# Cortina Systems® IXF1104 4-Port Gigabit Ethernet Media Access Controller

## Datasheet

The Cortina Systems® Cortina Systems® IXF1104 4-Port Gigabit Ethernet Media Access Controller (IXF1104 MAC) supports IEEE 802.3* 10/100/1000 Mbps applications. The IXF1104 MAC supports a System Packet Interface Phase 3 (SPI3) system interface to a network processor or ASIC, and concurrently supports copper and fiber physical layer devices (PHYs).

The copper PHY interface supports the standard and reduced pin-count Gigabit Media Independent Interface (GMII and RGMII) for high-port-count applications. For fiber applications the integrated Serializer/Deserializer (SerDes) on each port supports direct connection to optical modules to reduce PCB area requirements and system cost.

## Product Features

- Four Independent Ethernet MAC Ports for Copper or Fiber Physical layer connectivity.
    - IEEE 802.3 compliant
    - Independent Enable/Disable of any port
- Copper Mode:
    - RGMII for 10/100/1000 Mbps links
    - GMII for 1000 Mbps full-duplex links
    - IEEE 802.3 MDIO interface
- Fiber Mode:
    - Integrated SerDes interface for direct connection to 1000BASE-X optical modules
    - IEEE 802.3 auto-negotiation or forced mode
    - Supports SFP MSA-compatible transceivers
- SPI3 interface supports data transfers up to 4 Gbps in both modes:
    - 32-bit Multi-PHY mode (133 MHz)
    - 4 x 8-bit Single-PHY mode (125 MHz)
- IEEE 802.3-compliant Flow Control
    - Loss-less up to 9.6 KB packets and 5 km links
    - Jumbo frame support for 9.6 KB packets
- Internal per-channel FIFOs: 32 KB Rx, 10 KB Tx
- Flexible 32/16/8-bit CPU interface

- Programmable Packet handling
    - Filter broadcast, multicast, unicast, VLAN and errored packets
    - Automatically pad undersized Tx packets
    - Remove CRC from Rx packets
- Performance Monitoring and Diagnostics
    - RMON Statistics
    - CRC calculation and error detection
    - Detection of length error, runt, or overly large packets
    - Counters for dropped and errored packets
    - Loopback modes
    - JTAG boundary scan
- .18 m CMOS process technology
    - 1.8 V core, 2.5 V RGMII, GMII, OMI, and 3.3 V SPI3 and CPU
- Operating Temperature Ranges:
    - Copper Mode:-40°C to +85°C
    - Fiber Mode:0°C to +70°C
- Package Options:
    - 552-ball Ceramic BGA (standard)
    - 552-ball Ceramic BGA (RoHS-compliant)
    - 552-ball Plastic FC-BGA (contact your Cortina Sales Representative)

## Applications

- Load Balancing Systems
- MultiService Switches
- Web Caching Appliances
- Intelligent Backplane Interfaces
- Edge Routers
- Redundant Line Cards

- Base Station Controllers and Transceivers
- Serving GPRS Support Nodes (SGSN)
- Gateway GPRS Support Nodes (GGSN)
- Packet Data Serving Nodes (PDSN)
- DSL Access Multiplexers (DSLAM)
- Cable Modem Termination Systems (CMTS)

# Revision History

| Revision Number: 10.0 Revision Date: 28 November 2006 | |
| --- | --- |
| Page # | Description |
| N/A | First release of this document from Cortina Systems, Inc. |

| Revision Number: 009 Revision Date: 27-Oct-2005 | |
| --- | --- |
| Page # | Description |
| page 71 | Modified Figure 8, *Ethernet Frame Format* [changed Preamble byte count to 7 bytes]. |
| page 133 | Section 45, *RGMII Power* [changed $V_{CC}$ to $V_{DD}$ in $I_{IH}$ and $I_{IL}$] |
| page 111 | Added bullet to Section 5.7.3, *I²C Module Configuration Interface*: *The I2C interface only supports random single-byte reads and does not guarantee coherency when reading two-byte registers.* |
| page 219 | Replaced Figure 56, *FC-PBGA Package (Top and Bottom Views)*, on page 219. |
| page 225 | Modified Table 182, *SPI3 Receive Configuration ($0x701)*. |
| page 238 | Modified Table 206, *Optical Module Control Ports 0 - 3 ($0x79A)*: changed default values. |
| page 239 | Modified Table 207, *$I^2C$ Control Ports 0 - 3 ($0x79B)*. |
| page 227 | Modified Table 213, *I2C Data Ports 0 - 9 ($0x79F)* (changed address from $0x79C to $0x79F). |
| page 220 | Added Section 9.3.3, *Top Label Marking Example*. |
| page 235 | Modified Table 157, *Product Information* and Figure 60, *Ordering Information – Sample* under Section 10.0, *Product Ordering Information*. |
| page 1 | Added 552-ball Flip Chip-PBGA (FC-PBGA) and Product Ordering Number information. |
| page 222 | Added FC-PBGA ordering number under Table 157, *Product Information*. |

| Revision Number: 008 Revision Date: August 1, 2005 (Sheet 1 of 2) | |
| --- | --- |
| Page # | Description |
| page 1 | Added 552-ball Ceramic Ball Grid Array (CBGA) compliant with RoHS and Product Ordering Number information. |
| page 1 | Added 552-ball Flip Chip-PBGA (FC-PBGA) and Product Ordering Number information. |
| page 58 | Modified Table 12, *JTAG Interface Signal Descriptions*: changed Standard to 3.3 V LVTTL from 2.5 V CMOS. |
| page 72 | Modified Figure 9, *PAUSE Frame Format* [changed Preamble byte count to 7 bytes]. |
| page 84 | Modified Figure 11, *MPHY Transmit Logical Timing* [updated TDAT[31:0]]. |
| page 85 | Modified Figure 12, *MPHY Receive Logical Timing* [updated RDAT[31:0]]. |
| page 87 | Modified Figure 14, *SPHY Transmit Logical Timing* [updated TDAT[7:0]]. |
| page 88 | Modified Figure 15, *SPHY Receive Logical Timing* [updated RDAT[7:0] and RPRTY]. |
| page 123 | Modified Figure 35, *Read Timing Diagram - Asynchronous Interface*: changed uPx_ADD[12:0] to uPx_ADD[10:0]. |
| page 126 | Added paragraphs two and three under Section 5.11, *Loopback Modes*. |

| Revision Number: 008 Revision Date: August 1, 2005 (Sheet 2 of 2) | |
| --- | --- |
| **Page #** | **Description** |
| page 130 | Changed 3.3 V CMOS to 2.5 V CMOS under Section 5.12.5, *JTAG Clock, on page 130*. |
| page 128 | Added Section 6.2, *Disable and Enable Port Sequences*. |
| page 133 | Modified Table 45, *RGMII Power* [changed $V_{OH}$, $V_{OL}$, $V_{IH}$, $V_{IL}$ minimum conditions to $V_{DD}$ and changed $V_{IN}$ value to $V_{DD}$ + .3]. |
| page 134 | Modified Table 46, *SPI3 Receive Interface Signal Parameters* [changed RFCLK duty cycle to 45 min and 55 max; Changed Min for RFCLK frequency to 90]. |
| page 136 | Modified Table 47, *SPI3 Transmit Interface Signal Parameters* [changed TFCLK duty cycle to 45 min and 55 max]. |
| page 141 | Changed MDC to MDIO Output delay max for t3 for 2.5 MHz from 200 to 300 in Table 52, *MDIO Timing Parameters, on page 141*. |
| page 167 | Modified Table 90, *TX Config Word ($ Port_Index + 0x17)* [changed default value for the register from "0x0001A0" to "0x000001A0" and changed default value for bit 6 (Half Duplex) from 1 to 0]. |
| page 179 | Modified Table 96, *PHY Control ($ Port Index + 0x60)* [added "Need one-sentence descriptions of register" and register default value]. |
| page 180 | Modified Table 97, *PHY Status ($ Port Index + 0x61)* [added "Need one-sentence descriptions of register" and register default value]. |
| page 181 | Modified Table 98, *PHY Identification 1 ($ Port Index + 0x62)* [added "Need one-sentence descriptions of register" and register default value]. |
| page 182 | Modified Table 99, *PHY Identification 2 ($ Port Index + 0x63)* [added "Need one-sentence descriptions of register" and register default value]. |
| page 182 | Modified Table 100, *Auto-Negotiation Advertisement ($ Port Index + 0x64)* [added "Need one-sentence descriptions of register" and register default value]. |
| page 183 | Modified Table 101, *Auto-Negotiation Link Partner Base Page Ability ($ Port Index + 0x65)* [added "Need one-sentence descriptions of register" and register default value]. |
| page 184 | Modified Table 102, *Auto-Negotiation Expansion ($ Port Index + 0x66)* [added "Need one-sentence descriptions of register" and register default value]. |
| page 185 | Modified Table 103, *Auto-Negotiation Next Page Transmit ($ Port Index + 0x67)* [added "Need one-sentence descriptions of register" and register default value]. |
| page 221 | Modified Table 178, *MDIO Single Read and Write Data ($0x681)* [changed MDIO write data to "MDIO write data to external device"]. |
| page 223 | Modified Table 181, *SPI3 Transmit and Global Configuration ($0x700)* [changed default value for bits 3:0 from "0" to "1" and changed default value for entire register from "0x0020000F" to "0x00200000"]. |
| page 225 | Modified Table 182, *SPI3 Receive Configuration ($0x701)* [changed default value for bits 11:8 from "0xF" to "0x1"]. |
| page 238 | Modified Table 206, *Optical Module Control Ports 0 - 3 ($0x79A)* [changed default value for bits 16:13 from "0xF" to "0x1"]. |
| page 219 | Added Figure 56, *FC-PBGA Package (Top and Bottom Views), on page 219* and Figure 57, *FC-PBGA Mechanical Specifications, on page 220*. |
| page 221 | Replaced Figure 58, *Package Marking Example−Cortina Systems Marking*. |
| 229 | Added Section 9.4, "RoHS Compliance" on page 229. |
| page 222 | Added CBGA RoHS-compliant and FC-PBGA ordering information under Table 157, *Product Information*. |
| page 222 | Added FC-PBGA ordering number under Table 157, *Product Information*. |

<table>
<tr><td colspan="2" align="center"><b>Revision Number: 007</b><br><b>Revision Date: March 24, 2004</b><br><b>(Sheet 1 of 5)</b></td></tr>
<tr><td><b>Page #</b></td><td><b>Description</b></td></tr>
<tr><td>All</td><td>Globally replaced GBIC with Optical Module Interface.</td></tr>
<tr><td>All</td><td>Globally edited signal names.</td></tr>
<tr><td>All</td><td>Globally changed SerDes and PLL analog power ball names as follows:<br>TXAVTT and RXAVTT changed to AVDD1P8_2<br>TXAV25 and RXAV25 changed to AVDD2P5_2<br>PLL1_VDDA and PLL2_VDDA changed to AVDD1P8_1<br>PLL3_VDDA changed to AVDD2P5_1<br>PLL1_GNDA, PLL2_GNDA, and PLL3_GNDA changed to GND</td></tr>
<tr><td>1</td><td>Reworded and rearranged the Product Features section on page one<br>Changed Jumbo frame support from "10 kbytes" to "9.6 KB".</td></tr>
<tr><td>21</td><td>Changed heading to Section 2.0, "General Description" [was Section 2.0, "Block Diagram"].</td></tr>
<tr><td>23/37</td><td>Reversed sections as follows:<br>Section 3.0, "Ball Assignments and Ball List Tables"<br>Section 4.0, "Ball Assignments and Signal Descriptions"</td></tr>
<tr><td>24</td><td>Modified Table 1 "Ball List in Alphanumeric Order by Signal Name":<br>Changed A10 from VCC to VDD<br>Changed C12 from VCC to VDD<br>Changed D11 from VCC to VDD<br>Changed J20 from GND to VDD<br><br>Changed Ball A1 from NC to No Pad.<br>Changed Balls A2, A3, A22, A23, A24, B1, B2, B23, B24, C1, C24, AB1, AB24, AC1, AC2, AC23, AC24, AD1, AD2, AD3, AD22, AD23, AD24 from NC to No Ball.</td></tr>
<tr><td>30</td><td>Modified Table 2 "Ball List in Alphanumeric Order by Ball Location"<br>Changed A10 from VCC to VDD<br>Changed C12 form VCC to VDD<br>Changed D11 from VCC to VDD<br>Changed J20 from GND to VDD<br><br>Changed Ball A1 from NC to No Pad.<br>Changed Balls A2, A3, A22, A23, A24, B1, B2, B23, B24, C1, C24, AB1, AB24, AC1, AC2, AC23, AC24, AD1, AD2, AD3, AD22, AD23, AD24 from NC to No Ball.</td></tr>
<tr><td>38</td><td>Updated Figure 4 "Interface Signals" [modified SPI3 interface signals and added MPHY and SPHY categories; modified signal names].</td></tr>
<tr><td>39</td><td>Broke old Table 1, "IXF1104 Signal Descriptions" into the following:<br>Table 3 "SPI3 Interface Signal Descriptions" on page 39 through Table 14 "Power Supply Signal Descriptions" on page 56</td></tr>
<tr><td>39</td><td>Modified Table 3 "SPI3 Interface Signal Descriptions" on page 39 [edited description for DTPA; added text to TFCLK description; added text to RFCLK description].</td></tr>
<tr><td>50</td><td>Modified Table 6 "RGMII Interface Signal Descriptions" [Added Ball Designators; added notes under descriptions].</td></tr>
<tr><td>51</td><td>Modified Table 7 "CPU Interface Signal Descriptions" [UPX_DATA[16]: deleted J10, added M10].</td></tr>
<tr><td>53</td><td>Modified Table 9 "Optical Module Interface Signal Descriptions" [added Ball Designators].</td></tr>
<tr><td>54</td><td>Modified Table 10 "MDIO Interface Signal Descriptions" [moved note from MDC to MDIO].</td></tr>
<tr><td>56</td><td>Modified Table 14 "Power Supply Signal Descriptions" [added Ball Designators A4, A21, and AD21 to GND; added AVDD1P8_1, AVDD1P8_2, AVDD2P5_1, and AVDD2P5_2].</td></tr>
</table>

| Revision Number: 007 <br> Revision Date: March 24, 2004 <br> (Sheet 2 of 5) | |
|---|---|
| **Page #** | **Description** |
| 39 | Modified Section 4.3, "Signal Description Tables" [changed heading from "Signal Naming Conventions; added new headings Section 4.1.1, "Signal Name Conventions" and Section 4.1.2, "Register Address Conventions"; and added/enhanced material under headings. |
| 58 | Added new Section 4.5, "Multiplexed Ball Connections" with Table 16 "Line Side Interface Multiplexed Balls" and Table 17 "SPI3 MPHY/SPHY Interface". |
| 63 | Modified Section 4.7, "Power Supply Sequencing" [changed language under this section and added Section 4.7.1, "Power-Up Sequence" and Section 4.7.2, "Power-Down Sequence"]. |
| 63 | Modified Table 5 "Power Supply Sequencing" [deleted 3.3 V Supplies Stable; changed Apply 1.8 V to VDD, AVDD1P8_1, and AVDD1P8_2; changed Apply 2.5 V to AVDD2P5_1 and AVDD2P5_2]. |
| 61 | Modified Table 18 "Definition of Output and Bi-directional Balls During Hardware Reset" [changed comments for Optical Modules]. |
| 64 | Modified Table 20 "Pull-Up/Pull-Down and Unused Ball Guidelines" [changed TRST_L to pull-down; added MDIO, UPX_RDY_L, I$^2$C_DATA_3:0, and TX_DISABLE_3:0]. |
| 64 | Added new Section 4.9, "Analog Power Filtering" [including Figure 6 "Analog Power Supply Filter Network" on page 65 and Table 21 "Analog Power Balls" on page 65]. |
| 66 | Modified/edited text under Section 5.1, "Media Access Controller (MAC)" [rearranged and created new bullets]. |
| 67 | Modified first paragraph under Section 5.1.1.1, "Padding of Undersized Frames on Transmit". |
| 67 | Modified entire Section 5.1.1.3, "Filtering of Receive Packets". |
| 68 | Added new Section 5.1.1.3.6, "Filter CRC Error Packets". |
| 69 | Added note under Table 22 "CRC Errored Packets Drop Enable Behavior". |
| 69 | Added new Section 5.1.2, "Flow Control" including Figure 7 "Packet Buffering FIFO", Figure 8 "Ethernet Frame Format", and Figure 9 "PAUSE Frame Format". |
| 73 | Replaced Section 5.1.2.1.5, "Transmit Pause Control Interface" [added Table 23 "Valid Decodes for TXPAUSEADD[2:0]" and modified Table 10 "Transmit Pause Control Interface". |
| 74 | Modified Figure 10 "Transmit Pause Control Interface" |
| 75 | Added note under Section 5.1.3.1, "Configuration". |
| 76 | Added table note to Table 24 "Operational Mode Configuration Registers". |
| 77 | Added note under Section 5.1.4.3, "Fiber Forced Mode". |
| 79 | Modified Section 5.1.6.2, "TX Statistics" [added text to third sentence in first paragraph]. |
| 79 | Modified Section 5.1.6.3, "Loss-less Flow Control" [changed "two kilometers" to "five kilometers" in last sentence. |
| 80 | Modified Section 5.1.7.1.2, "RX FIFO" [changed 10 KB to 9.6 KB; added text to last paragraph]. |
| 83 | Rewrote/replaced Section 5.2, "SPI3 Interface". |
| 86 | Edited signal names in Figure 13 "MPHY 32-Bit Interface". |
| 90 | Edited signal names in Figure 16 "SPHY Connection for Two IXF1104 MAC Ports (8-Bit Interface)". |
| 91 | Added new Section 5.2.2.9, "SPI3 Flow Control". <br> [Removed old "Packet-Level and Byte-Level Transfers" section.} |
| 94 | Modified Figure 17 "MAC GMII Interconnect" [edited signal names]. |
| NA | Removed old Section 5.3.3 Electrical Requirements and Table 27 "Electrical Requirements" – changed Input high current Max from 40 to 15 and Input low current Min from -600 to -15. |
| 96 | Added a note under Section 5.4, "Reduced Gigabit Media Independent Interface (RGMII)". |

| Revision Number: 007 Revision Date: March 24, 2004 (Sheet 3 of 5) | |
|---|---|
| **Page #** | **Description** |
| 96 | Modified Figure 18 "RGMII Interface" [edited signal names]. |
| 98 | Modified Figure 19 "TX_CTL Behavior" [changed signal names]. |
| 98 | Modified Figure 20 "RX_CTL Behavior" [changed signal names]. |
| 99 | Modified Section 5.5, "MDIO Control and Interface" [changed 3.3 us to 3.3 ms in fourth paragraph, third sentence]. |
| 103 | Modified/replaced all text under Section 5.6, "SerDes Interface" on page 103 [added Table 29 "SerDes Driver TX Power Levels"]. |
| NA | Removed old Section 5.6.2.4 AC/DC Coupling. |
| NA | Removed old Section 5.6.2.9 System Jitter. |
| 107 | Modified Table 30 "IXF1104 MAC-to-SFP Optical Module Interface Connections" [edited signal names]. |
| 107 | Modified/replaced text and deleted old "Figure 19. Typical GBIC Module Functional Diagram" under Section 5.7, "Optical Module Interface"]. |
| 108 | Modified second sentence under Section 5.7.2.2.1, "MOD_DEF_0:3". |
| 109 | Modified second sentence under Section 5.7.2.2.3, "RX_LOS_0:3". |
| 109 | Removed third paragraph under Section 5.7.2.2.7, "RX_LOS_INT". |
| 110 | Modified first and second paragraphs under Section 5.7.3, "I²C Module Configuration Interface". |
| 111 | Modified Section 5.7.3.3, "I2C Write Operation" [edited portions of text]. |
| 116 | Modified Table 31 "LED Interface Signal Descriptions" [changed 0.5 MHz to 720 Hz for LED_CLK under Signal Description]. |
| 119 | Modified Table 35 "LED Behavior (Fiber Mode)" [changed links under Description to "Link LED Enable ($0x502)"]. |
| NA | Removed old Figure 30 "CPU – External and Internal Connections". |
| 123 | Modified Table 37 "Byte Swapper Behavior" [edited/added new values]. |
| 123 | Modified second paragraph under Section 5.10, "TAP Interface (JTAG)" |
| 126 | Modified Figure 33 "SPI3 Interface Loopback Path". |
| 126 | Added note under Section 5.11.2, "Line Side Interface Loopback". |
| 127 | Modified Figure 34 "Line Side Interface Loopback Path". |
| 127 | Changed Section 5.12, "Clocks" [from GBIC output clock to I$^2$C Clock]. |
| 129 | Changed Section 5.12.6, "I2C Clock" [from GBIC Clock to I$^2$C Clock]. |
| 130 | Added new Section 6.0, "Applications". |
| 132 | Modified Table 39 "Absolute Maximum Ratings" [changed SerDes analog power to AVDD1P8_2 and AVDD2P5_2; changed "PLL1_VDDA and PLL2_VDDA to AVDD1P8_1; changed PLL3_VDDA to AVDD2P5_1. |
| 133 | Modified Table 40 "Recommended Operating Conditions" [changed SerDes analog power to AVDD1P8_2 and AVDD2P5_2; changed "PLL1_VDDA and PLL2_VDDA to AVDD1P8_1; changed PLL3_VDDA to AVDD2P5_1. |
| 134 | Modified Table 42 "SerDes Transmit Characteristics" [included SerDes power driver level information]. |
| 142 | Modified Table 49 "GMII 1000BASE-T Transmit Signal Parameters" (changed Min values for t1 and t2. |

