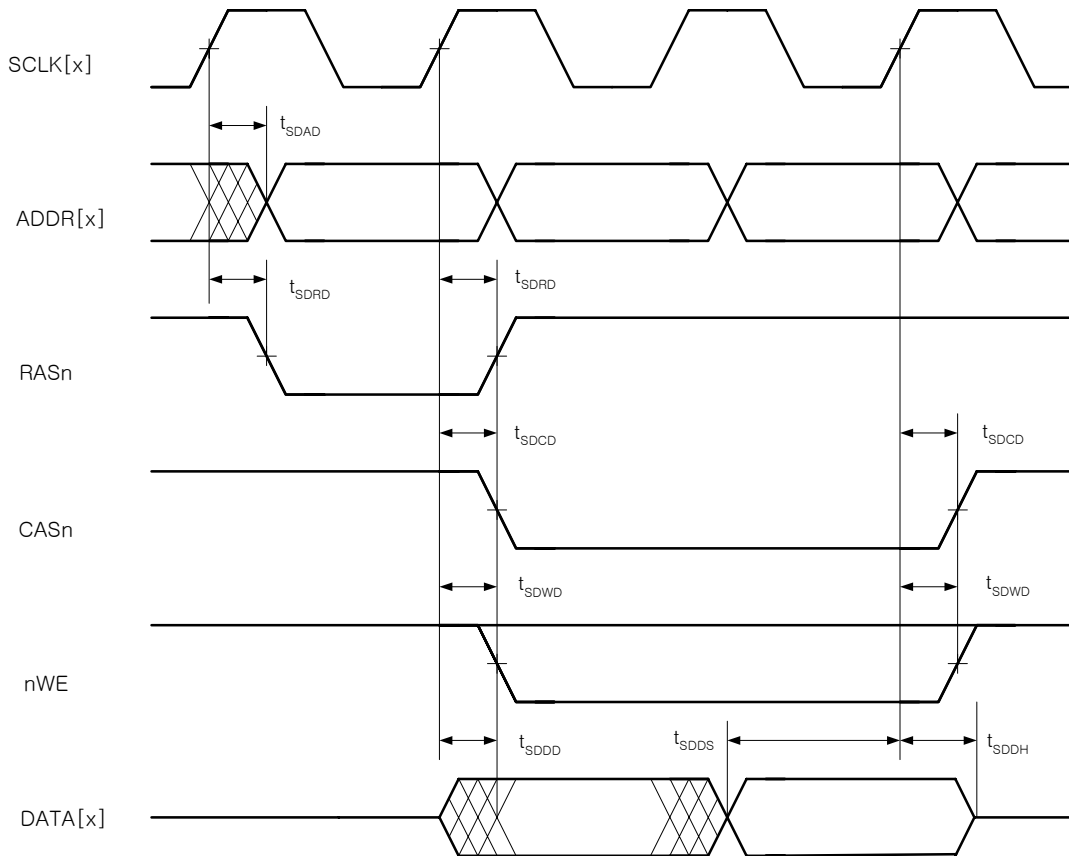## 3.4.2.  SDRAM Timing



**Figure 3.4 SDRAM Timing Diagram**

### 3.4.3. Ethernet Timing (MII/100Mbps)
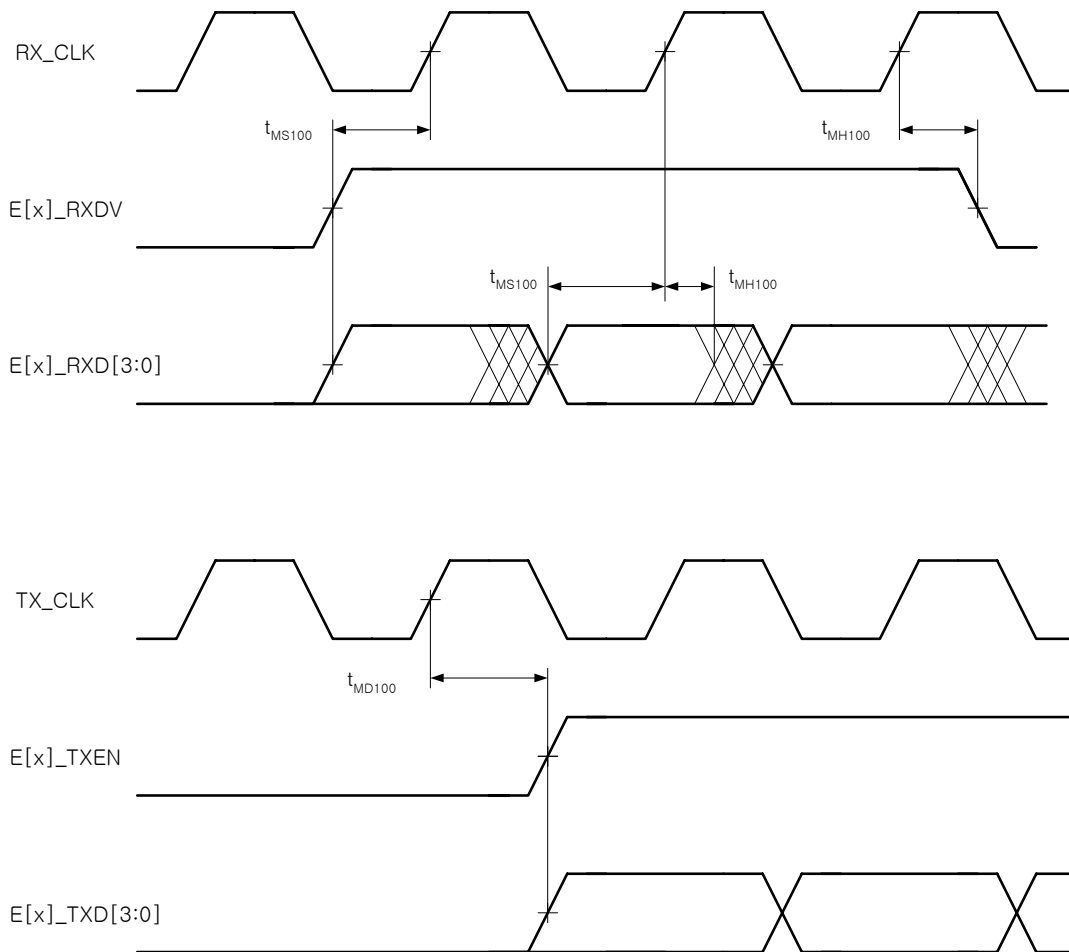


**Figure 3.5 Ethernet MII Timing