<table>
<tr><td colspan="2" align="center">**Revision Number: 007**<br>**Revision Date: March 24, 2004**<br>**(Sheet 4 of 5)**</td></tr>
<tr><td>**Page #**</td><td>**Description**</td></tr>
<tr><td>143</td><td>Modified Table 50 "GMII 1000BASE-T Receive Signal Parameters" (changed Min values for t1 and t2.</td></tr>
<tr><td>146</td><td>Replaced old MDIO Timing diagram and table with Figure 43 "MDIO Write Timing Diagram", Figure 44 "MDIO Read Timing Diagram", and Table 52 "MDIO Timing Parameters".</td></tr>
<tr><td>156</td><td>Broke up the old Register Map into Table 59 "MAC Control Registers ($ Port Index + Offset)", Table 60 "MAC RX Statistics Registers ($ Port Index + Offset)", Table 61 "MAC TX Statistics Registers ($ Port Index + Offset)", Table 62 "PHY Autoscan Registers ($ Port Index + Offset)", Table 63 "Global Status and Configuration Registers ($ 0x500 - 0X50C)", Table 64 "RX FIFO Registers ($ 0x580 - 0x5BF)", Table 65 "TX FIFO Registers ($ 0x600 - 0x63E)", Table 66 "MDIO Registers ($ 0x680 - 0x683)", Table 67 "SPI3 Registers ($ 0x700 - 0x716)", Table 68 "SerDes Registers ($ 0x780 - 0x798)", and Table 69 "Optical Module Registers ($ 0x799 - 0x79F)".</td></tr>
<tr><td>159</td><td>Edited Table 63 "Global Status and Configuration Registers ($ 0x500 - 0X50C)" [no offset].</td></tr>
<tr><td>159</td><td>Edited Table 64 "RX FIFO Registers ($ 0x580 - 0x5BF)" [no offset].</td></tr>
<tr><td>160</td><td>Edited Table 65 "TX FIFO Registers ($ 0x600 - 0x63E)" [no offset].</td></tr>
<tr><td>161</td><td>Edited Table 66 "MDIO Registers ($ 0x680 - 0x683)" [no offset].</td></tr>
<tr><td>161</td><td>Edited Table 67 "SPI3 Registers ($ 0x700 - 0x716)" [no offset].</td></tr>
<tr><td>162</td><td>Edited Table 68 "SerDes Registers ($ 0x780 - 0x798)" [no offset].</td></tr>
<tr><td>162</td><td>Edited Table 69 "Optical Module Registers ($ 0x799 - 0x79F)" [no offset].</td></tr>
<tr><td>163</td><td>Modified Table 71 "Desired Duplex ($ Port_Index + 0x02)" [changed 100 Mbps to 1000 Mbps in register description.</td></tr>
<tr><td>167</td><td>Modified Table 82 "MAC IF Mode and RGMII Speed ($ Port_Index + 0x10)" [Added text to register description.]</td></tr>
<tr><td>168</td><td>Modified Table 84 "FC Enable ($ Port_Index + 0x12)" [changed description for bits 1:0].</td></tr>
<tr><td>169</td><td>Modified Table 88 "RX Config Word ($ Port_Index + 0x16)" [edited Register Description text; changed description and type for bits 13:12].</td></tr>
<tr><td>170</td><td>Modified Table 89 "TX Config Word ($ Port_Index + 0x17)" [edited description and type for bits 14, 13:12.</td></tr>
<tr><td>171</td><td>Modified Table 90 "Diverse Config Write ($ Port_Index + 0x18)" [edited description and type for bits 18:8; changed bits 3:1 to Reserved; added table note 2].</td></tr>
<tr><td>172</td><td>Renamed/modified Table 91 "RX Packet Filter Control ($ Port_Index + 0x19)" [old register name - added RX to heading; added table note 2].</td></tr>
<tr><td>174</td><td>Modified Table 93 "MAC RX Statistics ($ Port_Index + 0x20 – + 0x39)" [added note to RxPauseMacControlReceivedCounter description; edited note 3 and added note 4].</td></tr>
<tr><td>178</td><td>Modified Table 94 "MAC TX Statistics ($ Port_Index +0x40 – +0x58)" [changed "1526-max" to "1523 - max frame size" for Txpkts1519toMaxOctets description].</td></tr>
<tr><td>193</td><td>Modified Table 113 "RX FIFO High Watermark Port 0 ($0x580)", Table 114 "RX FIFO High Watermark Port 1 ($0x581)", Table 115 "RX FIFO High Watermark Port 2 ($0x582)", and Table 116 "RX FIFO High Watermark Port 3 ($0x583)" [changed bits 11:0 description].</td></tr>
<tr><td>195</td><td>Renamed and modified Table 121 "RX FIFO Overflow Frame Drop Counter Ports 0 - 3 ($0x594 – 0x597)"<br>[old register name: RX FIFO Number of Frames Removed Ports 0 to 3; renamed bit names to match register names; removed "This register gets updated after one cycle of sw reset is applied" under Description].</td></tr>
<tr><td>196</td><td>Modified Table 123 "RX FIFO Errored Frame Drop Enable ($0x59F)" [renamed bit names to match register name].</td></tr>
</table>

| Revision Number: 007<br>Revision Date: March 24, 2004<br>(Sheet 5 of 5) ||
|---|---|
| **Page #** | **Description** |
| 198 | Renamed/modified Table 125 "RX FIFO Errored Frame Drop Counter Ports 0 - 3 ($0x5A2 - 0x5A5)" on page 198 [older register name: RX FIFO Dropped Packet Counter for Ports 0 to 3; renamed bit names to match register name]. |
| 199 | Modified Table 126 "RX FIFO SPI3 Loopback Enable for Ports 0 - 3 ($0x5B2)" [renamed heading and bit name; changed description and type for bits 7:0]. |
| 201 | Renamed Table 128 "RX FIFO Transfer Threshold Port 0 ($0x5B8)" on page 201 [from "RX FIFO Jumbo Packet Size; changed bit names and edited/added text under description]. |
| 207 | Modified Table 136 "Loop RX Data to TX FIFO (Line-Side Loopback) Ports 0 - 3 ($0x61F)" [renamed heading and bit name]. |
| 208 | Modified Table 138 "TX FIFO Overflow Frame Drop Counter Ports 0 - 3 ($0x621 – 0x624)" [renamed from TX FIFO Number of Frames Removed Ports 3 - 0]. |
| 209 | Modified Table 139 "TX FIFO Errored Frame Drop Counter Ports 0 - 3 ($0x625 – 0x629)" [renamed from TX FIFO Number of Dropped Packets Ports 0-3 and text under the description]. |
| 210 | Modified Table 141 "TX FIFO Port Drop Enable ($0x63D)" [changed description for bits 3:0]. |
| 211 | Modified Table 142 "MDIO Single Command ($0x680)" [changed default; changed description and default for bits 9:8; changed default for bits 4:0]. |
| 212 | Modified Table 144 "Autoscan PHY Address Enable ($0x682)" [added note to register description]. |
| 213 | Modified Table 146 "SPI3 Transmit and Global Configuration ($0x700)" [broke out bits 19:16, 7:4, and 3:0 and changed description text]. |
| 215 | Modified Table 147 "SPI3 Receive Configuration ($0x701)" [broke out bits and modified all text adding SPHY and MPHY modes]. |
| 221 | Modified Table 152 "Clock and Interface Mode Change Enable Ports 0 - 3 ($0x794)" [deleted second paragraph of the Register Description; renamed bits to match caption; changed text under Description]. |
| 222 | Added note under Section 8.4.11, "Optical Module Register Overview". |
| 222 | Modified Table 153 "Optical Module Status Ports 0-3 ($0x799)" [edited register description]. |
| 222 | Modified Table 154 "Optical Module Control Ports 0 - 3 ($0x79A)" [changed register description]. |
| NA | Removed/Reserved Table 190 "TX and RX AC/DC Coupling Selection ($7x780)". |
| NA | Deleted old Figure 19, "Typical GBIC Module Functional Diagram" under Section 5.7, "Optical Module Interface". |
| NA | Removed old Section 5.1.1.5, "Pause Command Frames." |
| 180(old) | Removed old Table 13. TX FIFO Mini Frame Size for MAC and Padding Enable Port 0 to 3 Register (Addr: 0x63E) and replaced with Reserved. |

| Revision Number: 006<br>Revision Date: August 21, 2003<br>(Sheet 1 of 2) ||
|---|---|
| **Page #** | **Description** |
| 19 | Modified Table 1 "IXF1104 Signal Descriptions" |
| 53 | Modified Section 5.1.1.1, "Padding of Undersized Frames on Transmit". |
| 60 | Modified text for etherStatsCollision in Table 9 "RMON Additional Statistics". |
| 87 | Modified Table 17 "IXF1104-to-Optical Module Interface Connections" |

| Revision Number: 006<br>Revision Date: August 21, 2003<br>(Sheet 2 of 2) | |
|---|---|
| **Page #** | **Description** |
| 65 | Modified first paragraph under Section 5.3.1.2, "Clock Rates". |
| 87 | Modified Section 5.8.2.1, "High-Speed Serial Interface". |
| 100 | Modified Figure 27 "Microprocessor — External and Internal Connections". |
| 110 | Changed PECL to LVDS under Section 6.1, "DC Specifications". |
| 113 | Modified table note 4 in Table 32 "SPI3 Receive Interface Signal Parameters". |
| 119 | Modified Table 37 "SerDes Timing Parameters". |
| 125 | Modified Table 40 "Microprocessor Interface Write Cycle AC Signal Parameters". |
| 140 | Modified Table 53 "IPG Receive and Transmit Time Register (Addr: Port_Index + 0x0A – + 0x0C)". |
| 143 | Modified Table 60 "Short Runs Threshold Register (Addr: Port_Index + 0x14)". |
| 143 | Modified Table 61 "Discard Unknown Control Frame Register (Addr: Port_Index + 0x15)". |
| 143 | Modified Table 62 "RX Config Word Register Bit Definition (Addr: Port_Index + 0x16)". |
| 145 | Modified Table 64 "DiverseConfigWrite Register (Addr: Port_Index + 0x18)". |
| 148 | Modified Table 67 "RX Statistics Registers (Addr: Port_Index + 0x20 – + 0x39)". |
| 163 | Modified Table 82 "Microprocessor Interface Register (Addr: 0x508)". |
| 164 | Modified Table 84 "LED Flash Rate Register (Addr: 0x50A)". |
| 169 | Modified Table 93 "RX FIFO Errored Frame Drop Enable Register (Addr: 0x59F)". |
| 170 | Modified Table 96 "RX FIFO Loopback Enable for Ports 0 - 3 Register (Addr: 0x5B2)". |
| 171 | Added Table 98 "RX FIFO Jumbo Packet Size 0-3 Register (Addr: 0x5B8 – 0x5BB". |
| 172 | Added Table 99 "RX FIFO Jumbo Packet Size Port 0 Register Bit Definitions (Addr: 0x5B8)". |
| 172 | Added Table 100 "RX FIFO Jumbo Packet Size Port 1 Register Bit Definitions (Addr: 0x5B9)". |
| 172 | Added Table 101 "RX FIFO Jumbo Packet Size Port 2 Register Bit Definitions (Addr: 0x5BA)". |
| 172 | Added Table 102 "RX FIFO Jumbo Packet Size Port 3 Register Bit Definitions (Addr: 0x5BB)". |
| 178 | Modified Table 110 "TX FIFO Number of Dropped Packets Register Ports 0-3 (Addr: 0x625 – 0x629)". |
| 177 | Modified Table 108 "TX FIFO Port Reset Register (Addr: 0x620)". |
| 177 | Modified Table 108 "TX FIFO Port Reset Register (Addr: 0x620)". |
| 177 | Modified Table 107 "Loop RX Data to TX FIFO Register Ports 0 - 3 (Addr: 0x61F)". |
| 179 | Added Table 111 "TX FIFO Occupancy Counter for Ports 0 - 3 Registers (Addr: 0x62D – 0x630)". |
| 180 | Added Table 112 "TX FIFO Port Drop Enable Register (Addr: 0x63D)". |
| 181 | Modified Table 114 "MDI Single Command Register (Addr: 0x680)". |
| 186 | Added Table 122 "Tx and Rx Power-Down Register (Addr: 0x787)". |
| 194 | Replaced Figure 53 "IXF1104 Example Package Marking". |

| Revision 005 Revision Date: April 30, 2003 | |
| --- | --- |
| **Page #** | **Description** |
| | Initial external release. |

| Revisions 001 through 004 Revision Date: April 2001 – December 2002 | |
| --- | --- |
| **Page #** | **Description** |
| | Internal releases. |

# Contents

# Figures

# Tables

# 1.0 Introduction

This document contains information on the IXF1104 MAC, a four-port Gigabit Media Access Controller that supports IEEE 802.3 10/100/1000 Mbps applications from Cortina Systems, Inc. (Cortina).

## 1.1 What You Will Find in This Document

This document contains the following sections:

- Section 2.0, *General Description*, on page 23 provides the block diagram system architecture.
- Section 3.0, *Ball Assignments and Ball List Tables*, on page 25 shows the signal naming methodology and signal descriptions.
- Section 4.0, *Ball Assignments and Signal Descriptions*, on page 38 illustrates and lists the IXF1104 ball grid diagram with two ball list tables (by signal name and ball location)
- Section 5.0, *Functional Descriptions*, on page 66 gives detailed information about the operation of the IXF1104 including general features, and interface types and descriptions.
- Section 7.0, *Electrical Specifications*, on page 129 provides information on the product-operating parameters, electrical specifications, and timing parameters.
- Section 8.0, *Register Set*, on page 148 illustrates and lists the memory map, detailed descriptions, default values for the register set, and detailed information on each register.
- Section 9.0, *Mechanical Specifications*, on page 216 illustrates the packaging information.
- Section 10.0, *Product Ordering Information*, on page 222 provides ordering information.

## 1.2 Related Documents

| Document |
| --- |
| IXF1104 Media Access Controller Design and Layout Guide |
| IXF1104 Media Access Controller Thermal Design Considerations |
| IXF1104 Media Access Controller Development Kit Manual |
| IXF1104 Media Access Controller Specification Update |

# 2.0    General Description

The IXF1104 MAC provides up to a 4.0 Gbps interface to four individual 10/100/
1000 Mbps full-duplex or 10/100 Mbps half-duplex-capable Ethernet Media Access
Controllers (MACs). The network processor is supported through a System Packet
Interface Phase 3 (SPI3) media interface. The following PHY interfaces are selected on a
per-port basis:

- Serializer/Deserializer (SerDes) with Optical Module Interface support
- Gigabit Media Independent Interface (GMII)
- Reduced Gigabit Media Independent Interface (RGMII).

Figure 1 illustrates the IXF1104 MAC block diagram.

**Figure 1      Block Diagram**
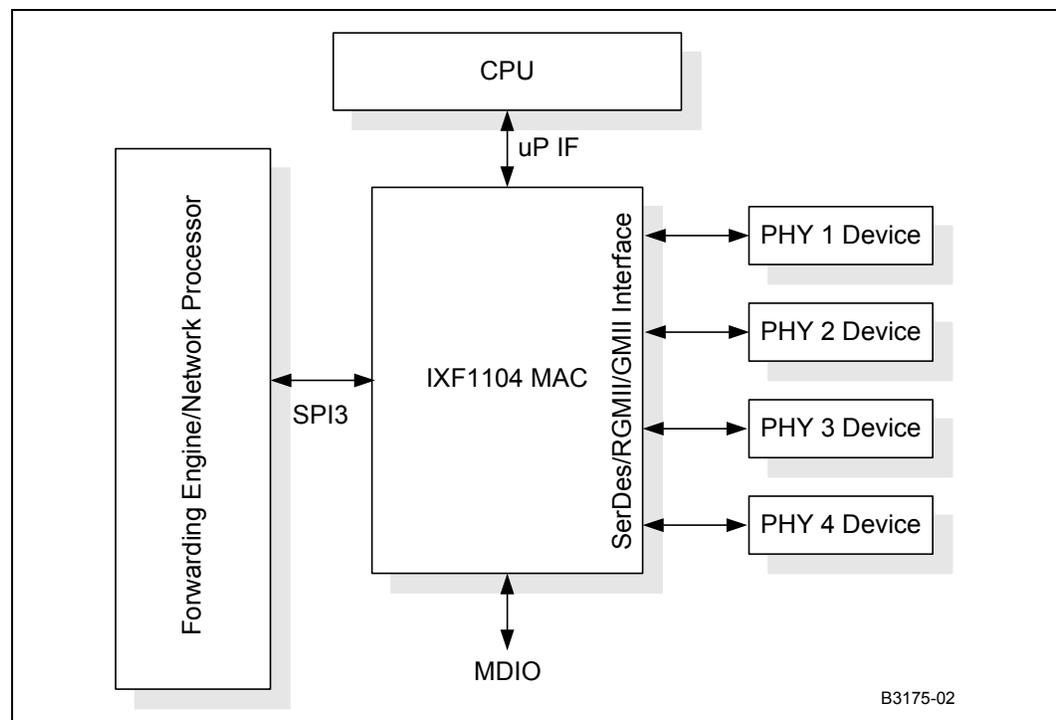


Figure 2 illustrates the IXF1104 MAC internal architecture.