Diagram (100Mbps)**
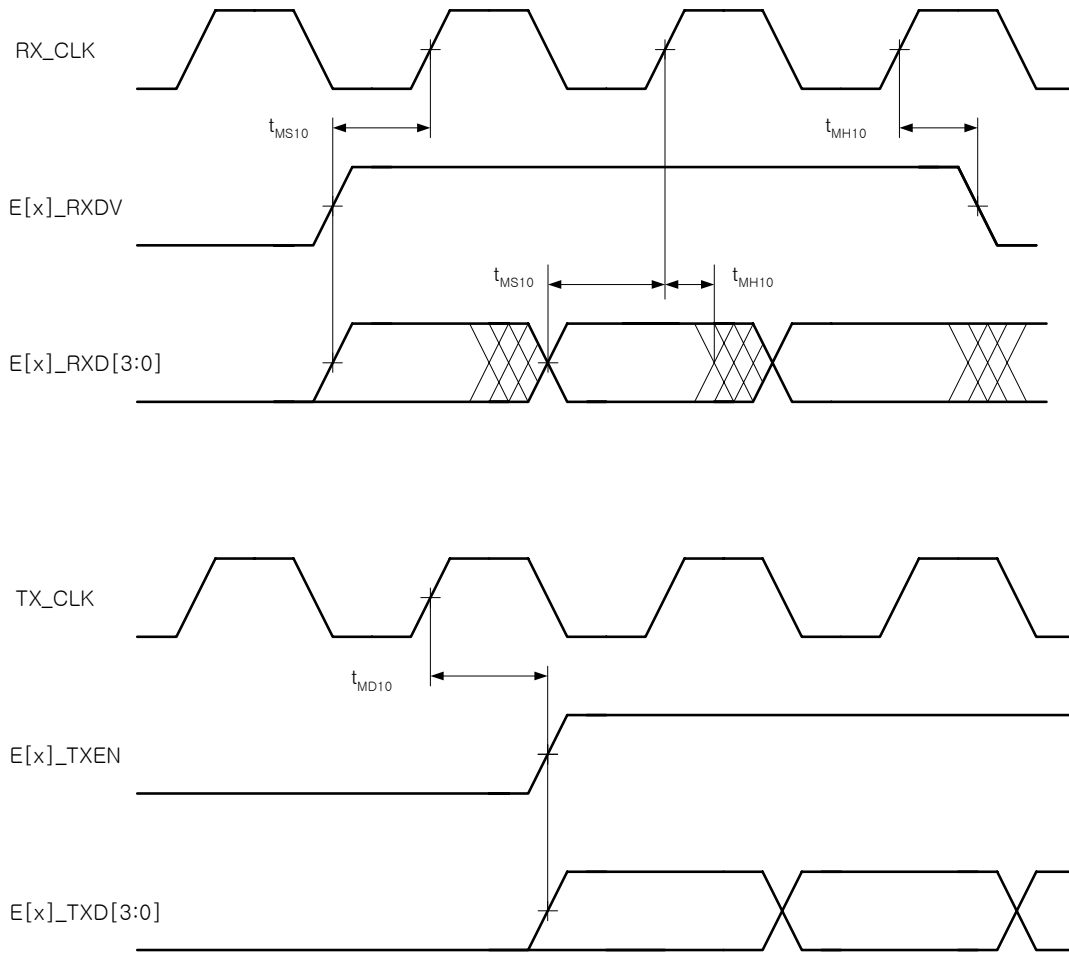
## 3.4.4.    Ethernet Timing (MII/10Mbps)