**Figure 2      Internal Architecture**



B3176-01
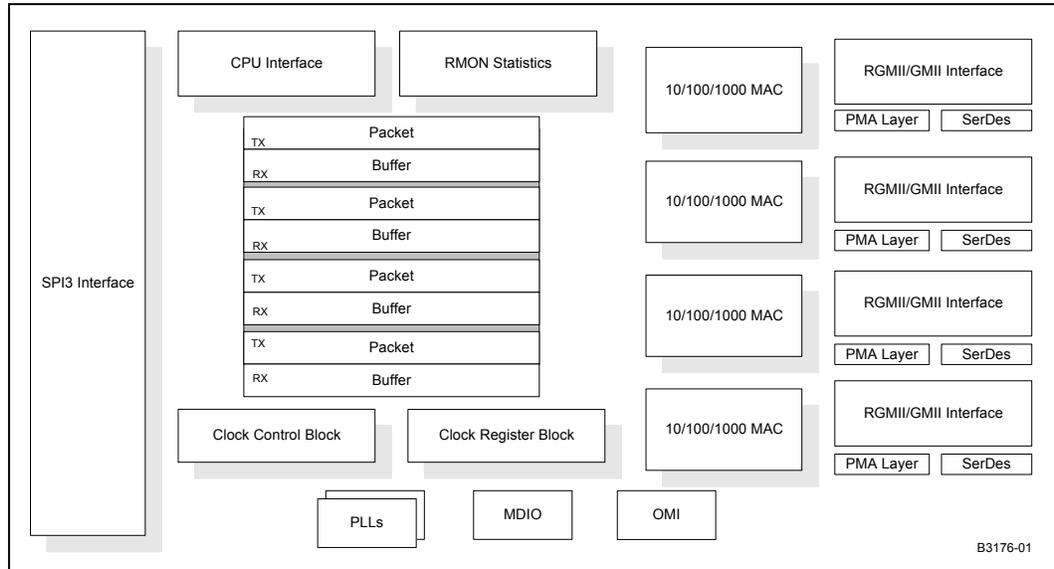
# 3.0 Ball Assignments and Ball List Tables

## 3.1 Ball Assignments

See Figure 3, Table 1, *Ball List in Alphanumeric Order by Signal Name,* on page 26, and Table 2, *Ball List in Alphanumeric Order by Ball Location,* on page 32 for the IXF1104 MAC ball assignments.

**Figure 3    552-Ball CBGA Assignments (Top View)**



B1458-01

## 3.2 Ball List Tables

### 3.2.1 Balls Listed in Alphabetic Order by Signal Name

Table 1 shows the ball locations and signal names arranged in alphanumeric order by signal name.

The following table notes relate to Table 1 and Table 2:

1. **GMII Ball Connection:**
   See Table 16 for connection in RGMII or fiber mode.

2. **SPI3 Ball Connection:**
   See Table 17 for proper SPHY and MPHY connection.

3. **Fiber Mode Ball Connection:**
   See Table 16 for use in RGMII and GMII (copper mode).

**Table 1     Ball List in Alphanumeric Order by Signal Name**

| Signal Name | Ball Location | Signal Name | Ball Location | Signal Name | Ball Location |
|---|---|---|---|---|---|
| AVDD1P8_1 | A5 | GND | D12 | GND | L10 |
| AVDD1P8_1 | A20 | GND | D13 | GND | L12 |
| AVDD1P8_2 | T23 | GND | D17 | GND | L13 |
| AVDD1P8_2 | AB16 | GND | D21 | GND | L15 |
| AVDD2P5_1 | AD20 | GND | F2 | GND | M4 |
| AVDD2P5_2 | R18 | GND | F6 | GND | M8 |
| AVDD2P5_2 | U14 | GND | F10 | GND | M11 |
| CLK125 | AD19 | GND | F15 | GND | M14 |
| COL_0[1] | AB6 | GND | F19 | GND | M17 |
| COL_1[1] | AB10 | GND | F23 | GND | M21 |
| COL_2[1] | AD15 | GND | H4 | GND | N4 |
| COL_3[1] | AB17 | GND | H8 | GND | N8 |
| CRS_0[1] | AA5 | GND | H12 | GND | N11 |
| CRS_1[1] | AA9 | GND | H13 | GND | N14 |
| CRS_2[1] | AB15 | GND | H17 | GND | N17 |
| CRS_3[1] | AC16 | GND | H21 | GND | N21 |
| DTPA_0[2] | D3 | GND | J10 | GND | P10 |
| DTPA_1[2] | L1 | GND | J15 | GND | P12 |
| DTPA_2[2] | A9 | GND | K2 | GND | P13 |
| DTPA_3[2] | J7 | GND | K6 | GND | P15 |
| GND | B6 | GND | K9 | GND | R2 |
| GND | B10 | GND | K11 | GND | R6 |
| GND | B15 | GND | K14 | GND | R9 |
| GND | B19 | GND | K16 | GND | R11 |
| GND | D4 | GND | K19 | GND | R14 |
| GND | D8 | GND | K23 | GND | R16 |

| Signal Name | Ball Location | Signal Name | Ball Location | Signal Name | Ball Location |
|---|---|---|---|---|---|
| GND | R19 | $I^2C\_DATA\_3^3$ | P24 | NC | R20 |
| GND | R23 | LED_CLK | K24 | NC | T6 |
| GND | T10 | LED_DATA | M22 | NC | T7 |
| GND | T15 | LED_LATCH | L22 | NC | T8 |
| GND | U4 | MDC[4] | W24 | NC | T9 |
| GND | U8 | MDIO[4] | V21 | NC | T21 |
| GND | U12 | MOD_DEF_INT | N22 | NC | T22 |
| GND | U13 | NC | D24 | NC | U5 |
| GND | U17 | NC | E12 | NC | U7 |
| GND | U21 | NC | F11 | NC | U9 |
| GND | W2 | NC | G15 | NC | U11 |
| GND | W6 | NC | H7 | NC | U18 |
| GND | W10 | NC | H18 | NC | V9 |
| GND | W15 | NC | J21 | NC | V10 |
| GND | W19 | NC | K7 | NC | V11 |
| GND | W23 | NC | K18 | NC | V13 |
| GND | AA4 | NC | K20 | NC | AB18 |
| GND | AA8 | NC | K22 | NC | AD4 |
| GND | AA12 | NC | L18 | NC | AD5 |
| GND | AA13 | NC | L19 | No Ball | A2 |
| GND | AA17 | NC | L21 | No Ball | A3 |
| GND | AA21 | NC | M7 | No Ball | A22 |
| GND | AC6 | NC | M18 | No Ball | A23 |
| GND | AC10 | NC | M20 | No Ball | A24 |
| GND | AC15 | NC | N3 | No Ball | B1 |
| GND | AC19 | NC | N18 | No Ball | B2 |
| GND | AC14 | NC | P2 | No Ball | B23 |
| GND | L20 | NC | P4 | No Ball | B24 |
| GND | L5 | NC | P6 | No Ball | C1 |
| GND | R7 | NC | P7 | No Ball | C24 |
| GND | AB12 | NC | P8 | No Ball | AB1 |
| GND | A4 | NC | P17 | No Ball | AB24 |
| GND | A21 | NC | P18 | No Ball | AC1 |
| GND | AD21 | NC | R5 | No Ball | AC2 |
| $I^2C\_CLK$ | L23 | NC | R10 | No Ball | AC23 |
| $I^2C\_DATA\_0^3$ | L24 | NC | R12 | No Ball | AC24 |
| $I^2C\_DATA\_1^3$ | M24 | NC | R13 | No Ball | AD1 |
| $I^2C\_DATA\_2^3$ | N24 | NC | R15 | No Ball | AD2 |

| Signal Name | Ball Location | Signal Name | Ball Location | Signal Name | Ball Location |
|---|---|---|---|---|---|
| No Ball | AD3 | RENB_0[2] | A13 | RX_N_1[3] | U22 |
| No Ball | AD22 | RENB_1[2] | A18 | RX_N_2[3] | R24 |
| No Ball | AD23 | RENB_2[2] | C19 | RX_N_3[3] | V24 |
| No Ball | AD24 | RENB_3[2] | E24 | RX_P_0[3] | P22 |
| No Pad | A1 | REOP_0[2] | C16 | RX_P_1[3] | V22 |
| PTPA[2] | B11 | REOP_1[2] | D18 | RX_P_2[3] | T24 |
| RDAT_0[2] | A15 | REOP_2[2] | C23 | RX_P_3[3] | U24 |
| RDAT_1[2] | A14 | REOP_3[2] | J19 | RXC_0[1] | V4 |
| RDAT_2[2] | B14 | RERR_0[2] | A16 | RXC_1[1] | AD11 |
| RDAT_3[2] | C14 | RERR_1[2] | G17 | RXC_2[1] | AA24 |
| RDAT_4[2] | C13 | RERR_2[2] | D20 | RXC_3[1] | V23 |
| RDAT_5[2] | D14 | RERR_3[2] | H20 | RXD0_0[1] | V8 |
| RDAT_6[2] | E14 | RFCLK[2] | A19 | RXD0_1[1] | Y9 |
| RDAT_7[2] | F14 | RMOD0[2] | G14 | RXD0_2[1] | Y20 |
| RDAT_8[2] | A17 | RMOD1[2] | G13 | RXD0_3[1] | Y17 |
| RDAT_9[2] | C17 | RPRTY_0[2] | E15 | RXD1_0[1] | V7 |
| RDAT_10[2] | D16 | RPRTY_1[2] | G16 | RXD1_1[1] | Y11 |
| RDAT_11[2] | E16 | RPRTY_2[2] | E20 | RXD1_2[1] | Y21 |
| RDAT_12[2] | F16 | RPRTY_3[2] | F20 | RXD1_3[1] | Y18 |
| RDAT_13[2] | E17 | RSOP_0[2] | B16 | RXD2_0[1] | W7 |
| RDAT_14[2] | E18 | RSOP_1[2] | C18 | RXD2_1[1] | W11 |
| RDAT_15[2] | F18 | RSOP_2[2] | E23 | RXD2_2[1] | Y22 |
| RDAT_16[2] | B20 | RSOP_3[2] | J18 | RXD2_3[1] | Y19 |
| RDAT_17[2] | B22 | RSX[2] | E13 | RXD3_0[1] | Y7 |
| RDAT_18[2] | C20 | RVAL_0[2] | C15 | RXD3_1[1] | W9 |
| RDAT_19[2] | C21 | RVAL_1[2] | B18 | RXD3_2[1] | Y23 |
| RDAT_20[2] | C22 | RVAL_2[2] | E19 | RXD3_3[1] | W18 |
| RDAT_21[2] | D22 | RVAL_3[2] | F22 | RXD4_0[1] | Y6 |
| RDAT_22[2] | E22 | RX_DV_0[1] | V5 | RXD4_1[1] | AD10 |
| RDAT_23[2] | E21 | RX_DV_1[1] | AB11 | RXD4_2[1] | W22 |
| RDAT_24[2] | G18 | RX_DV_2[1] | Y24 | RXD4_3[1] | T16 |
| RDAT_25[2] | G19 | RX_DV_3[1] | V18 | RXD5_0[1] | Y5 |
| RDAT_26[2] | G20 | RX_ER_0[1] | W5 | RXD5_1[1] | AC11 |
| RDAT_27[2] | G21 | RX_ER_1[1] | Y12 | RXD5_2[1] | V20 |
| RDAT_28[2] | G22 | RX_ER_2[1] | AA22 | RXD5_3[1] | T17 |
| RDAT_29[2] | G23 | RX_ER_3[1] | U20 | RXD6_0[1] | AB5 |
| RDAT_30[2] | G24 | RX_LOS_INT[3] | P19 | RXD6_1[1] | AA11 |
| RDAT_31[2] | F24 | RX_N_0[3] | R22 | RXD6_2[1] | V19 |

| Signal Name | Ball Location | Signal Name | Ball Location | Signal Name | Ball Location |
|---|---|---|---|---|---|
| RXD6_3[1] | T18 | TDAT28[2] | G8 | TX_ER_2[1] | AD17 |
| RXD7_0[1] | AC5 | TDAT29[2] | G9 | TX_ER_3[1] | AB13 |
| RXD7_1[1] | Y10 | TDAT30[2] | F5 | TX_FAULT_INT[3] | P23 |
| RXD7_2[1] | W20 | TDAT31[2] | F7 | TX_N_0[3] | Y14 |
| RXD7_3[1] | T19 | TDI | J24 | TX_N_1[3] | AD14 |
| STPA[2] | C11 | TDO | H24 | TX_N_2[3] | Y16 |
| SYS_RST_L | AD12 | TENB_0[2] | B7 | TX_N_3[3] | AD18 |
| TADR0[2] | A11 | TENB_1[2] | E2 | TX_P_0[3] | Y13 |
| TADR1[2] | A12 | TENB_2[2] | C9 | TX_P_1[3] | AD13 |
| TCLK | J22 | TENB_3[2] | J4 | TX_P_2[3] | W16 |
| TDAT0[2] | B3 | TEOP_0[2] | A7 | TX_P_3[3] | AC18 |
| TDAT1[2] | C2 | TEOP_1[2] | F3 | TXC_0[1] | AA1 |
| TDAT2[2] | C3 | TEOP_2[2] | E4 | TXC_1[1] | AD7 |
| TDAT3[2] | D1 | TEOP_3[2] | H5 | TXC_2[1] | AC20 |
| TDAT4[2] | C4 | TERR_0[2] | A8 | TXC_3[1] | AB14 |
| TDAT5[2] | C5 | TERR_1[2] | K1 | TXD0_0[1] | Y1 |
| TDAT6[2] | B5 | TERR_2[2] | E11 | TXD0_1[1] | AC7 |
| TDAT7[2] | C6 | TERR_3[2] | J8 | TXD0_2[1] | AB20 |
| TDAT8[2] | F1 | TFCLK[2] | D7 | TXD0_3[1] | V14 |
| TDAT9[2] | G1 | TMOD0[2] | A6 | TXD1_0[1] | Y2 |
| TDAT10[2] | G2 | TMOD1[2] | D9 | TXD1_1[1] | AB7 |
| TDAT11[2] | H1 | TMS | H22 | TXD1_2[1] | AB21 |
| TDAT12[2] | J1 | TPRTY_0[2] | D5 | TXD1_3[1] | V15 |
| TDAT13[2] | J2 | TPRTY_1[2] | G3 | TXD2_0[1] | Y3 |
| TDAT14[2] | J3 | TPRTY_2[2] | B9 | TXD2_1[1] | AB9 |
| TDAT15[2] | H3 | TPRTY_3[2] | J6 | TXD2_2[1] | AB22 |
| TDAT16[2] | E5 | TRST_L | J23 | TXD2_3[1] | V16 |
| TDAT17[2] | E6 | TSOP_0[2] | C7 | TXD3_0[1] | AA3 |
| TDAT18[2] | E7 | TSOP_1[2] | E3 | TXD3_1[1] | AD9 |
| TDAT19[2] | E8 | TSOP_2[2] | C10 | TXD3_2[1] | AB23 |
| TDAT20[2] | E9 | TSOP_3[2] | J5 | TXD3_3[1] | V17 |
| TDAT21[2] | E10 | TSX | E1 | TXD4_0[1] | AB3 |
| TDAT22[2] | F9 | TX_EN_0[1] | AB2 | TXD4_1[1] | AA7 |
| TDAT23[2] | C8 | TX_EN_1[1] | Y8 | TXD4_2[1] | AD16 |
| TDAT24[2] | G4 | TX_EN_2[1] | AC22 | TXD4_3[1] | AA14 |
| TDAT25[2] | G5 | TX_EN_3[1] | V12 | TXD5_0[1] | AC3 |
| TDAT26[2] | G6 | TX_ER_0[1] | W1 | TXD5_1[1] | AB8 |
| TDAT27[2] | G7 | TX_ER_1[1] | AD6 | TXD5_2[1] | AB19 |

| Signal Name | Ball Location | Signal Name | Ball Location | Signal Name | Ball Location |
|---|---|---|---|---|---|
| TXD5_3[1] | Y15 | UPX_DATA11 | N7 | VDD | K17 |
| TXD6_0[1] | AB4 | UPX_DATA12 | L8 | VDD | K21 |
| TXD6_1[1] | AD8 | UPX_DATA13 | H9 | VDD | L9 |
| TXD6_2[1] | AA20 | UPX_DATA14 | J9 | VDD | L11 |
| TXD6_3[1] | AA16 | UPX_DATA15 | N10 | VDD | L14 |
| TXD7_0[1] | Y4 | UPX_DATA16 | M10 | VDD | L16 |
| TXD7_1[1] | AC9 | UPX_DATA17 | K10 | VDD | P9 |
| TXD7_2[1] | AA18 | UPX_DATA18 | G10 | VDD | P11 |
| TXD7_3[1] | W14 | UPX_DATA19 | H11 | VDD | P14 |
| TXPAUSE_ADD0 | N20 | UPX_DATA20 | G11 | VDD | P16 |
| TXPAUSE_ADD1 | P20 | UPX_DATA21 | K12 | VDD | R4 |
| TXPAUSE_ADD2 | P21 | UPX_DATA22 | G12 | VDD | R8 |
| TXPAUSEFR | T20 | UPX_DATA23 | K13 | VDD | R17 |
| UPX_ADD0 | P3 | UPX_DATA24 | H14 | VDD | R21 |
| UPX_ADD1 | N1 | UPX_DATA25 | K15 | VDD | T11 |
| UPX_ADD2 | P1 | UPX_DATA26 | N15 | VDD | T14 |
| UPX_ADD3 | R1 | UPX_DATA27 | M15 | VDD | U10 |
| UPX_ADD4 | T1 | UPX_DATA28 | J16 | VDD | U15 |
| UPX_ADD5 | U1 | UPX_DATA29 | H16 | VDD | W4 |
| UPX_ADD6 | V1 | UPX_DATA30 | J17 | VDD | W21 |
| UPX_ADD7 | V2 | UPX_DATA31 | L17 | VDD | AA6 |
| UPX_ADD8 | V3 | uPx_Rd_L | V6 | VDD | AA10 |
| UPX_ADD9 | U3 | uPx_Rdy_L | M1 | VDD | AA15 |
| UPX_ADD10 | T3 | UPX_WIDTH0 | U16 | VDD | AA19 |
| UPX_BADD0 | T2 | UPX_WIDTH1 | T5 | VDD | C12 |
| UPX_BADD1 | W3 | uPx_Wr_L | T4 | VDD | D11 |
| uPx_Cs_L | R3 | VDD | D6 | VDD | J20 |
| UPX_DATA0 | L2 | VDD | D10 | VDD | A10 |
| UPX_DATA1 | K3 | VDD | D15 | VDD2 | B4 |
| UPX_DATA2 | L3 | VDD | D19 | VDD2 | B8 |
| UPX_DATA3 | M3 | VDD | F4 | VDD2 | B12 |
| UPX_DATA4 | L4 | VDD | F21 | VDD2 | D2 |
| UPX_DATA5 | N5 | VDD | H10 | VDD2 | F8 |
| UPX_DATA6 | M5 | VDD | H15 | VDD2 | F12 |
| UPX_DATA7 | K5 | VDD | J11 | VDD2 | H2 |
| UPX_DATA8 | P5 | VDD | J14 | VDD2 | H6 |
| UPX_DATA9 | L6 | VDD | K4 | VDD2 | J12 |
| UPX_DATA10 | L7 | VDD | K8 | VDD2 | M2 |

| Signal Name | Ball Location |
|:---:|:---:|
| VDD2 | M6 |
| VDD2 | M9 |
| VDD2 | M12 |
| VDD3 | B13 |
| VDD3 | B17 |
| VDD3 | B21 |
| VDD3 | D23 |
| VDD3 | F13 |
| VDD3 | F17 |
| VDD3 | H19 |
| VDD3 | H23 |
| VDD3 | J13 |
| VDD3 | M13 |
| VDD3 | M16 |
| VDD3 | M19 |
| VDD3 | M23 |
| VDD4 | N13 |
| VDD4 | N16 |
| VDD4 | N19 |
| VDD4 | N23 |
| VDD4 | T13 |
| VDD4 | U19 |
| VDD4 | U23 |
| VDD4 | W13 |
| VDD4 | W17 |
| VDD4 | AA23 |
| VDD4 | AC13 |
| VDD4 | AC17 |
| VDD4 | AC21 |
| VDD5 | N2 |
| VDD5 | N6 |
| VDD5 | N9 |
| VDD5 | N12 |
| VDD5 | T12 |
| VDD5 | U2 |
| VDD5 | U6 |
| VDD5 | W8 |
| VDD5 | W12 |

| Signal Name | Ball Location |
|:---:|:---:|
| VDD5 | AA2 |
| VDD5 | AC4 |
| VDD5 | AC8 |
| VDD5 | AC12 |

## 3.2.2 Balls Listed in Alphabetic Order by Ball Location

Table 2 shows the ball locations and signal names arranged in order by ball location.

**Table 2    Ball List in Alphanumeric Order by Ball Location**

| Ball Location | Signal Name | Ball Location | Signal Name | Ball Location | Signal Name |
|---|---|---|---|---|---|
| A1 | No Pad | B12 | VDD2 | C23 | REOP_2[2] |
| A2 | No Ball | B13 | VDD3 | C24 | No Ball |
| A3 | No Ball | B14 | RDAT_2[2] | D1 | TDAT3[2] |
| A4 | GND | B15 | GND | D2 | VDD2 |
| A5 | AVDD1P8_1 | B16 | RSOP_0[2] | D3 | DTPA_0[2] |
| A6 | TMOD0[2] | B17 | VDD3 | D4 | GND |
| A7 | TEOP_0[2] | B18 | RVAL_1[2] | D5 | TPRTY_0[2] |
| A8 | TERR_0[2] | B19 | GND | D6 | VDD |
| A9 | DTPA_2[2] | B20 | RDAT_16[2] | D7 | TFCLK[2] |
| A10 | VDD | B21 | VDD3 | D8 | GND |
| A11 | TADR0[2] | B22 | RDAT_17[2] | D9 | TMOD1[2] |
| A12 | TADR1[2] | B23 | No Ball | D10 | VDD |
| A13 | RENB_0[2] | B24 | No Ball | D11 | VDD |
| A14 | RDAT_1[2] | C1 | No Ball | D12 | GND |
| A15 | RDAT_0[2] | C2 | TDAT1[2] | D13 | GND |
| A16 | RERR_0[2] | C3 | TDAT2[2] | D14 | RDAT_5[2] |
| A17 | RDAT_8[2] | C4 | TDAT4[2] | D15 | VDD |
| A18 | RENB_1[2] | C5 | TDAT5[2] | D16 | RDAT_10[2] |
| A19 | RFCLK[2] | C6 | TDAT7[2] | D17 | GND |
| A20 | AVDD1P8_1 | C7 | TSOP_0[2] | D18 | REOP_1[2] |
| A21 | GND | C8 | TDAT23[2] | D19 | VDD |
| A22 | No Ball | C9 | TENB_2[2] | D20 | RERR_2[2] |
| A23 | No Ball | C10 | TSOP_2[2] | D21 | GND |
| A24 | No Ball | C11 | STPA[2] | D22 | RDAT_21[2] |
| B1 | No Ball | C12 | VDD | D23 | VDD3 |
| B2 | No Ball | C13 | RDAT_4[2] | D24 | NC |
| B3 | TDAT0[2] | C14 | RDAT_3[2] | E1 | TSX |
| B4 | VDD2 | C15 | RVAL_0[2] | E2 | TENB_1[2] |
| B5 | TDAT6[2] | C16 | REOP_0[2] | E3 | TSOP_1[2] |
| B6 | GND | C17 | RDAT_9[2] | E4 | TEOP_2[2] |
| B7 | TENB_0[2] | C18 | RSOP_1[2] | E5 | TDAT16[2] |
| B8 | VDD2 | C19 | RENB_2[2] | E6 | TDAT17[2] |
| B9 | TPRTY_2[2] | C20 | RDAT_18[2] | E7 | TDAT18[2] |
| B10 | GND | C21 | RDAT_19[2] | E8 | TDAT19[2] |
| B11 | PTPA[2] | C22 | RDAT_20[2] | E9 | TDAT20[2] |

| Ball Location | Signal Name | Ball Location | Signal Name | Ball Location | Signal Name |
|---|---|---|---|---|---|
| E10 | TDAT21$^2$ | F24 | RDAT_31$^2$ | H14 | UPX_DATA24 |
| E11 | TERR_2$^2$ | G1 | TDAT9$^2$ | H15 | VDD |
| E12 | NC | G2 | TDAT10$^2$ | H16 | UPX_DATA29 |
| E13 | RSX$^2$ | G3 | TPRTY_1$^2$ | H17 | GND |
| E14 | RDAT_6$^2$ | G4 | TDAT24$^2$ | H18 | NC |
| E15 | RPRTY_0$^2$ | G5 | TDAT25$^2$ | H19 | VDD3 |
| E16 | RDAT_11$^2$ | G6 | TDAT26$^2$ | H20 | RERR_3$^2$ |
| E17 | RDAT_13$^2$ | G7 | TDAT27$^2$ | H21 | GND |
| E18 | RDAT_14$^2$ | G8 | TDAT28$^2$ | H22 | TMS |
| E19 | RVAL_2$^2$ | G9 | TDAT29$^2$ | H23 | VDD3 |
| E20 | RPRTY_2$^2$ | G10 | UPX_DATA18 | H24 | TDO |
| E21 | RDAT_23$^2$ | G11 | UPX_DATA20 | J1 | TDAT12$^2$ |
| E22 | RDAT_22$^2$ | G12 | UPX_DATA22 | J2 | TDAT13$^2$ |
| E23 | RSOP_2$^2$ | G13 | RMOD1$^2$ | J3 | TDAT14$^2$ |
| E24 | RENB_3$^2$ | G14 | RMOD0$^2$ | J4 | TENB_3$^2$ |
| F1 | TDAT8$^2$ | G15 | NC | J5 | TSOP_3$^2$ |
| F2 | GND | G16 | RPRTY_1$^2$ | J6 | TPRTY_3$^2$ |
| F3 | TEOP_1$^2$ | G17 | RERR_1$^2$ | J7 | DTPA_3$^2$ |
| F4 | VDD | G18 | RDAT_24$^2$ | J8 | TERR_3$^2$ |
| F5 | TDAT30$^2$ | G19 | RDAT_25$^2$ | J9 | UPX_DATA14 |
| F6 | GND | G20 | RDAT_26$^2$ | J10 | GND |
| F7 | TDAT31$^2$ | G21 | RDAT_27$^2$ | J11 | VDD |
| F8 | VDD2 | G22 | RDAT_28$^2$ | J12 | VDD2 |
| F9 | TDAT22$^2$ | G23 | RDAT_29$^2$ | J13 | VDD3 |
| F10 | GND | G24 | RDAT_30$^2$ | J14 | VDD |
| F11 | NC | H1 | TDAT11$^2$ | J15 | GND |
| F12 | VDD2 | H2 | VDD2 | J16 | UPX_DATA28 |
| F13 | VDD3 | H3 | TDAT15$^2$ | J17 | UPX_DATA30 |
| F14 | RDAT_7$^2$ | H4 | GND | J18 | RSOP_3$^2$ |
| F15 | GND | H5 | TEOP_3$^2$ | J19 | REOP_3$^2$ |
| F16 | RDAT_12$^2$ | H6 | VDD2 | J20 | VDD |
| F17 | VDD3 | H7 | NC | J21 | NC |
| F18 | RDAT_15$^2$ | H8 | GND | J22 | TCLK |
| F19 | GND | H9 | UPX_DATA13 | J23 | TRST_L |
| F20 | RPRTY_3$^2$ | H10 | VDD | J24 | TDI |
| F21 | VDD | H11 | UPX_DATA19 | K1 | TERR_1$^2$ |
| F22 | RVAL_3$^2$ | H12 | GND | K2 | GND |
| F23 | GND | H13 | GND | K3 | UPX_DATA1 |

| Ball Location | Signal Name | Ball Location | Signal Name | Ball Location | Signal Name |
|---|---|---|---|---|---|
| K4 | VDD | L18 | NC | N8 | GND |
| K5 | UPX_DATA7 | L19 | NC | N9 | VDD5 |
| K6 | GND | L20 | GND | N10 | UPX_DATA15 |
| K7 | NC | L21 | NC | N11 | GND |
| K8 | VDD | L22 | LED_LATCH | N12 | VDD5 |
| K9 | GND | L23 | $I^2C\_CLK$ | N13 | VDD4 |
| K10 | UPX_DATA17 | L24 | $I^2C\_DATA\_0$[3] | N14 | GND |
| K11 | GND | M1 | uPx_Rdy_L | N15 | UPX_DATA26 |
| K12 | UPX_DATA21 | M2 | VDD2 | N16 | VDD4 |
| K13 | UPX_DATA23 | M3 | UPX_DATA3 | N17 | GND |
| K14 | GND | M4 | GND | N18 | NC |
| K15 | UPX_DATA25 | M5 | UPX_DATA6 | N19 | VDD4 |
| K16 | GND | M6 | VDD2 | N20 | TXPAUSE_ADD0 |
| K17 | VDD | M7 | NC | N21 | GND |
| K18 | NC | M8 | GND | N22 | MOD_DEF_INT |
| K19 | GND | M9 | VDD2 | N23 | VDD4 |
| K20 | NC | M10 | UPX_DATA16 | N24 | $I^2C\_DATA\_2$[3] |
| K21 | VDD | M11 | GND | P1 | UPX_ADD2 |
| K22 | NC | M12 | VDD2 | P2 | NC |
| K23 | GND | M13 | VDD3 | P3 | UPX_ADD0 |
| K24 | LED_CLK | M14 | GND | P4 | NC |
| L1 | DTPA_1[2] | M15 | UPX_DATA27 | P5 | UPX_DATA8 |
| L2 | UPX_DATA0 | M16 | VDD3 | P6 | NC |
| L3 | UPX_DATA2 | M17 | GND | P7 | NC |
| L4 | UPX_DATA4 | M18 | NC | P8 | NC |
| L5 | GND | M19 | VDD3 | P9 | VDD |
| L6 | UPX_DATA9 | M20 | NC | P10 | GND |
| L7 | UPX_DATA10 | M21 | GND | P11 | VDD |
| L8 | UPX_DATA12 | M22 | LED_DATA | P12 | GND |
| L9 | VDD | M23 | VDD3 | P13 | GND |
| L10 | GND | M24 | $I^2C\_DATA\_1$[3] | P14 | VDD |
| L11 | VDD | N1 | UPX_ADD1 | P15 | GND |
| L12 | GND | N2 | VDD5 | P16 | VDD |
| L13 | GND | N3 | NC | P17 | NC |
| L14 | VDD | N4 | GND | P18 | NC |
| L15 | GND | N5 | UPX_DATA5 | P19 | RX_LOS_INT[3] |
| L16 | VDD | N6 | VDD5 | P20 | TXPAUSE_ADD1 |
| L17 | UPX_DATA31 | N7 | UPX_DATA11 | P21 | TXPAUSE_ADD2 |

| Ball Location | Signal Name | Ball Location | Signal Name | Ball Location | Signal Name |
|---|---|---|---|---|---|
| P22 | RX_P_0[3] | T12 | VDD5 | V2 | UPX_ADD7 |
| P23 | TX_FAULT_INT[3] | T13 | VDD4 | V3 | UPX_ADD8 |
| P24 | I$^2$C_DATA_3[3] | T14 | VDD | V4 | RXC_0[1] |
| R1 | UPX_ADD3 | T15 | GND | V5 | RX_DV_0[1] |
| R2 | GND | T16 | RXD4_3[1] | V6 | uPx_Rd_L |
| R3 | uPx_Cs_L | T17 | RXD5_3[1] | V7 | RXD1_0[1] |
| R4 | VDD | T18 | RXD6_3[1] | V8 | RXD0_0[1] |
| R5 | NC | T19 | RXD7_3[1] | V9 | NC |
| R6 | GND | T20 | TXPAUSEFR | V10 | NC |
| R7 | GND | T21 | NC | V11 | NC |
| R8 | VDD | T22 | NC | V12 | TX_EN_3[1] |
| R9 | GND | T23 | AVDD1P8_2 | V13 | NC |
| R10 | NC | T24 | RX_P_2[3] | V14 | TXD0_3[1] |
| R11 | GND | U1 | UPX_ADD5 | V15 | TXD1_3[1] |
| R12 | NC | U2 | VDD5 | V16 | TXD2_3[1] |
| R13 | NC | U3 | UPX_ADD9 | V17 | TXD3_3[1] |
| R14 | GND | U4 | GND | V18 | RX_DV_3[1] |
| R15 | NC | U5 | NC | V19 | RXD6_2[1] |
| R16 | GND | U6 | VDD5 | V20 | RXD5_2[1] |
| R17 | VDD | U7 | NC | V21 | MDIO[4] |
| R18 | AVDD2P5_2 | U8 | GND | V22 | RX_P_1[3] |
| R19 | GND | U9 | NC | V23 | RXC_3[1] |
| R20 | NC | U10 | VDD | V24 | RX_N_3[3] |
| R21 | VDD | U11 | NC | W1 | TX_ER_0[1] |
| R22 | RX_N_0[3] | U12 | GND | W2 | GND |
| R23 | GND | U13 | GND | W3 | UPX_BADD1 |
| R24 | RX_N_2[3] | U14 | AVDD2P5_2 | W4 | VDD |
| T1 | UPX_ADD4 | U15 | VDD | W5 | RX_ER_0[1] |
| T2 | UPX_BADD0 | U16 | UPX_WIDTH0 | W6 | GND |
| T3 | UPX_ADD10 | U17 | GND | W7 | RXD2_0[1] |
| T4 | uPx_Wr_L | U18 | NC | W8 | VDD5 |
| T5 | UPX_WIDTH1 | U19 | VDD4 | W9 | RXD3_1[1] |
| T6 | NC | U20 | RX_ER_3[1] | W10 | GND |
| T7 | NC | U21 | GND | W11 | RXD2_1[1] |
| T8 | NC | U22 | RX_N_1[3] | W12 | VDD5 |
| T9 | NC | U23 | VDD4 | W13 | VDD4 |
| T10 | GND | U24 | RX_P_3[3] | W14 | TXD7_3[1] |
| T11 | VDD | V1 | UPX_ADD6 | W15 | GND |

| Ball Location | Signal Name | Ball Location | Signal Name | Ball Location | Signal Name |
|---|---|---|---|---|---|
| W16 | TX_P_2[3] | AA6 | VDD | AB20 | TXD0_2[1] |
| W17 | VDD4 | AA7 | TXD4_1[1] | AB21 | TXD1_2[1] |
| W18 | RXD3_3[1] | AA8 | GND | AB22 | TXD2_2[1] |
| W19 | GND | AA9 | CRS_1[1] | AB23 | TXD3_2[1] |
| W20 | RXD7_2[1] | AA10 | VDD | AB24 | No Ball |
| W21 | VDD | AA11 | RXD6_1[1] | AC1 | No Ball |
| W22 | RXD4_2[1] | AA12 | GND | AC2 | No Ball |
| W23 | GND | AA13 | GND | AC3 | TXD5_0[1] |
| W24 | MDC[4] | AA14 | TXD4_3[1] | AC4 | VDD5 |
| Y1 | TXD0_0[1] | AA15 | VDD | AC5 | RXD7_0[1] |
| Y2 | TXD1_0[1] | AA16 | TXD6_3[1] | AC6 | GND |
| Y3 | TXD2_0[1] | AA17 | GND | AC7 | TXD0_1[1] |
| Y4 | TXD7_0[1] | AA18 | TXD7_2[1] | AC8 | VDD5 |
| Y5 | RXD5_0[1] | AA19 | VDD | AC9 | TXD7_1[1] |
| Y6 | RXD4_0[1] | AA20 | TXD6_2[1] | AC10 | GND |
| Y7 | RXD3_0[1] | AA21 | GND | AC11 | RXD5_1[1] |
| Y8 | TX_EN_1[1] | AA22 | RX_ER_2[1] | AC12 | VDD5 |
| Y9 | RXD0_1[1] | AA23 | VDD4 | AC13 | VDD4 |
| Y10 | RXD7_1[1] | AA24 | RXC_2[1] | AC14 | GND |
| Y11 | RXD1_1[1] | AB1 | No Ball | AC15 | GND |
| Y12 | RX_ER_1[1] | AB2 | TX_EN_0[1] | AC16 | CRS_3[1] |
| Y13 | TX_P_0[3] | AB3 | TXD4_0[1] | AC17 | VDD4 |
| Y14 | TX_N_0[3] | AB4 | TXD6_0[1] | AC18 | TX_P_3[3] |
| Y15 | TXD5_3[1] | AB5 | RXD6_0[1] | AC19 | GND |
| Y16 | TX_N_2[3] | AB6 | COL_0[1] | AC20 | TXC_2[1] |
| Y17 | RXD0_3[1] | AB7 | TXD1_1[1] | AC21 | VDD4 |
| Y18 | RXD1_3[1] | AB8 | TXD5_1[1] | AC22 | TX_EN_2[1] |
| Y19 | RXD2_3[1] | AB9 | TXD2_1[1] | AC23 | No Ball |
| Y20 | RXD0_2[1] | AB10 | COL_1[1] | AC24 | No Ball |
| Y21 | RXD1_2[1] | AB11 | RX_DV_1[1] | AD1 | No Ball |
| Y22 | RXD2_2[1] | AB12 | GND | AD2 | No Ball |
| Y23 | RXD3_2[1] | AB13 | TX_ER_3[1] | AD3 | No Ball |
| Y24 | RX_DV_2[1] | AB14 | TXC_3[1] | AD4 | NC |
| AA1 | TXC_0[1] | AB15 | CRS_2[1] | AD5 | NC |
| AA2 | VDD5 | AB16 | AVDD1P8_2 | AD6 | TX_ER_1[1] |
| AA3 | TXD3_0[1] | AB17 | COL_3[1] | AD7 | TXC_1[1] |
| AA4 | GND | AB18 | NC | AD8 | TXD6_1[1] |
| AA5 | CRS_0[1] | AB19 | TXD5_2[1] | AD9 | TXD3_1[1] |

| Ball Location | Signal Name |
|---|---|
| AD10 | RXD4_1[1] |
| AD11 | RXC_1[1] |
| AD12 | SYS_RST_L |
| AD13 | TX_P_1[3] |
| AD14 | TX_N_1[3] |
| AD15 | COL_2[1] |
| AD16 | TXD4_2[1] |
| AD17 | TX_ER_2[1] |
| AD18 | TX_N_3[3] |
| AD19 | CLK125 |
| AD20 | AVDD2P5_1 |
| AD21 | GND |
| AD22 | No Ball |
| AD23 | No Ball |
| AD24 | No Ball |

# 4.0 Ball Assignments and Signal Descriptions

## 4.1 Naming Conventions

### 4.1.1 Signal Name Conventions

Signal names begin with a Signal Mnemonic, and can also contain one or more of the following designations: a differential pair designation, a serial designation, a port designation (RGMII interface), and an active low designation. Signal naming conventions are as follows:

**Differential Pair + Port Designation.** The positive and negative components of differential pairs tied to a specific port are designated by the Signal Mnemonic, immediately followed by an underscore and either P (positive component) or N (negative component), and an underscore followed by the port designation. For example, SerDes interface signals for port 0 are identified as TX_P_0 and TX_N_0.

**Serial Designation.** A set of signals that are not tied to any specific port are designated by the Signal Mnemonic, followed by a bracketed serial designation. For example, the set of 11 CPU Address Bus signals is identified as UPX_ADD[10:0].

**Port Designation**. Individual signals that apply to a particular port are designated by the Signal Mnemonic, immediately followed by an underscore and the Port Designation. For example, RGMII Transmit Control signals are identified as TX_CTL_0, TX_CTL_1, TX_CTL_2, and so on.

**Port Bus Designation.** A set of bus signals that apply to a particular port are designated by the Signal Mnemonic, immediately followed by a bracketed bus designation, followed by an underscore and the port designation. For example, RGMII transmit data bus signals are identified as TD[3:0]_0, TD[3:0]_1, TD[3:0]_2, and so on.

**Active Low Designation.** A control input or indicator output that is active Low is designated by a final suffix consisting of an underscore followed by an upper case "L". For example, the CPU cycle complete identifier is shown as UPX_RDY_L.

### 4.1.2 Register Address Conventions

Registers located in on-chip memory are accessed using a register address, which is provided in Hex notation. A Register Address is indicated by the dollar sign ($), followed by the memory location in Hex.

## 4.2 Interface Signal Groups

This section describes the IXF1104 MAC signals in groups according to the associated interface or function. Figure 4 shows the various interfaces available on the IXF1104 MAC.