**Figure 3.6 Ethernet MII Timing Diagram (10Mbps)**
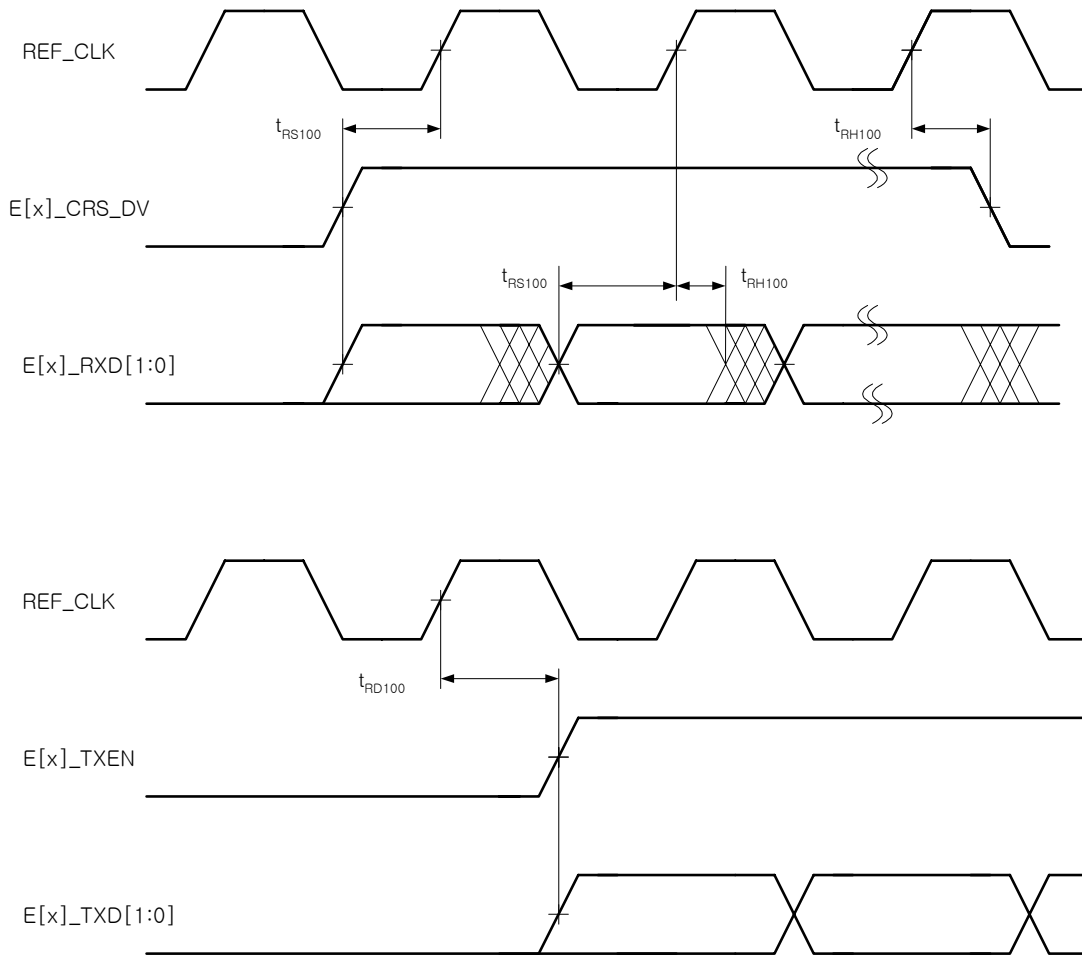
## 3.4.5.    Ethernet Timing (RMII)



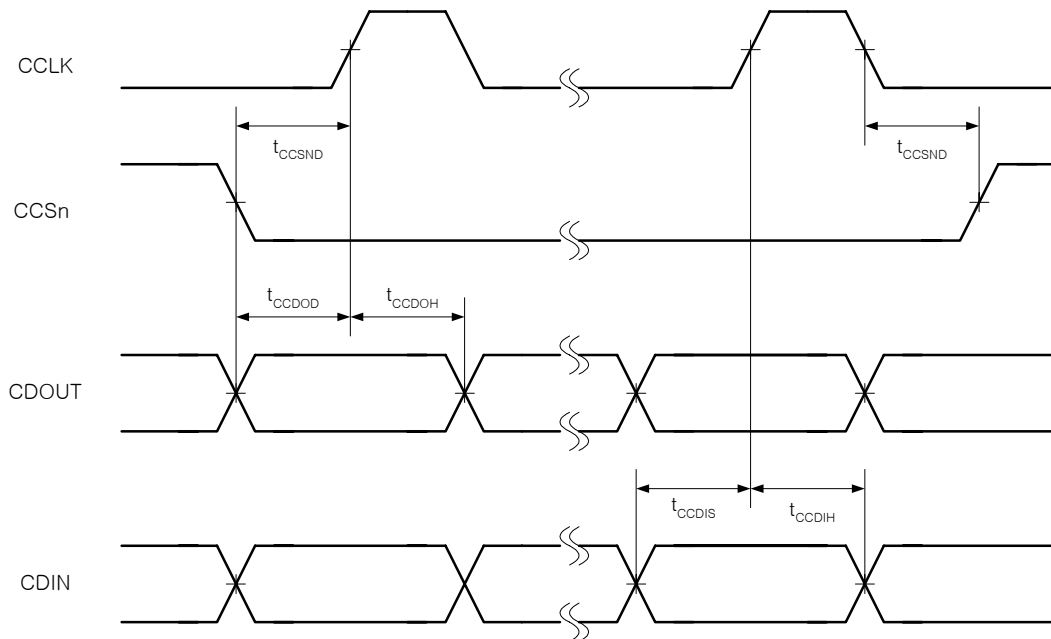**Figure 3.7 Ethernet RMII Timing Diagram**