**Figure 4**    **Interface Signals**



B3181-02
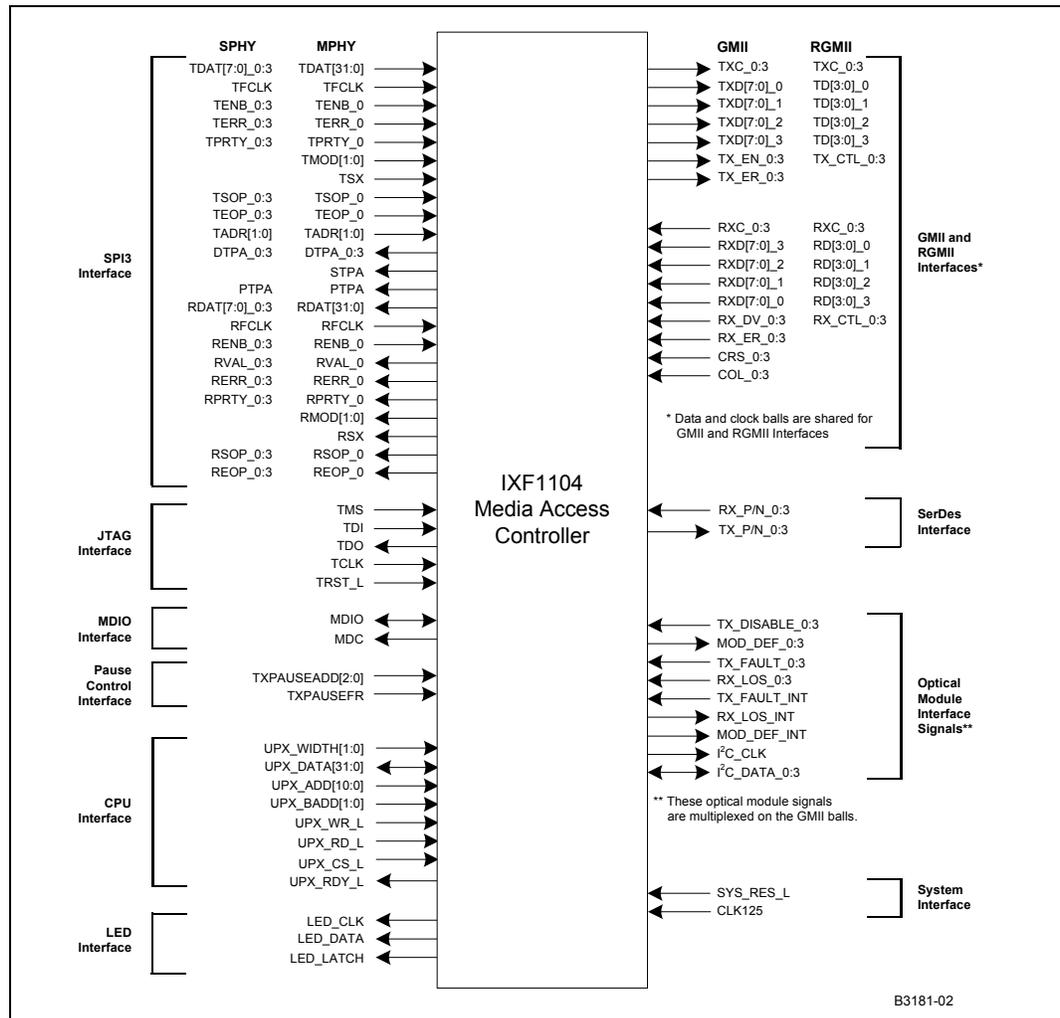
## 4.3    Signal Description Tables

The I/O signals, power supplies, or ground returns associated with each IXF1104 MAC connection ball are described in Table 3 through Table 14.

**Table 3        SPI3 Interface Signal Descriptions (Sheet 1 of 8)**

| Signal Name | | Ball Designator | Type | Standard | Description |
|---|---|---|---|---|---|
| MPHY | SPHY | | | | |
| TDAT31<br>TDAT30<br>TDAT29<br>TDAT28<br>TDAT27<br>TDAT26<br>TDAT25<br>TDAT24 | TDAT7_3<br>TDAT6_3<br>TDAT5_3<br>TDAT4_3<br>TDAT3_3<br>TDAT2_3<br>TDAT1_3<br>TDAT0_3 | F7<br>F5<br>G9<br>G8<br>G7<br>G6<br>G5<br>G4 | Input | 3.3 V LVTTL | **Transmit Data Bus.** Carries payload data to the IXF1104 MAC egress path.<br>*Mode*  *Bits*<br>32-bit Multi-PHY  [31:24]<br>*4 x 8 Single-PHY*  [7:0] for port 3 |
| TDAT23<br>TDAT22<br>TDAT21<br>TDAT20<br>TDAT19<br>TDAT18<br>TDAT17<br>TDAT16 | TDAT7_2<br>TDAT6_2<br>TDAT5_2<br>TDAT4_2<br>TDAT3_2<br>TDAT2_2<br>TDAT1_2<br>TDAT0_2 | C8<br>F9<br>E10<br>E9<br>E8<br>E7<br>E6<br>E5 | Input | 3.3 V LVTTL | **Transmit Data Bus.** Carries payload data to the IXF1104 MAC egress path.<br>*Mode*  *Bits*<br>32-bit Multi-PHY  [23:16]<br>*4 x 8 Single-PHY*  [7:0] for port 2 |
| TDAT15<br>TDAT14<br>TDAT13<br>TDAT12<br>TDAT11<br>TDAT10<br>TDAT9<br>TDAT8 | TDAT7_1<br>TDAT6_1<br>TDAT5_1<br>TDAT4_1<br>TDAT3_1<br>TDAT2_1<br>TDAT1_1<br>TDAT0_1 | H3<br>J3<br>J2<br>J1<br>H1<br>G2<br>G1<br>F1 | Input | 3.3 V LVTTL | **Transmit Data Bus.** Carries payload data to the IXF1104 MAC egress path.<br>*Mode*  *Bits*<br>32-bit Multi-PHY  [15:8]<br>*4 x 8 Single-PHY*  [7:0] for port 1 |
| TDAT7<br>TDAT6<br>TDAT5<br>TDAT4<br>TDAT3<br>TDAT2<br>TDAT1<br>TDAT0 | TDAT7_0<br>TDAT6_0<br>TDAT5_0<br>TDAT4_0<br>TDAT3_0<br>TDAT2_0<br>TDAT1_0<br>TDAT0_0 | C6<br>B5<br>C5<br>C4<br>D1<br>C3<br>C2<br>B3 | Input | 3.3 V LVTTL | **Transmit Data Bus.** Carries payload data to the IXF1104 MAC egress path.<br>*Mode*  *Bits*<br>32-bit Multi-PHY  7:0]<br>*4 x 8 Single-PHY*  [7:0] for port 0 |
| TFCLK | TFCLK | D7 | Input | 3.3 V LVTTL | **Transmit Clock.** TFCLK is the clock associated with all transmit signals. Data and control lines are sampled on the rising edge of TFCLK (frequency operation range 90 - 133 MHz). |
| TPRTY_0 | TPRTY_0<br>TPRTY_1<br>TPRTY_2<br>TPRTY_3 | D5<br>G3<br>B9<br>J6 | Input | 3.3 V LVTTL | **Transmit Parity.** TPRTY indicates odd parity for the TDAT bus. TPRTY is valid only when a channel asserts either TENB or TSX. Odd parity is the default configuration; however, even parity can be selected (see Table 146, *SPI3 Transmit and Global Configuration ($0x700), on page 204*).<br>*32-bit Multi-PHY mode:* TPRTY_0 is the parity bit covering all 32 bits.<br>*4 x 8 Single-PHY mode: T*PRTY_0:3 bits correspond to the respective TDAT[3:0]_*n* channels. |

**Table 3     SPI3 Interface Signal Descriptions (Sheet 2 of 8)**

| Signal Name | | Ball Designator | Type | Standard | Description |
|---|---|---|---|---|---|
| MPHY | SPHY | | | | |
| TENB_0 | TENB_0<br>TENB_1<br>TENB_2<br>TENB_3 | B7<br>E2<br>C9<br>J4 | Input | 3.3 V<br>LVTTL | **Transmit Write Enable.**<br>TENB_0:3 asserted causes an attached PHY to process TDAT[$n$], TMOD, TSOP, TEOP and TERR signals.<br>*32-bit Multi-PHY mode:* TENB_0 is the enable bit for all 32 bits.<br>*4 x 8 Single-PHY mode:* TENB_0:3 bits correspond to the respective TDAT[3:0]_$n$ channels and their associated control and status signals. |
| TERR_0 | TERR_0<br>TERR_1<br>TERR_2<br>TERR_3 | A8<br>K1<br>E11<br>J8 | Input | 3.3 V<br>LVTTL | **Transmit Error.**<br>TERR indicates that there is an error in the current packet. TERR is valid when simultaneously asserted with TEOP and TENB.<br>*32-bit Multi-PHY mode:* TERR_0 is the bit asserted for all 32 bits.<br>*4 x 8 Single-PHY mode:* Each bit of TERR_0:3 corresponds to the respective TDAT[3:0]_$n$ channel. |
| TSOP_0 | TSOP_0<br>TSOP_1<br>TSOP_2<br>TSOP_3 | C7<br>E3<br>C10<br>J5 | Input | 3.3 V<br>LVTTL | **Transmit Start-of-Packet.**<br>TSOP indicates the start of a packet and is valid when asserted simultaneously with TENB.<br>*32-bit Multi-PHY mode:* TSOP_0 is the bit asserted for all 32 bits.<br>*4 x 8 Single-PHY mode:* Each bit of TSOP_0:3 corresponds to the respective TDAT[3:0]_$n$ channel. |
| TEOP_0 | TEOP_0<br>TEOP_1<br>TEOP_2<br>TEOP_3 | A7<br>F3<br>E4<br>H5 | Input | 3.3 V<br>LVTTL | **Transmit End-of-Packet.**<br>TEOP indicates the end of a packet and is valid when asserted simultaneously with TENB.<br>*32-bit Multi-PHY mode:* TEOP_0 is the bit asserted for all 32 bits.<br>*4 x 8 Single-PHY mode:* Each bit of TEOP_0:3 corresponds to the respective TDAT[3:0]_$n$ channel. |

**Table 3    SPI3 Interface Signal Descriptions (Sheet 3 of 8)**

| Signal Name | | Ball Designator | Type | Standard | Description |
|---|---|---|---|---|---|
| MPHY | SPHY | | | | |
| TMOD1<br>TMOD0 | NA | D9<br>A6 | Input | 3.3 V<br>LVTTL | **TMOD[1:0] Transmit Word Modulo.**<br><br>*32-bit Multi-PHY mode:* TMOD[1:0] indicates the valid data bytes of TDAT[31:0]. During transmission, TMOD[1:0] should always be "00" until the last double word is transferred on TDAT[31:0]. TMOD[1:0] specifies the valid bytes of TDAT when TEOP is asserted:<br><br>*TMOD[1:0] – Valid Bytes of TDAT*<br>00 = 4 bytes [31:0]<br>01 = 3 bytes [31:8]<br>10 = 2 bytes [31:16]<br>11 = 1 byte [31:24]<br><br>TENB must be asserted simultaneously for TMOD[1:0] to be valid.<br><br>*4 x 8 Single-PHY mode:* MOD[1:0] is not required. |
| TSX | NA | E1 | Input | 3.3 V<br>LVTTL | **Transmit Start of Transfer.**<br><br>*32-bit Multi-PHY mode:* TSX asserted with TENB = 1 indicates that the PHY address is present on TDAT[7:0]. The valid values on TDAT[7:0] are 3, 2, 1, and 0. When TENB = 0, TSX is not used by the PHY device.<br>***Note:*** Only TDAT[1:0] are relevant; all other bits are "Don't Care".<br><br>*4 x 8 Single-PHY mode:* TSX is not used. |
| TADR1<br>TADR0 | TADR1<br>TADR0 | A12<br>A11 | Input | 3.3 V<br>LVTTL | **TADR[1:0] Transmit PHY Address.**<br>The value on TADR[1:0] selects one of the PHY ports that drives the PTPA signal after the rising edge of TFCLK. |
| DTPA_0<br>DTPA_1<br>DTPA_2<br>DTPA_3 | DTPA_0<br>DTPA_1<br>DTPA_2<br>DTPA_3 | D3<br>L1<br>A9<br>J7 | Output | 3.3 V<br>LVTTL | **DTPA_0:3 Direct Transmit Packet Available.**<br>A direct status indication for transmit FIFOs of ports 0:3.<br><br>When High, DTPA indicates that the amount of data in the TX FIFO is below the TX FIFO High watermark. When the High watermark is crossed, DTPA transitions Low to indicate that the TX FIFO is almost full. It stays Low until the amount of data in the TX FIFO goes back below the TX FIFO Low watermark. At this point, DTPA transitions High to indicate that the programmed number of bytes are now available for data transfers.<br><br>***Note:*** For more information, see Table 132, *TX FIFO High Watermark Ports 0 - 3 ($0x600 – 0x603),* on page 195 and Table 133, *TX FIFO Low Watermark Register Ports 0 - 3 ($0x60A – 0x60D),* on page 196.<br><br>DTPA is updated on the rising edge of TFCLK. |

**Table 3        SPI3 Interface Signal Descriptions (Sheet 4 of 8)**

| Signal Name | | Ball Designator | Type | Standard | Description |
|---|---|---|---|---|---|
| MPHY | SPHY | | | | |
| STPA | NA | C11 | Output | 3.3 V LVTTL | **Selected-PHY Transmit Packet Available.**<br><br>STPA is only meaningful in a 32-bit multi-PHY mode.<br><br>STPA is a direct status indication for transmit FIFOs of ports 0:3.<br><br>When High, STPA indicates that the amount of data in the TX FIFO, specified by the latest in-band address, is below the TX FIFO High watermark. When the High watermark is crossed, STPA transitions Low to indicate the TX FIFO is almost full. It stays Low until the amount of data in the TX FIFO goes back below the TX FIFO Low watermark. At this point, STPA transitions High to indicate that the programmed number of bytes are now available for data transfers.<br><br>*Note:*  For more information, see Table 132, *TX FIFO High Watermark Ports 0 - 3 ($0x600 – 0x603), on page 195* and Table 133, *TX FIFO Low Watermark Register Ports 0 - 3 ($0x60A – 0x60D), on page 196*.<br><br>STPA provides the status indication for the selected port to avoid FIFO overflows while polling is performed. The port reported by STPA is updated on the following rising edge of TFCLK after TSX is sampled as asserted. STPA is updated on the rising edge of TFCLK. |
| PTPA | PTPA | B11 | Output | 3.3 V LVTTL | **Polled-PHY Transmit Packet Available.**<br><br>PTPA allows the polling of the port selected by the TADR address bus.<br><br>When High, PTPA indicates that the amount of data in the TX FIFO is below the TX FIFO High watermark. When the High watermark is crossed, PTPA transitions Low to indicate that the TX FIFO is almost full. It stays Low until the amount data in the TX FIFO goes back below the TX FIFO Low watermark. At this point, PTPA transitions High to indicate that the programmed number of bytes are now available for data transfers.<br><br>*Note:*  For more information, see Table 132, *TX FIFO High Watermark Ports 0 - 3 ($0x600 – 0x603), on page 195* and Table 133, *TX FIFO Low Watermark Register Ports 0 - 3 ($0x60A – 0x60D), on page 196*.<br><br>The port reported by PTPA is updated on the following rising edge of TFCLK after the port address on TADR is sampled by the PHY device.<br><br>PTPA is updated on the rising edge of TFCLK. |

**Table 3        SPI3 Interface Signal Descriptions (Sheet 5 of 8)**

| Signal Name | | Ball Designator | Type | Standard | Description |
|---|---|---|---|---|---|
| **MPHY** | **SPHY** | | | | |
| RDAT31<br>RDAT30<br>RDAT29<br>RDAT28<br>RDAT27<br>RDAT26<br>RDAT25<br>RDAT24 | RDAT7_3<br>RDAT6_3<br>RDAT5_3<br>RDAT4_3<br>RDAT3_3<br>RDAT2_3<br>RDAT1_3<br>RDAT0_3 | F24<br>G24<br>G23<br>G22<br>G21<br>G20<br>G19<br>G18 | Output | 3.3 V<br>LVTTL | **Receive Data Bus.**<br>RDAT carries payload data and in-band addresses from the IXF1104 MAC.<br>*Mode*            *Bits*<br>32-bit Multi-PHY      [31:24]<br>*4 x 8 Single-PHY*      [7:0] for port 3 |
| RDAT23<br>RDAT22<br>RDAT21<br>RDAT20<br>RDAT19<br>RDAT18<br>RDAT17<br>RDAT16 | RDAT7_2<br>RDAT6_2<br>RDAT5_2<br>RDAT4_2<br>RDAT3_2<br>RDAT2_2<br>RDAT1_2<br>RDAT0_2 | E21<br>E22<br>D22<br>C22<br>C21<br>C20<br>B22<br>B20 | Output | 3.3 V<br>LVTTL | **Receive Data Bus.**<br>RDAT carries payload data and in-band addresses from the IXF1104 MAC.<br>*Mode*            *Bits*<br>32-bit Multi-PHY      [23:16]<br>*4 x 8 Single-PHY*      [7:0] for port 2 |
| RDAT15<br>RDAT14<br>RDAT13<br>RDAT12<br>RDAT11<br>RDAT10<br>RDAT9<br>RDAT8 | RDAT7_1<br>RDAT6_1<br>RDAT5_1<br>RDAT4_1<br>RDAT3_1<br>RDAT2_1<br>RDAT1_1<br>RDAT0_1 | F18<br>E18<br>E17<br>F16<br>E16<br>D16<br>C17<br>A17 | Output | 3.3 V<br>LVTTL | **Receive Data Bus.**<br>RDAT carries payload data and in-band addresses from the IXF1104 MAC.<br>*Mode*            *Bits*<br>32-bit Multi-PHY      [15:8]<br>*4 x 8 Single-PHY*      [7:0] for port 1 |
| RDAT7<br>RDAT6<br>RDAT5<br>RDAT4<br>RDAT3<br>RDAT2<br>RDAT1<br>RDAT0 | RDAT7_0<br>RDAT6_0<br>RDAT5_0<br>RDAT4_0<br>RDAT3_0<br>RDAT2_0<br>RDAT1_0<br>RDAT0_0 | F14<br>E14<br>D14<br>C13<br>C14<br>B14<br>A14<br>A15 | Output | 3.3 V<br>LVTTL | **Receive Data Bus.**<br>RDAT carries payload data and in-band addresses from the IXF1104 MAC.<br>*Mode*            *Bits*<br>32-bit Multi-PHY      [7:0]<br>*4 x 8 Single-PHY*      [7:0] for port 0 |
| RFCLK | RFCLK | A19 | Input | 3.3 V<br>LVTTL | **Receive Clock.**<br>RFCLK is the clock associated with all receive signals. Data and controls are driven on the rising edge of RFCLK (frequency operation range 90 - 133 MHz). |
| RPRTY_0 | RPRTY_0<br>RPRTY_1<br>RPRTY_2<br>RPRTY_3 | E15<br>G16<br>E20<br>F20 | Output | 3.3 V<br>LVTTL | **Receive Parity.**<br>RPRTY indicates odd parity for the RDAT bus. RPRTY is valid only when a channel asserts RENB or RSX. Odd parity is the default configuration; however, even parity can be selected (see Table 147 on page 207).<br>*32-bit Multi-PHY mode:* RPRTY_0 is the parity bit for all 32 bits.<br>*4 x 8 Single-PHY mode:* Each bit of RPRTY_0:3 corresponds to the respective RDAT[3:0]_*n* channel. |

**Table 3    SPI3 Interface Signal Descriptions (Sheet 6 of 8)**

| Signal Name | | Ball Designator | Type | Standard | Description |
|---|---|---|---|---|---|
| MPHY | SPHY | | | | |
| RENB_0 | RENB_0 RENB_1 RENB_2 RENB_3 | A13 A18 C19 E24 | Input | 3.3 V LVTTL | **Receive Read Enable.** The RENB signal controls the flow of data from the receive FIFOs. During data transfer, RVAL must be monitored as it indicates if the RDAT[31:0], RPRTY, RMOD[1:0], RSOP, REOP, RERR, and RSX are valid. The system may de-assert RENB at any time if it is unable to accept data from the IXF1104 MAC. When RENB is sampled Low, a read is performed from the receive FIFO and the RDAT[31:0], RPRTY, RMOD[1:0], RSOP, REOP, RERR, RSX and RVAL signals are updated on the following rising edge of RFCLK. When RENB is sampled High by the PHY device, a read is not performed, and the RDAT[31:0], RPRTY, RMOD[1:0], RSOP, REOP, RERR, RSX, and RVAL signals remain unchanged on the following rising edge of RFCLK. *32-bit Multi-PHY Mode:* RENB_0 covers all receive bits. *4 x 8 Single-PHY Mode:* The RENB_0:3 bits correspond to the per-port data and control signals. |
| RERR_0 | RERR_0 RERR_1 RERR_2 RERR_3 | A16 G17 D20 H20 | Output | 3.3 V LVTTL | **Receive Error.** RERR indicates that the current packet is in error. RERR is only asserted when REOP is asserted. Conditions that can cause RERR to be set include FIFO overflow, CRC error, code error, and runt or giant packets. *Note:* RERR can only be set for these conditions if bit 0 in the Section 147, *SPI3 Receive Configuration ($0x701)* is set to 1. RERR is considered valid only when RVAL is asserted. *32-bit Multi-PHY mode:* RERR_0 covers all 32 bits. *4 x 8 Single-PHY mode:* The RERR_0:3 bits correspond to the RDAT[7:0]_n channels. (*n* = 0, 1, 2, or 3) |

**Table 3          SPI3 Interface Signal Descriptions (Sheet 7 of 8)**

| Signal Name | | Ball Designator | Type | Standard | Description |
|---|---|---|---|---|---|
| **MPHY** | **SPHY** | | | | |
| RVAL_0 | RVAL_0<br>RVAL_1<br>RVAL_2<br>RVAL_3 | C15<br>B18<br>E19<br>F22 | Output | 3.3 V<br>LVTTL | **Receive Data Valid.**<br>RVAL indicates the validity of the receive data signals. RVAL is Low between transfers and assertion of RSX. It is also Low when the IXF1104 MAC pauses a transfer due to an empty receive FIFO. When a transfer is paused by holding RENB High, RVAL holds its value unchanged, although no new data is present on RDAT[31:0] until the transfer resumes. When RVAL is High, the RDAT[31:0], RMOD[1:0], RSOP, REOP, and RERR signals are valid. When RVAL is Low, the RDAT[31:0], RMOD[1:0], RSOP, REOP, and RERR signals are invalid and must be disregarded.<br>The RSX signal is valid only when RVAL is Low.<br>*32-bit Multi-PHY mode:* RVAL_0 covers all receive bits.<br>*4 x 8 Single-PHY mode:* The RVAL_0:3 bits correspond to the per-port data and control signals. |
| RSOP_0 | RSOP_0<br>RSOP_1<br>RSOP_2<br>RSOP_3 | B16<br>C18<br>E23<br>J18 | Output | 3.3 V<br>LVTTL | **Receive Start of Packet.**<br>RSOP indicates the start of a packet when asserted with RVAL.<br>*32-bit Multi-PHY mode:* RSOP_0 covers all 32 bits.<br>*4 x 8 Single-PHY mode:* The RSOP_0:3 bits correspond to the RDAT[7:0]_*n* channels. |

**Table 3** **SPI3 Interface Signal Descriptions (Sheet 8 of 8)**

| Signal Name | | Ball Designator | Type | Standard | Description |
|---|---|---|---|---|---|
| MPHY | SPHY | | | | |
| REOP_0 | REOP_0 REOP_1 REOP_2 REOP_3 | C16 D18 C23 J19 | Output | 3.3 V LVTTL | **Receive End of Packet.** REOP indicates the end of a packet when asserted with RVAL. *32-bit Multi-PHY mode:* REOP_0 covers all 32 bits. *4 x 8 Single-PHY mode:* The REOP_0:3 bits correspond to the RDAT[7:0]_*n* channels. |
| RMOD1 RMOD0 | NA | G13 G14 | Output | 3.3 V LVTTL | **Receive Word Modulo:** *32-bit Multi-PHY mode:* RMOD[1:0] indicates the valid bytes of data in RDAT[31:0]. During transmission, RMOD is always "00", except when the last double-word is transferred on RDAT[31:0]. RMOD[1:0] specifies the valid packet data bytes on RDAT[31:0] when REOP is asserted. *RMOD[1:0]        Valid Bytes of RDAT* 00 = 4 bytes [31:0] 01 = 3 bytes [31:8] 10 = 2 bytes [31:16] 11 = 1 byte [31:24] *4 x 8 Single-PHY mode:* RMOD[1:0] is not required. RMOD is considered valid only when RVAL is simultaneously asserted. RENB must be asserted for RMOD[1:0] to be valid. |
| RSX | NA | E13 | Output | 3.3 V LVTTL | **Receive Start of Transfer.** *32-bit Multi-PHY mode:* RSX indicates when the in-band port address is present on the RDAT bus. When RSX is High and RVAL = 0, the value of RDAT[7:0] is the address of the receive FIFO to be selected. Subsequent data transfers on RDAT are from the FIFO specified by this in-band address. Values of 0, 1, 2, and 3 select the corresponding port. RSX is ignored when RVAL is de-asserted. *4 x 8 Single-PHY mode:* RSX is ignored. |

**Table 4        SerDes Interface Signal Descriptions**

| Signal Name | Ball Designator | Type | Standard | Description |
|---|---|---|---|---|
| TX_P_0<br>TX_P_1<br>TX_P_2<br>TX_P_3 | Y13<br>AD13<br>W16<br>AC18 | Output | SerDes | Transmit Differential Output, Positive. |
| TX_N_0<br>TX_N_1<br>TX_N_2<br>TX_N_3 | Y14<br>AD14<br>Y16<br>AD18 | Output | SerDes | Transmit Differential Output, Negative. |
| RX_P_0<br>RX_P_1<br>RX_P_2<br>RX_P_3 | P22<br>V22<br>T24<br>U24 | Input | SerDes | Receive Differential Input, Positive.[1] |
| RX_N_0<br>RX_N_1<br>RX_N_2<br>RX_N_3 | R22<br>U22<br>R24<br>V24 | Input | SerDes | Receive Differential Input, Negative.[1] |
| 1.   Internally terminated differentially with 100 Ω. | | | | |

**Table 5    GMII Interface Signal Descriptions (Sheet 1 of 2)**

| Signal Name | Ball Designator | Type | Standard | Description |
|---|---|---|---|---|
| TXD7_0<br>TXD6_0<br>TXD5_0<br>TXD4_0<br>TXD3_0<br>TXD2_0<br>TXD1_0<br>TXD0_0<br><br>TXD7_1<br>TXD6_1<br>TXD5_1<br>TXD4_1<br>TXD3_1<br>TXD2_1<br>TXD1_1<br>TXD0_1<br><br>TXD7_2<br>TXD6_2<br>TXD5_2<br>TXD4_2<br>TXD3_2<br>TXD2_2<br>TXD1_2<br>TXD0_2<br><br>TXD7_3<br>TXD6_3<br>TXD5_3<br>TXD4_3<br>TXD3_3<br>TXD2_3<br>TXD1_3<br>TXD0_3 | Y4<br>AB4<br>AC3<br>AB3<br>AA3<br>Y3<br>Y2<br>Y1<br><br>AC9<br>AD8<br>AB8<br>AA7<br>AD9<br>AB9<br>AB7<br>AC7<br><br>AA18<br>AA20<br>AB19<br>AD16<br>AB23<br>AB22<br>AB21<br>AB20<br><br>W14<br>AA16<br>Y15<br>AA14<br>V17<br>V16<br>V15<br>V14 | Output | 2.5 V<br>CMOS | **Transmit Data.**<br>Each bus carries eight data bits [7:0] of the transmitted data stream to the PHY device.<br><br>**RGMII Mode:** When a port is configured in copper mode and the RGMII interface is selected, only bits TXD[3:0]_$n$ are used. The data is transmitted on both edges of TXC_0:3.<br><br>**Fiber Mode:** The following signals have multiplexed functions when a port is configured in fiber mode:<br>TXD4_$n$: TX_DISABLE_0:3 |
| TX_EN_0<br>TX_EN_1<br>TX_EN_2<br>TX_EN_3 | AB2<br>Y8<br>AC22<br>V12 | Output | 2.5 V<br>CMOS | **Transmit Enable.**<br>TX_EN indicates that valid data is being driven on the corresponding Transmit Data: TXD_0, TXD_1, TXD_2, and TXD_3. |
| TX_ER_0<br>TX_ER_1<br>TX_ER_2<br>TX_ER_3 | W1<br>AD6<br>AD17<br>AB13 | Output | 2.5 V<br>CMOS | **Transmit Error:**<br>TX_ER indicates a transmit error in the corresponding Transmit Data: TXD_0, TXD_1, TXD_2, and TXD_3. |
| TXC_0<br>TXC_1<br>TXC_2<br>TXC_3 | AA1<br>AD7<br>AC20<br>AB14 | Output | 2.5 V<br>CMOS | **Source Synchronous Transmit Clock.**<br>This clock is supplied synchronous to the transmit data bus in either RGMII or GMII mode.<br><br>*Note:*    Shares the same balls as RXC on the RGMII interface. |
| *Note:*    Refer to the RGMII interface for shared data and clock signals. | | | | |

**Table 5** **GMII Interface Signal Descriptions (Sheet 2 of 2)**

| Signal Name | Ball Designator | Type | Standard | Description |
|---|---|---|---|---|
| RXD7_0<br>RXD6_0<br>RXD5_0<br>RXD4_0<br>RXD3_0<br>RXD2_0<br>RXD1_0<br>RXD0_0<br><br>RXD7_1<br>RXD6_1<br>RXD5_1<br>RXD4_1<br>RXD3_1<br>RXD2_1<br>RXD1_1<br>RXD0_1<br><br>RXD7_2<br>RXD6_2<br>RXD5_2<br>RXD4_2<br>RXD3_2<br>RXD2_2<br>RXD1_2<br>RXD0_2<br><br>RXD7_3<br>RXD6_3<br>RXD5_3<br>RXD4_3<br>RXD3_3<br>RXD2_3<br>RXD1_3<br>RXD0_3 | AC5<br>AB5<br>Y5<br>Y6<br>Y7<br>W7<br>V7<br>V8<br><br>Y10<br>AA11<br>AC11<br>AD10<br>W9<br>W11<br>Y11<br>Y9<br><br>W20<br>V19<br>V20<br>W22<br>Y23<br>Y22<br>Y21<br>Y20<br><br>T19<br>T18<br>T17<br>T16<br>W18<br>Y19<br>Y18<br>Y17 | Input | 2.5 V<br>CMOS | **Receive Data:**<br>Each bus carries eight data bits [7:0] of the received data stream.<br>**RGMII Mode:** When a port ID is configured in copper mode and the RGMII interface is selected, only bits RXD[3:0]_$n$ are used to receive data.<br>**Fiber Mode**: The following signals have multiplexed functions when a port is configured in fiber mode:<br>RXD4_$n$: MOD_DEF_0:3<br>RXD5_$n$: TX_FAULT_0:3<br>RXD6_$n$: RX_LOS_0:3 |
| RX_DV_0<br>RX_DV_1<br>RX_DV_2<br>RX_DV_3 | V5<br>AB11<br>Y24<br>V18 | Input | 2.5 V<br>CMOS | **Receive Data Valid.**<br>RX_DV indicates that valid data is being driven on Receive Data: RXD[7:0]_$n$. |
| RX_ER_0<br>RX_ER_1<br>RX_ER_2<br>RX_ER_3 | W5<br>Y12<br>AA22<br>U20 | Input | 2.5 V<br>CMOS | **Receive Error.**<br>RX_ER indicates an error in Receive Data: RXD[7:0]_$n$. |
| CRS_0<br>CRS_1<br>CRS_2<br>CRS_3 | AA5<br>AA9<br>AB15<br>AC16 | Input | 2.5 V<br>CMOS | **Carrier Sense.**<br>CRS indicates the PHY device has detected a carrier. |
| RXC_0<br>RXC_1<br>RXC_2<br>RXC_3 | V4<br>AD11<br>AA24<br>V23 | Input | 2.5 V<br>CMOS | **Receiver Reference Clock.**<br>RXC operates at:<br>125 MHz for 1 Gigabit<br><br>**Note:** Shares the same balls as RXC on the RGMII interface. |
| **Note:** Refer to the RGMII interface for shared data and clock signals. | | | | |

**Table 6**     **RGMII Interface Signal Descriptions (Sheet 1 of 2)**

| Signal Name | Ball Designator | Type | Standard | Description |
|---|---|---|---|---|
| TXC_0<br>TXC_1<br>TXC_2<br>TXC_3 | AA1<br>AD7<br>AC20<br>AB14 | Output | 2.5 V CMOS | **Source Synchronous Transmit Clock.**<br>This clock is supplied synchronous to the transmit data bus in either RGMII or GMII mode. |
| TD3_0<br>TD2_0<br>TD1_0<br>TD0_0<br><br>TD3_1<br>TD2_1<br>TD1_1<br>TD0_1<br><br>TD3_2<br>TD2_2<br>TD1_2<br>TD0_2<br><br>TD3_3<br>TD2_3<br>TD1_3<br>TD0_3 | AA3<br>Y3<br>Y2<br>Y1<br><br>AD9<br>AB9<br>AB7<br>AC7<br><br>AB23<br>AB22<br>AB21<br>AB20<br><br>V17<br>V16<br>V15<br>V14 | Output | 2.5 V CMOS | **Transmit Data.**<br>Bits [3:0] are clocked on the rising edge of TXC. Bits [7:4] are clocked on the falling edge of TXC.<br><br>*Note:* Shares data signals TXD[3:0]_*n* with the GMII interface. |
| TX_CTL_0<br>TX_CTL_1<br>TX_CTL_2<br>TX_CTL_3 | AB2<br>Y8<br>AC22<br>V12 | Output | 2.5 V CMOS | **Transmit Control.**<br>TX_CTL is TX_EN on the rising edge of TXC and a logical derivative of TX_EN and TX_ER on the falling edge of TXC.<br><br>*Note:* TX_CTL multiplexes with TX_EN_*n* on the GMII interface. |

**Table 6          RGMII Interface Signal Descriptions (Sheet 2 of 2)**

| Signal Name | Ball Designator | Type | Standard | Description |
|---|---|---|---|---|
| RXC_0<br>RXC_1<br>RXC_2<br>RXC_3 | V4<br>AD11<br>AA24<br>V23 | Input | 2.5 V CMOS | **Receiver Reference Clock.**<br>Operates at:<br>125 MHz for 1 Gigabit<br>25 MHz for 100 Mbps<br>2.5 MHz for 10 Mbps<br><br>*Note:*   Shares the same balls as RXC on the GMII interface. |
| RD3_0<br>RD2_0<br>RD1_0<br>RD0_0<br><br>RD3_1<br>RD2_1<br>RD1_1<br>RD0_1<br><br>RD3_2<br>RD2_2<br>RD1_2<br>RD0_2<br><br>RD3_3<br>RD2_3<br>RD1_3<br>RD0_3 | Y7<br>W7<br>V7<br>V8<br><br>W9<br>W11<br>Y11<br>Y9<br><br>Y23<br>Y22<br>Y21<br>Y20<br><br>W18<br>Y19<br>Y18<br>Y17 | Input | 2.5 V CMOS | **Receive Data.**<br>Bits [3:0] are clocked on the rising edge of RXC. Bits [7:4] are clocked on the falling edge of RXC.<br><br>*Note:*   Shares balls with RXD[3:0]_0 on the GMII interface. |
| RX_CTL_0<br>RX_CTL_1<br>RX_CTL_2<br>RX_CTL_3 | V5<br>AB11<br>Y24<br>V18 | Input | 2.5 V CMOS | **Receive Control.**<br>RX_CTL is RX_DV on the rising edge of RXC and a logical derivative of RX_DV and RERR on the falling edge of RXC.<br><br>*Note:*   RX_CTL shares the same balls as RX_DV on the GMII interface. |

**Table 7       CPU Interface Signal Descriptions (Sheet 1 of 2)**

| Signal Name | Ball Designator | Type | Standard | Description |
|---|---|---|---|---|
| UPX_ADD10<br>UPX_ADD9<br>UPX_ADD8<br>UPX_ADD7<br>UPX_ADD6<br>UPX_ADD5<br>UPX_ADD4<br>UPX_ADD3<br>UPX_ADD2<br>UPX_ADD1<br>UPX_ADD0 | T3<br>U3<br>V3<br>V2<br>V1<br>U1<br>T1<br>R1<br>P1<br>N1<br>P3 | Input | 3.3 V LVTTL | UPX_ADD is the address bus from the microprocessor. |
| UPX_BADD1<br>UPX_BADD0 | W3<br>T2 | Input | 3.3 V LVTTL | *16-bit mode:* The data word select uses UPX_BADD1.<br><br>*8-bit mode:* UPX_BADD[1:0] selects the individual bytes. |
| UPX_DATA31<br>UPX_DATA30<br>UPX_DATA29<br>UPX_DATA28<br>UPX_DATA27<br>UPX_DATA26<br>UPX_DATA25<br>UPX_DATA24<br>UPX_DATA23<br>UPX_DATA22<br>UPX_DATA21<br>UPX_DATA20<br>UPX_DATA19<br>UPX_DATA18<br>UPX_DATA17<br>UPX_DATA16<br>UPX_DATA15<br>UPX_DATA14<br>UPX_DATA13<br>UPX_DATA12<br>UPX_DATA11<br>UPX_DATA10<br>UPX_DATA9<br>UPX_DATA8<br>UPX_DATA7<br>UPX_DATA6<br>UPX_DATA5<br>UPX_DATA4<br>UPX_DATA3<br>UPX_DATA2<br>UPX_DATA1<br>UPX_DATA0 | L17<br>J17<br>H16<br>J16<br>M15<br>N15<br>K15<br>H14<br>K13<br>G12<br>K12<br>G11<br>H11<br>G10<br>K10<br>M10<br>N10<br>J9<br>H9<br>L8<br>N7<br>L7<br>L6<br>P5<br>K5<br>M5<br>N5<br>L4<br>M3<br>L3<br>K3<br>L2 | Input/<br>Output | 3.3 V LVTTL | **Data bus.**<br>**32-bit mode:** Uses [31:0]<br>*16-bit mode:* Uses [15:0]<br>*8-bit mode:* Uses [7:0] |
| UPX_CS_L | R3 | Input | 3.3 V LVTTL | **Chip Select.** Active Low. |
| UPX_WR_L | T4 | Input | 3.3 V LVTTL | **Write Strobe.** Active Low. |
| UPX_RD_L | V6 | Input | 3.3 V LVTTL | **Read Strobe.** Active Low. |

**Table 7        CPU Interface Signal Descriptions (Sheet 2 of 2)**

| Signal Name | Ball Designator | Type | Standard | Description |
|---|---|---|---|---|
| UPX_RDY_L | M1 | Open Drain Output* | 3.3 V LVTTL | **Cycle complete indicator.**<br>Active Low.<br><br>*Note:* An external pull-up resistor is required for proper operation.<br>*Note:* *Dual-mode I/O<br>**Normal operation:** Open drain output<br>**Boundary Scan Mode:** Standard CMOS output |
| UPX_WIDTH1<br>UPX_WIDTH0 | T5<br>U16 | Input | 3.3 V LVTTL | **Data bus width select.**<br>UPX_WIDTH[1:0] specifies the CPU bus width.<br><br>**UPX_WIDTH[1:0]** **Mode**<br>00 8-bit<br>01 16-bit<br>1x 32-bit |