## 3.4.6.   SPI Timing



**Figure 3.8 SPI Timing Diagram**

**Version 1.5**

**Table 3.4 A.C. Electrical Characteristics**

(TA = 0°C to 70°C, VDD = 3.0V to 3.3V, Unit = ns)

| Parameter | Description | Min | Typ | Max |
|---|---|---|---|---|
| $t_{SCLK}$ | Clock period for SCLK | – | – | – |
| $t_{SCLKH}$ | Clock high time for SCLK | – | – | – |
| $t_{SCLKL}$ | Clock low time for SCLK | – | – | – |
| $t_{SCLKF}$ | Clock falling time for SCLK | – | – | – |
| $t_{SCLKR}$ | Clock rising time for SCLK | – | – | – |
| $t_{MDCLK}$ | Clock period for MDC | – | – | – |
| $t_{MDCLKH}$ | Clock high time for MDC | – | – | – |
| $t_{MDCLKL}$ | Clock low time for MDC | – | – | – |
| $t_{MDCLKF}$ | Clock falling time for MDC | – | – | – |
| $t_{MDCLKR}$ | Clock rising time for MDC | – | – | – |
| $t_{CCLK}$ | Clock period for CCLK | – | – | – |
| $t_{CCLKH}$ | Clock high time for CCLK | – | – | – |
| $t_{CCLKL}$ | Clock low time for CCLK | – | – | – |
| $t_{CCLKF}$ | Clock falling time for CCLK | – | – | – |
| $t_{CCLKR}$ | Clock rising time for CCLK | – | – | – |
| $t_{SDAD}$ | SDRAM address output delay | – | – | – |
| $t_{SDRD}$ | SDRAM RAS output delay | – | – | – |
| $t_{SDCD}$ | SDRAM CAS output delay | – | – | – |
| $t_{SDWD}$ | SDRAM WE output delay | – | – | – |
| $t_{SDDD}$ | SDRAM DATA output delay | – | – | – |
| $t_{SDDS}$ | SDRAM DATA setup time | – | – | – |
| $t_{SDDH}$ | SDRAM DATA hold time | – | – | – |
| $t_{MS100}$ | MII setup time for 100Mbps | – | – | – |
| $t_{MH100}$ | MII hold time for 100Mbps | – | – | – |
| $t_{MD100}$ | MII output delay for 100Mbps | – | – | – |
| $t_{MS10}$ | MII setup time for 10Mbps | – | – | – |
| $t_{MH10}$ | MII hold time for 10Mbps | – | – | – |
| $t_{MD10}$ | MII output delay for 10Mbps | – | – | – |
| $t_{RS100}$ | RMII setup time | – | – | – |

**Version 1.5**

| $t_{RH100}$ | RMII hold time | – | – | – |
|---|---|---|---|---|
| $t_{RD100}$ | RMII output delay | – | – | – |
| $t_{CCSND}$ | SPI delay from CCSn active to rising edge of CCLK | – | – | – |
| $t_{CCDOD}$ | SPI delay from CDOUT valid to rising edge of CCLK | – | – | – |
| $t_{CCDIS}$ | SPI CDIN setup time | – | – | – |
| $t_{CCDIH}$ | SPI CDIN hold time | – | – | – |