**Table 8        Transmit Pause Control Interface Signal Descriptions**

| Signal Name | Ball Designator | Type | Standard | Description |
|---|---|---|---|---|
| TXPAUSEADD2<br>TXPAUSEADD1<br>TXPAUSEADD0 | P21<br>P20<br>N20 | Input | 2.5 V CMOS | TXPAUSEADD[2:0] is the port selection address for pause frame insertion. |
| TXPAUSEFR | T20 | Input | 2.5 V CMOS | TX Pause Interface Strobe. |

**Table 9**      **Optical Module Interface Signal Descriptions (Sheet 1 of 2)**

| Signal Name | Ball Designator | Type | Standard | Description |
|---|---|---|---|---|
| TX_DISABLE_0<br>TX_DISABLE_1<br>TX_DISABLE_2<br>TX_DISABLE_3 | AB3<br>AA7<br>AD16<br>AA14 | Open Drain Output* | 2.5 V CMOS | **Transmit Disable:**<br>TX_DISABLE_0:3 outputs disable the Optical Module Interface transmitter. An external pull-up resistor usually resident in an optical module is required for proper operation.<br><br>*Note:* These signals are multiplexed with the TXD[4]_$n$ bits of the GMII Interface<br>*Note:* *Dual-mode I/O<br>Normal operation: Open drain output<br>Boundary Scan Mode: Standard CMOS output |
| MOD_DEF_0<br>MOD_DEF_1<br>MOD_DEF_2<br>MOD_DEF_3 | Y6<br>AD10<br>W22<br>T16 | Input | 2.5 V CMOS | MOD_DEF_0:3 inputs determine when an Optical Module Interface is present.<br><br>*Note:* These signals are multiplexed with the RXD[4]_$n$ bits of the GMII interface. |
| RX_LOS_0<br>RX_LOS_1<br>RX_LOS_2<br>RX_LOS_3 | AB5<br>AA11<br>V19<br>T18 | Input | 2.5 V CMOS | RX_LOS_0:3 inputs determine when the Optical Module Interface receiver loses synchronization.<br><br>*Note:* These signals are multiplexed with the RXD[6]_$n$ bits of the GMII interface. |
| TX_FAULT_0<br>TX_FAULT_1<br>TX_FAULT_2<br>TX_FAULT_3 | Y5<br>AC11<br>V20<br>T17 | Input | 2.5 V CMOS | TX_FAULT_0:3 inputs determine an Optical Module Interface transmitter fault.<br><br>*Note:* These signals are multiplexed with the RXD[5]_$n$ bits of the GMII Interface. |
| RX_LOS_INT | P19 | Open Drain Output* | 2.5 V CMOS | **Receiver Loss of Signal Interrupt.**<br>RX_LOS_INT is an open drain interrupt output to signal an RX_LOS condition.<br><br>*Note:* An external pull-up resistor is required for proper operation.<br>*Note:* *Dual-mode I/O<br>Normal operation: Open drain output<br>Boundary Scan Mode: Standard CMOS output |
| TX_FAULT_INT | P23 | Open Drain Output* | 2.5 V CMOS | **Transmitter Fault Interrupt.**<br>TX_FAULT_INT is an open drain interrupt output that signals a TX_FAULT condition.<br><br>*Note:* An external pull-up resistor is required for proper operation.<br>*Note:* *Dual-mode I/O<br>Normal operation: Open drain output<br>Boundary Scan Mode: Standard CMOS output |

**Table 9**      **Optical Module Interface Signal Descriptions (Sheet 2 of 2)**

| Signal Name | Ball Designator | Type | Standard | Description |
|---|---|---|---|---|
| MOD_DEF_INT | N22 | Open Drain Output* | 2.5 V CMOS | **Module Definition Interrupt**.<br>MOD_DEF_INT is an open drain interrupt output that signals a MOD_DEF condition.<br><br>*Note:* An external pull-up resistor is required for proper operation.<br>*Note:* *Dual-mode I/O<br>Normal operation: Open drain output<br>Boundary Scan Mode: Standard CMOS output |
| $I^2C$_CLK | L23 | Output | 2.5 V CMOS | $I^2C$_CLK is the clock used for the $I^2C$ bus interface. |
| $I^2C$ DATA_0<br>$I^2C$ DATA_1<br>$I^2C$ DATA_2<br>$I^2C$ DATA_3 | L24<br>M24<br>N24<br>P24 | Input/ Open Drain Output* | 2.5 V CMOS | **$I^2C$ Data Bus.**<br>$I^2C$ DATA_0:3 are the data I/Os for the $I^2C$ bus interface.<br><br>*Note:* An external pull-up resistor is required for proper operation.<br>*Note:* *Dual-mode I/O<br>Normal operation: Input/ open drain output<br>Boundary Scan Mode: Standard CMOS output |

**Table 10**      **MDIO Interface Signal Descriptions**

| Signal Name | Ball Designator | Type | Standard | Description |
|---|---|---|---|---|
| MDIO | V21 | Input/ Output | 2.5 V CMOS | MDIO is the management data input and output.<br><br>*Note:* An external pull-up resistor is required for proper operation. |
| MDC | W24 | Output | 2.5 V CMOS | MDC is the management clock to external devices. |

**Table 11**      **LED Interface Signal Descriptions**

| Signal Name | Ball Designator | Type | Standard | Description |
|---|---|---|---|---|
| LED_CLK | K24 | Output | 2.5 V CMOS | LED_CLK is the clock output for the LED block. |
| LED_DATA | M22 | Output | 2.5 V CMOS | LED_DATA is the data output for the LED block. |
| LED_LATCH | L22 | Output | 2.5 V CMOS | LED_LATCH is the latch enable for the LED block. |

**Table 12      JTAG Interface Signal Descriptions**

| Signal Name | Ball Designator | Type | Standard | Description |
|---|---|---|---|---|
| TCLK | J22 | Input | 3.3 V LVTTL | JTAG Test Clock |
| TMS | H22 | Input | 3.3 V LVTTL | Test Mode Select |
| TDI | J24 | Input | 3.3 V LVTTL | Test Data Input |
| TDO | H24 | Output | 3.3 V LVTTL | Test Data Output |
| TRST_L | J23 | Input | 3.3 V LVTTL | Test Reset; reset input for JTAG test |

**Table 13      System Interface Signal Descriptions**

| Signal Name | Ball Designator | Type | Standard | Description |
|---|---|---|---|---|
| CLK125 | AD19 | Input | 2.5 V CMOS | CLK125 is the input clock to PLL; 125 MHz +/- 50 ppm |
| SYS_RES_L | AD12 | Input | 2.5 V CMOS | SYS_RES_L is the system hard reset (active Low). |

**Table 14      Power Supply Signal Descriptions (Sheet 1 of 2)**

| Signal Name | Ball Designator | | | | Type | Standard | Description |
|---|---|---|---|---|---|---|---|
| GND | A4  A21  B6  B10<br>B15  B19  D4  D8<br>D12  D13  D17  D21<br>F2  F6  F10  F15<br>F19  F23  H4  H8<br>H12  H13  H17  H21<br>J10  J15  K2  K6<br>K9  K11  K14  K16<br>K19  K23  L5  L10<br>L12  L13  L15  L20<br>M4  M8  M11  M14<br>M17  M21  N4  N8<br>N11  N14  N17  N21<br>P10  P12  P13  P15<br>R2  R6  R7  R9<br>R11  R14  R16  R19<br>R23  T10  T15  U4<br>U8  U12  U13  U17<br>U21  W2  W6  W10<br>W15  W19  W23  AA4<br>AA8  AA12  AA13  AA17<br>AA21  AB12  AC6  AC10<br>AC14  AC15  AC19  AD21 | | | | Input | – | Digital ground |
| AVDD1P8_1 | A5  A20 | | | | Input | 1.8 V | Analog 1.8 V supply |
| AVDD1P8_2 | AB16  T23 | | | | Input | 1.8 V | Analog 1.8 V supply |
| AVDD2P5_1 | AD20 | | | | Input | 2.5 V | Analog 2.5 V supply |

**Table 14      Power Supply Signal Descriptions (Sheet 2 of 2)**

| Signal Name | Ball Designator | | | | Type | Standard | Description |
|---|---|---|---|---|---|---|---|
| AVDD2P5_2 | U14 | R18 | | | Input | 2.5 V | Analog 2.5 V supply |
| VDD | A10<br>D11<br>F21<br>J14<br>K17<br>L14<br>P14<br>R17<br>U10<br>AA6 | C12<br>D15<br>H10<br>J20<br>K21<br>L16<br>P16<br>R21<br>U15<br>AA10 | D6<br>D19<br>H15<br>K4<br>L9<br>P9<br>R4<br>T11<br>W4<br>AA15 | D10<br>F4<br>J11<br>K8<br>L11<br>P11<br>R8<br>T14<br>W21<br>AA19 | Input | 1.8 V | Digital 1.8 V supply |
| VDD2 | B4<br>F8<br>J12<br>M12 | B8<br>F12<br>M2 | B12<br>H2<br>M6 | D2<br>H6<br>M9 | Input | 3.3 V | Digital 3.3 V supply |
| VDD3 | B13<br>F13<br>J13<br>M23 | B17<br>F17<br>M13 | B21<br>H19<br>M16 | D23<br>H23<br>M19 | Input | 3.3 V | Digital 3.3 V supply |
| VDD4 | N13<br>T13<br>W17<br>AC21 | N16<br>U19<br>AA23 | N19<br>U23<br>AC13 | N23<br>W13<br>AC17 | Input | 2.5 V | Digital 2.5 V supply |
| VDD5 | N2<br>T12<br>W12<br>AC12 | N6<br>U2<br>AA2 | N9<br>U6<br>AC4 | N12<br>W8<br>AC8 | Input | 2.5 V | Digital 2.5 V supply |

## 4.4      Ball Usage Summary

**Table 15      Ball Usage Summary**

| Type | Quantity |
|---|---|
| Inputs | 158 |
| Outputs | 126 |
| Bi-directional | 37 |
| Total Signals | 321 |
| Power | 75 |
| Ground | 82 |
| No Connects | 74 |
| **Total** | **552** |

## 4.5 Multiplexed Ball Connections

### 4.5.1 GMII/RGMII/SerDes/OMI Multiplexed Ball Connections

Table 16 lists the balls used for the line-side interfaces (GMII, RGMII, SerDes/OMI) and provides a guide to connect these balls. Some of these balls are multiplexed depending on the mode of operation selected for that port.

**Note:**    Do not connect any balls marked as unused (NC).

**Table 16    Line Side Interface Multiplexed Balls (Sheet 1 of 2)**

| Copper Mode | | Fiber Mode | Unused Port | Ball Designator | | | |
|---|---|---|---|---|---|---|---|
| GMII Signal | RGMII Signal | Optical Module/ SerDes Signal | | | | | |
| TXC_0:3 | TXC_0:3 | NC | NC | AA1 | AD7 | AC20 | AB14 |
| TXD[3:0]_0 TXD[3:0]_1 TXD[3:0]_2 TXD[3:0]_3 | TD[3:0]_0 TD[3:0]_1 TD[3:0]_2 TD[3:0]_3 | NC | NC | AA3 AD9 AB23 V17 | Y3 AB9 AB22 V16 | Y2 AB7 AB21 V15 | Y1 AC7 AB20 V14 |
| TXD4_0:3 | NC | TX_DISABLE_0:3[2] | NC | AB3 | AA7 | AD16 | AA14 |
| TXD[7:5]_0 TXD[7:5]_1 TXD[7:5]_2 TXD[7:5]_3 | NC | NC | NC | Y4 AC9 AA18 W14 | AB4 AD8 AA20 AA16 | AC3 AB8 AB19 Y15 | |
| TX_EN_0:3 | TX_CTL_0:3 | NC | NC | AB2 | Y8 | AC22 | V12 |
| TX_ER_0:3 | NC | NC | NC | W1 | AD6 | AD17 | AB13 |
| RXC_0:3 | RXC_0:3 | GND | GND | V4 | AD11 | AA24 | V23 |
| RXD[3:0]_0 RXD[3:0]_1 RXD[3:0]_2 RXD[3:0]_3 | RD[3:0]_0 RD[3:0]_1 RD[3:0]_2 RD[3:0]_3 | GND | GND | Y7 W9 Y23 W18 | W7 W11 Y22 Y19 | V7 Y11 Y21 Y18 | V8 Y9 Y20 Y17 |
| RXD4_0:3 | GND | MOD_DEF_0:3[1] | GND | Y6 | AD10 | W22 | T16 |
| RXD5_0:3 | GND | TX_FAULT_0:3[1] | GND | Y5 | AC11 | V20 | T17 |
| RXD6_0:3 | GND | RX_LOS_0:3[1] | GND | AB5 | AA11 | V19 | T18 |
| RXD7_0:3 | GND | GND | GND | AC5 | Y10 | W20 | T19 |
| RX_DV_0:3 | RX_CTL_0:3 | GND | GND | V5 | AB11 | Y24 | V18 |
| RX_ER_0:3 | GND | GND | GND | W5 | Y12 | AA22 | U20 |
| CRS_0:3 | GND | GND | GND | AA5 | AA9 | AB15 | AC16 |
| COL_0:3 | GND | GND | GND | AB6 | AB10 | AD15 | AB17 |
| GND | GND | RX_P_0:3 | GND | P22 | V22 | T24 | U24 |
| GND | GND | RX_N_0:3 | GND | R22 | U22 | R24 | V24 |
| NC | NC | TX_P_0:3 | NC | Y13 | AD13 | W16 | AC18 |
| NC | NC | TX_N_0:3 | NC | Y14 | AD14 | Y16 | AD18 |
| NC | NC | TX_FAULT_INT[2] | NC | P23 | | | |

1. An external pull-up resistor is required with most optical modules.
2. An open drain I/O, external 4.7 k $\Omega$ pull-up resistor is required.

**Table 16**      **Line Side Interface Multiplexed Balls (Sheet 2 of 2)**

| Copper Mode | | Fiber Mode | Unused Port | Ball Designator | | | |
|---|---|---|---|---|---|---|---|
| GMII Signal | RGMII Signal | Optical Module/ SerDes Signal | | | | | |
| NC | NC | RX_LOS_INT[2] | NC | P19 | | | |
| NC | NC | MOD_DEF_INT[2] | NC | N22 | | | |
| MDC | MDC | NC | NC | W24 | | | |
| MDIO[2] | MDIO[2] | NC | NC | V21 | | | |
| NC | NC | $I^2$C_CLK | NC | L23 | | | |
| NC | NC | $I^2$C_DATA_0:3[2] | NC | L24 | M24 | N24 | P24 |

1. An external pull-up resistor is required with most optical modules.
2. An open drain I/O, external 4.7 k $\Omega$ pull-up resistor is required.

## 4.5.2     SPI3 MPHY/SPHY Ball Connections

Table 17 lists the balls used for the SPI3 Interface and provides a guide to connect these balls in MPHY and SPHY mode.

**Table 17**      **SPI3 MPHY/SPHY Interface (Sheet 1 of 3)**

| SPI3 Signals | | Ball Number | | | | Comments |
|---|---|---|---|---|---|---|
| MPHY | SPHY | | | | | |
| TDAT[31:24] | TDAT[7:0]_3 | F7 G7 | F5 G6 | G9 G5 | G8 G4 | **MPHY:** Consists of a single 32-bit data bus **SPHY:** Separate 8-bit data bus for each Ethernet port |
| TDAT[23:16] | TDAT[7:0]_2 | C8 E8 | F9 E7 | E10 E6 | E9 E5 | |
| TDAT[15:8] | TDAT[7:0]_1 | H3 H1 | J3 G2 | J2 G1 | J1 F1 | |
| TDAT[7:0] | TDAT[7:0]_0 | C6 D1 | B5 C3 | C5 C2 | C4 B3 | |
| TFCLK | TFCLK | D7 | | | | To achieve maximum bandwidth, set TFCLK as follows: **MPHY:** 133 MHz **SPHY:** 125 MHz. |
| TPRTY_0 | TPRTY_0 | D5 | | | | **MPHY:** Use TPRTY_0 as the TPRTY signal. **SPHY:** Each port has its own dedicated TPRTY_n signal. |
| GND | TPRTY_1 | G3 | | | | |
| GND | TPRTY_2 | B9 | | | | |
| GND | TPRTY_3 | J6 | | | | |
| TENB_0 | TENB_0 | B7 | | | | **MPHY:** Use TENB_0 as the TENB signal. **SPHY:** Each port has its own dedicated TENB_n signal. |
| VDD2 | TENB_1 | E2 | | | | |
| VDD2 | TENB_2 | C9 | | | | |
| VDD2 | TENB_3 | J4 | | | | |

### Table 17    SPI3 MPHY/SPHY Interface (Sheet 2 of 3)

| SPI3 Signals | | Ball Number | | | | Comments |
|---|---|---|---|---|---|---|
| MPHY | SPHY | | | | | |
| TERR_0 | TERR_0 | A8 | | | | **MPHY:** Use TERR_0 as the TERR signal. **SPHY:** Each port has its own dedicated TERR_*n* signal |
| GND | TERR_1 | K1 | | | | |
| GND | TERR_2 | E11 | | | | |
| GND | TERR_3 | J8 | | | | |
| TSOP_0 | TSOP_0 | C7 | | | | **MPHY:** Use TSOP_0 as the TSOP signal. **SPHY:** Each port has a dedicated TSOP_*n* signal. |
| GND | TSOP_1 | E3 | | | | |
| GND | TSOP_2 | C10 | | | | |
| GND | TSOP_3 | J5 | | | | |
| TEOP_0 | TEOP_0 | A7 | | | | **MPHY:** Use TEOP_0 as the TEOP signal. **SPHY:** Each port has a dedicated TEOP_*n* signal. |
| GND | TEOP_1 | F3 | | | | |
| GND | TEOP_2 | E4 | | | | |
| GND | TEOP_3 | H5 | | | | |
| TMOD[1:0] | GND | D9 | A6 | | | TSX and TMOD[1:0] are only applicable in MPHY mode. |
| TSX | GND | E1 | | | | |
| TADR[1:0] | TADR[1:0] | A12 | A11 | | | Used to address port for PTPA signal. |
| PTPA | PTPA | B11 | | | | PTPA can be used in MPHY and SPHY modes. |
| DTPA_0:3 | DTPA_0:3 | D3 | L1 | A9 | J7 | DTPA is available on a per-port basis in both MPHY and SPHY modes. |
| STPA | NC | C11 | | | | STPA is only applicable in MPHY mode. |
| RDAT[31:24] | RDAT[7:0]_3 | F24 G21 | G24 G20 | G23 G19 | G22 G18 | |
| RDAT[23:16] | RDAT[7:0]_2 | E21 C21 | E22 C20 | D22 B22 | C22 B20 | **MPHY:** Consists of a single 32 bit data bus. **SPHY:** Separate 8-bit data bus for each Ethernet port. |
| RDAT[15:8] | RDAT[7:0]_1 | F18 E16 | E18 D16 | E17 C17 | F16 A17 | |
| RDAT[7:0] | RDAT[7:0]_0 | F14 C14 | E14 B14 | D14 A15 | C13 A14, | |
| RFCLK | RFCLK | A19 | | | | To achieve maximum bandwidth, set RFCLK as follows: **MPHY:** 133 MHz. **SPHY:** 125 MHz. |
| RPRTY_0 | RPRTY_0 | E15 | | | | **MPHY:** Use RPRTY_0 as the RPRTY signal. **SPHY:** Each port has a dedicated RPRTY_*n* signal. |
| NC | RPRTY_1 | G16 | | | | |
| NC | RPRTY_2 | E20 | | | | |
| NC | RPRTY_3 | F20 | | | | |
| RENB_0 | RENB_0 | A13 | | | | **MPHY:** Use RENB_0 as the RENB signal. **SPHY:** Each port has a dedicated RENB_*n* signal |
| VDD2 | RENB_1 | A18 | | | | |
| VDD2 | RENB_2 | C19 | | | | |
| VDD2 | RENB_3 | E24 | | | | |