**Version 1.5**

# 4. Mechanical Characteristics
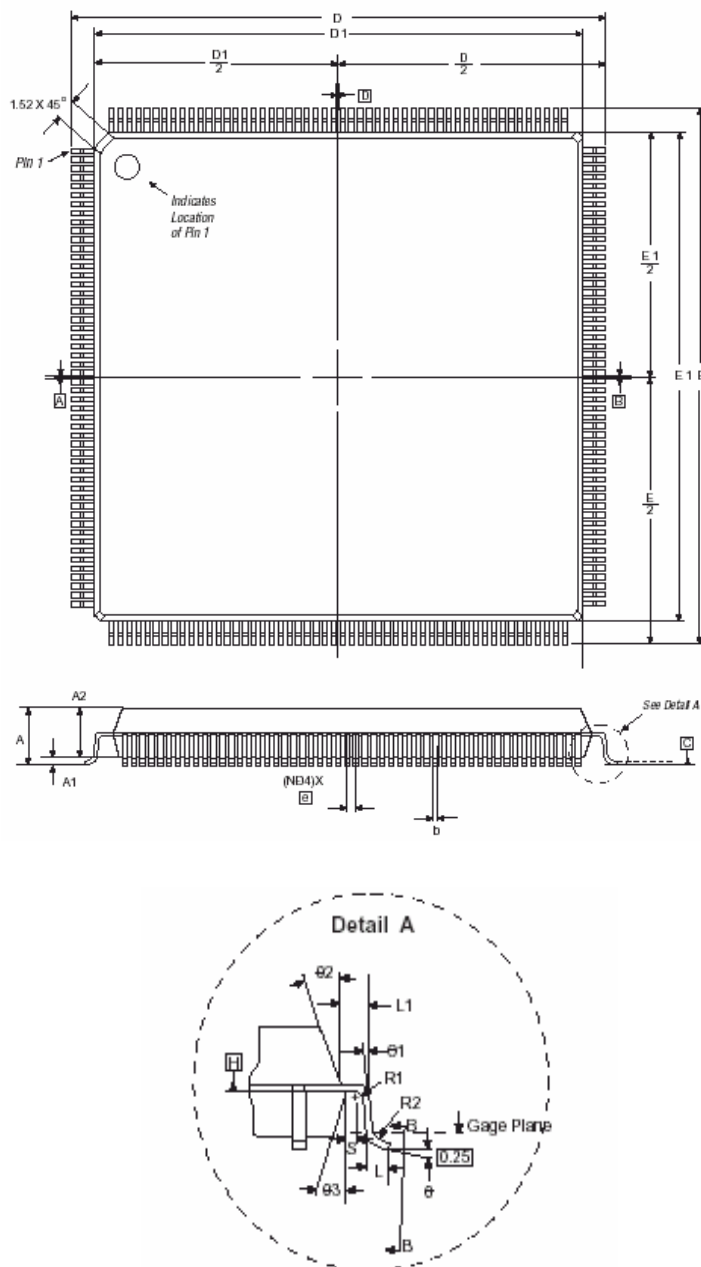


| Package Outline Figure Reference | | | |
|---|---|---|---|
| Symbol | Millimeters | | |
| | Min. | Nom. | Max. |
| A | – | – | 4.10 |
| A1 | 0.25 | – | 0.50 |
| A2 | 3.20 | 3.40 | 3.60 |
| D | 30.35 BSC | | |
| D1 | 27.90 | – | 28.10 |
| E | 30.35 BSC | | |
| E1 | 27.90 | – | 28.10 |
| e | 0.50 BSC | | |
| b | 0.17 | – | 0.27 |
| R2 | 0.08 | – | 0.25 |
| R1 | 0.08 | – | – |
| $\theta$ | 0° | 3.5° | 8° |
| $\theta 1$ | 0° | – | – |
| $\theta 2$ | 5° | – | 16° |
| $\theta 3$ | 5° | – | 16° |
| L | 0.46 | – | 0.66 |
| L1 | 0.40 | – | – |
| S | 0.20 | – | – |
| N | 208 | | |

**Figure 4.1 Mechanical Characteristics**

**Version 1.5**

# 5. Ordering Information

TBD