**Table 17**     **SPI3 MPHY/SPHY Interface (Sheet 3 of 3)**

| SPI3 Signals | | Ball Number | Comments |
|---|---|---|---|
| **MPHY** | **SPHY** | | |
| RERR_0 | RERR_0 | A16 | **MPHY:** Use RERR_0 as the RERR signal.<br>**SPHY:** Each port has a dedicated RERR_n signal |
| NC | RERR_1 | G17 | |
| NC | RERR_2 | D20 | |
| NC | RERR_3 | H20 | |
| RVAL_0 | RVAL_0 | C15 | **MPHY:** Use RVAL_0 as the RVAL signal.<br>**SPHY:** Each port has a dedicated RVAL_n signal. |
| NC | RVAL_1 | B18 | |
| NC | RVAL_2 | E19 | |
| NC | RVAL_3 | F22 | |
| RSOP_0 | RSOP_0 | B16 | **MPHY:** Use TSOP_0 as the TSOP signal.<br>**SPHY:** Each port has a dedicated TSOP_n signal. |
| NC | RSOP_1 | C18 | |
| NC | RSOP_2 | E23 | |
| NC | RSOP_3 | J18 | |
| REOP_0 | REOP_0 | C16 | **MPHY:** Use TEOP_0 as the TEOP signal.<br>**SPHY:** Each port has a dedicated TEOP_n signal. |
| NC | REOP_1 | D18 | |
| NC | REOP_2 | C23 | |
| NC | REOP_3 | J19 | |
| RMOD[1:0] | NC | G13    G14 | RSX and RMOD[1:0] are applicable only in MPHY mode. |
| RSX | NC | E13 | |

## 4.6     Ball State During Reset

**Table 18**     **Definition of Output and Bi-directional Balls During Hardware Reset (Sheet 1 of 2)**

| Interface | Ball Name | Ball Reset State | Comment |
|---|---|---|---|
| SPI3 | DTPA_0:3 | 0x0 | – |
| | STPA | 0x0 | – |
| | PTPA | 0x0 | – |
| | RDAT[31:0] | 0x00000000 | – |
| | RVAL_0:3 | 0x0 | – |
| | RERR_0:3 | 0x0 | – |
| | RPRTY_0:3 | 0x0 | – |
| | RMOD[1:0] | 0x0 | – |
| | RSX | 0x0 | – |
| | RSOP_0:3 | 0x0 | – |
| | REOP_0:3 | 0x0 | – |
| *Note:*   Z = High impedance. | | | |

**Table 18     Definition of Output and Bi-directional Balls During Hardware Reset (Sheet 2 of 2)**

| Interface | Ball Name | Ball Reset State | Comment |
|---|---|---|---|
| JTAG | TDO | 0x0 | – |
| MDIO | MDIO | High Z | Bi-directional |
|  | MDC | 0x0 | – |
| CPU | UPX_DATA[31:0] | High Z | Bi-directional |
|  | UPX_RDY_L | 0X1 | Open-drain output, requires an external pull-up |
| LED | LED_CLK | 0x0 | – |
|  | LED_DATA | 0x0 | – |
|  | LED_LATCH | 0x0 | – |
| GMII/RGMII | TXC_0:3 | High Z | Fiber mode is the default. Copper interfaces are disabled. |
|  | TXD[7:0]_0 | High Z | Fiber mode is the default.<br>Bit 4 is driven by the optical module as MOD_DEF_0. |
|  | TXD[7:0]_1 | High Z | Fiber mode is the default.<br>Bit 4 is driven by the optical module as MOD_DEF_1. |
|  | TXD[7:0]_2 | High Z | Fiber mode is the default.<br>Bit 4 is driven by the optical module as MOD_DEF_2. |
|  | TXD[7:0]_3 | High Z | Fiber mode is the default.<br>Bit 4 is driven by the optical module as MOD_DEF_3. |
|  | TX_EN_0:3 | High Z | Fiber mode is the default.<br>Copper interfaces are disabled. |
|  | TX_ER_0:3 | High Z | Fiber mode is the default.<br>Copper interfaces are disabled. |
| RGMII | TX_CTL_0:3 | High Z | Fiber mode is the default.<br>Copper interfaces are disabled. |
| SerDes | TX_P_0:3 | 0x0 | – |
|  | TX_N_0:3 | 0x0 | – |
| Optical Module | TX_FAULT_INT | High Z | Open-drain output, requires external pull-up. |
|  | RX_LOS_INT | High Z | Open-drain output, requires external pull-up. |
|  | MOD_DEF_INT | High Z | Open-drain output, requires external pull-up. |
|  | I$^2$C_CLK | 0x1 | – |
|  | I$^2$C_DATA_0:3 | 0xF | Open-drain output, requires external pull-up. |

*Note:*   Z = High impedance.

## 4.7     Power Supply Sequencing

Follow the power-up and power-down sequences described in this section to ensure correct IXF1104 MAC operation. The sequence described in Section 4.7 covers all IXF1104 MAC digital and analog supplies.

**Caution:**     Failure to follow the sequence described in this section might damage the IXF1104 MAC.

### 4.7.1 Power-Up Sequence

Ensure that the 1.8 V analog and digital supplies are applied and stable prior to application of the 2.5 V analog and digital supplies.

### 4.7.2 Power-Down Sequence

Remove the 2.5 V supplies prior to removing the 1.8 V power supplies (the reverse of the power-up sequence).

**Caution:** Damage can occur to the ESD structures within the analog I/Os if the 2.5 V digital and analog supplies exceed the 1.8 V digital and analog supplies by more than 2.0 V during power-up or power-down.

Figure 5 and Table 19 provide the IXF1104 MAC power supply sequencing.

**Figure 5      Power Supply Sequencing**



Note:    The 3.3 V supply (VDD2 and VDD3) can be applied at any point during this sequence.
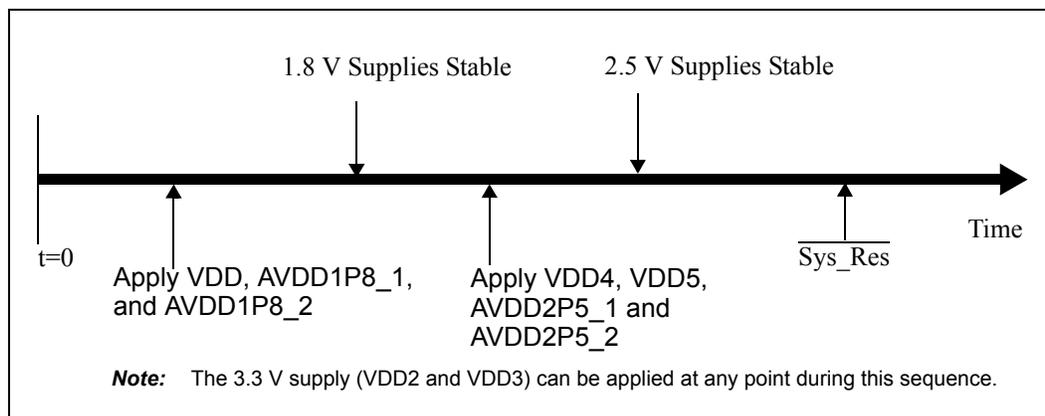
**Table 19      Power Supply Sequencing**

| Power Supply | Power-Up Order | Time Delta to Next Supply[1] | Notes |
|---|---|---|---|
| VDD, AVDD1P8_1, AVDD1P8_2 | First | 0 | 1.8 V supplies |
| VDD4, VDD5, AVDD2P5_1, AVDD2P5_2 | Second | 10 µs | 2.5 V supplies |

1.  The value of 10 µs given is a nominal value only. The exact time difference between the application of the 2.5 V analog supply is determined by a number of factors, depending on the power management method used.
*Note:*    To avoid damage to the IXF1104 MAC, the TXAV25 supply must not exceed the VDD supply by more than 2 V at any time during the power-up or power-down sequence.
*Note:*    The 3.3 V supply (VDD2 and VDD3) can be applied at any point during this sequence.

## 4.8      Pull-Up/Pull-Down Ball Guidelines

The signals shown in Table 20 require the addition of a pull-up or pull-down resistor to the board design for normal operation. Any balls marked as unused (NC) should be unconnected.

**Table 20     Pull-Up/Pull-Down and Unused Ball Guidelines**

| Pin Name | Pull-Up/Pull-Down | Comments |
|---|---|---|
| TX_FAULT_INT | Pull-up | 4.7 k $\Omega$ to 2.5 V. Optical module signal with open-drain I/O. |
| RX_LOS_INT | Pull-up | 4.7 k $\Omega$ to 2.5 V. Optical module signal with open-drain I/O. |
| MOD_DEF_INT | Pull-up | 4.7 k $\Omega$ to 2.5 V. Optical module signal with open-drain I/O. |
| TDI | Pull-up | 10 k $\Omega$ to 3.3 V. JTAG test pin. |
| TDO | Pull-up | 10 k $\Omega$ to 3.3 V. JTAG test pin. |
| TMS | Pull-up | 10 k $\Omega$ to 3.3 V. JTAG test pin. |
| TCLK | Pull-up | 10 k $\Omega$ to 3.3 V. JTAG test pin. |
| TRST_L | Pull-down | 10 k $\Omega$ to 3.3 V. JTAG test pin. |
| MDIO | Pull-up | 4.7 k $\Omega$ to 2.5 V |
| UPX_RDY_L | Pull-up | 4.7 k $\Omega$ to 3.3 V |
| $I^2C$_DATA_0:3 | Pull-up | 4.7 k $\Omega$ to 2.5 V |
| TX_DISABLE_0:3 | Pull-up | 4.7 k $\Omega$ to 2.5 V |

## 4.9     Analog Power Filtering

Table 21 illustrates an analog power supply filter network and Table 21 lists the analog power balls.

**Figure 6     Analog Power Supply Filter Network**
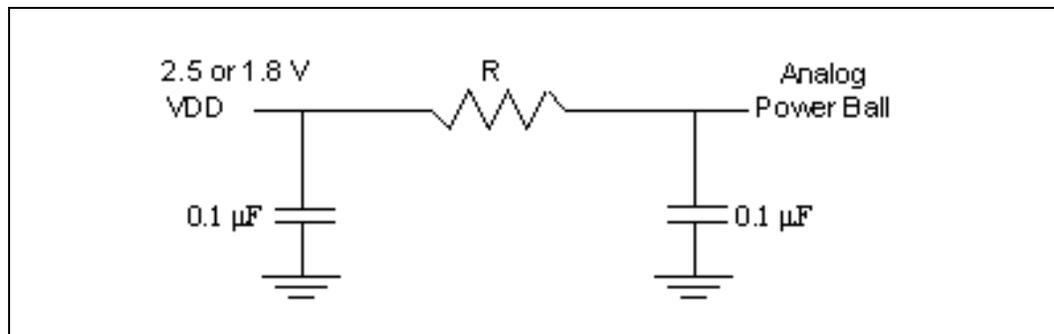


**Table 21     Analog Power Balls**

| Signal Name | Ball Designator | | Comments |
|---|---|---|---|
| AVDD1P8_1 | A5 | A20 | Need to provide a filter (see Figure 6). |
| AVDD2P5_1 | AD20 | | **R:** AVDD1P8_1 and AVDD2P5_1 = 5.6 $\Omega$ resistor. |
| AVDD1P8_2 | AB16 | T23 | Need to provide a filter (see Figure 6). |
| AVDD2P5_2 | U14 | R18 | **R:** AVDD1P8_2 and AVDD2P5_2 = 1.0 $\Omega$ resistor. |

# 5.0 Functional Descriptions

## 5.1 Media Access Controller (MAC)

The IXF1104 MAC main functional block consists of four independent 10/100/1000 Mbps Ethernet MACs, which support interfaces for fiber and copper connectivity.

- Copper Mode:
  — RGMII for 10/100/1000 Mbps full-duplex operation and 10/100 Mbps half-duplex operation
  — GMII for 1000 Mbps full-duplex operation
- Fiber Mode:
  — Integrated SerDes/OMI interface for direct connection to optical modules
  — 1000 Mbps full-duplex operation in fiber mode

The following features support copper and fiber modes:

- Programmable Options:
  — Automatic padding of transmitted packets that are less than the minimum frame size
  — Broadcast, multicast, and unicast address filtering on frames received
  — Filter and drop packets with errors
  — Pre-padded RX frames with two bytes (aligns the Ethernet payload on SPI3 and in network processor memories)
  — Remove CRC from RX frames
  — Append CRC to transmitted frames
- Performance Monitoring and Diagnostics:
  — Loopback modes
  — Detection of runt and overly large packets
  — Cyclic Redundancy Check (CRC) calculation and error detection
  — RMON statistics for dropped packets, packets with errors, etc.
- Compliant with IEEE Spec 802.3x standard for flow control
  — Receive and execute PAUSE Command Frames
- Support for non-standard packet sizes up to 10 KB including loss-less flow control

**Note:** The IXF1104 MAC does not support 10/100 Mbps operation when configured in GMII mode.

The IXF1104 MAC is fully integrated, designed for use with Ethernet 802.3 frame types, and compliant to all of the IEEE 802.3 MAC requirements.

The IXF1104 MAC adds preamble and Start-of-Frame Delimiter (SFD) to all frames sent to it (transmit path) and removes preamble and SFD on all frames received by it (receive path). A CRC check is also applied to all transmit and receive packets. CRC is optionally appended to transmit packets. CRC is removed optionally from receive packets after validation, and is not forwarded to SPI3. Packets with a bad CRC are marked, counted in the statistics block, and may be optionally dropped. A bad packet may be signaled with RERR on the SPI3 interface if it is not dropped.

The IXF1104 MAC operates only in full-duplex mode at 1000 Mbps rates on both SerDes and GMII interface connections. The IXF1104 MAC is capable of operation at 1000 Mbps, full-duplex in RGMII mode, and at full-duplex and half-duplex operation for 10/100 Mbps links.

## 5.1.1 Features for Fiber and Copper Mode

Section 5.1.1.1 through Section 5.1.1.4 cover IXF1104 MAC functions that are independent of the line-side interface.

### 5.1.1.1 Padding of Undersized Frames on Transmit

The padding feature allows Ethernet frames smaller than 64 bytes to be transferred from the SPI3 interface to the TX MAC and padded up to 64 bytes automatically by the MAC. This feature is enabled by setting bit 7 of the *Diverse Config Write ($ Port_Index + 0x18)*.

**Note:** When the user selects the padding function, the MAC core adds an automatically calculated CRC to the end of the transmitted packet.

### 5.1.1.2 Automatic CRC Generation

Automatic CRC Generation is used in conjunction with the padding feature to generate and append a correct CRC to any transmit frame. This feature is enabled by setting bit 6 of the *Diverse Config Write ($ Port_Index + 0x18)*.

### 5.1.1.3 Filtering of Receive Packets

This feature allows the IXF1104 MAC to filter receive packets under various conditions and drop the packets through an interaction with the Receive FIFO control.

#### 5.1.1.3.1 Filter on Unicast Packet Match

This feature is enabled when bit 0 of the *RX Packet Filter Control ($ Port_Index + 0x19)* = 1. Any frame received in this mode that does not match the Station Address (MAC address) is marked by the IXF1104 MAC to be dropped. The frame is dropped if the appropriate bit in the *RX FIFO Errored Frame Drop Enable ($0x59F)* = 1. Otherwise, the frame is sent out the SPI3 interface and may optionally be signaled with an RERR (see bit 0 in Section Table 147, *SPI3 Receive Configuration ($0x701),* on page 207).

When bit 0 of the *RX Packet Filter Control ($ Port_Index + 0x19)* = 0, all unicast frames are sent out the SPI3 interface.

**Note:** The VLAN filter overrides the unicast filter. Therefore, a VLAN frame cannot be filtered based on the unicast address.

#### 5.1.1.3.2 Filter on Multicast Packet Match

This feature is enabled when bit 1 of the *RX Packet Filter Control ($ Port_Index + 0x19)* = 1. Any frame received in this mode that does not match the Port Multicast Address (reserved multicast address recognized by IXF1104 MAC) is marked by the MAC to be dropped. The frame is dropped if the appropriate bit in the *RX FIFO Errored Frame Drop Enable ($0x59F)* = 1. Otherwise, the frame is sent out the SPI3 interface and may optionally be signaled with an RERR (see bit 0 in Section Table 147, *SPI3 Receive Configuration ($0x701),* on page 207).

When bit 1 of the *RX Packet Filter Control ($ Port_Index + 0x19)* = 0, all multicast frames are sent out the SPI3 interface.

### 5.1.1.3.3    Filter Broadcast Packets

This feature is enabled when bit 2 of the *RX Packet Filter Control ($ Port_Index + 0x19)* = 1. Any broadcast frame received in this mode is marked by the MAC to be dropped. The frame is dropped if the appropriate bit in the *RX FIFO Errored Frame Drop Enable ($0x59F)* = 1. Otherwise, the frame is sent out the SPI3 interface and may optionally be signaled with an RERR (see bit 0 in Section Table 147, *SPI3 Receive Configuration ($0x701),* on page 207).

When bit 2 of the *RX Packet Filter Control ($ Port_Index + 0x19)* = 0, all broadcast frames are sent out the SPI3 interface.

### 5.1.1.3.4    Filter VLAN Packets

This feature is enabled when bit 3 of the *RX Packet Filter Control ($ Port_Index + 0x19)* = 1. VLAN frames received in this mode are marked by the MAC to be dropped. The frame is dropped if the appropriate bit in the *RX FIFO Errored Frame Drop Enable ($0x59F)* = 1. Otherwise, the VLAN frame is sent out the SPI3 interface and may optionally be signaled with an RERR (see bit 0 in Section Table 147, *SPI3 Receive Configuration ($0x701),* on page 207).

When bit 3 of the *RX Packet Filter Control ($ Port_Index + 0x19)* = 0, all VLAN frames are sent out the SPI3 interface.

### 5.1.1.3.5    Filter Pause Packets

This feature is enabled when bit 4 of the *RX Packet Filter Control ($ Port_Index + 0x19)* = 0. Pause frames received in this mode are marked by the MAC to be dropped. The frame is dropped if the appropriate bit in the *RX FIFO Errored Frame Drop Enable ($0x59F)* = 1. Otherwise, the pause frame is sent out the SPI3 interface and may optionally be signaled with an RERR (see bit 0 in Section Table 147, *SPI3 Receive Configuration ($0x701),* on page 207).

When bit 4 of the *RX Packet Filter Control ($ Port_Index + 0x19)* = 1, all pause frames are sent out the SPI3 interface.

**Note:**    Pause packets are not filtered if flow control is disabled in the Section 84, *FC Enable ($ Port_Index + 0x12)*.

### 5.1.1.3.6    Filter CRC Error Packets

This feature is enabled when bit 5 of the Section 91, *RX Packet Filter Control ($ Port_Index + 0x19)* = 0. Frames received with an errored CRC are marked as bad frames and may optionally be dropped in the RX FIFO. Otherwise, the frames are sent to the SPI3 interface and may be optionally signaled with an RERR (see Table 22, *CRC Errored Packets Drop Enable Behavior*, on page 69).

When the CRC Error Pass Filter bit = 0 (Section 91, *RX Packet Filter Control ($ Port_Index + 0x19)*), it takes precedence over the other filter bits. Any packet (Pause, Unicast, Multicast or Broadcast packet) with a CRC error will be marked as a bad frame when the CRC Error Pass Filter bit = 0.

**Table 22      CRC Errored Packets Drop Enable Behavior**

| CRC Error Pass[1] | RX FIFO ErroredFrame Drop Enable[2] | RERR Enable[3] | Actions |
|---|---|---|---|
| 1 | x | x | When CRC Errored PASS = 1, CRC errored packets are not filtered and are passed to the SPI3 interface. They are not marked as bad, cannot be dropped, and cannot be signaled with RERR. |
| 0 | 0 | 1 | Packets are marked as bad but not dropped in the RX FIFO. These packets are sent to the SPI3 interface, and are signaled with an RERR to the switch or Network Processor. |
| 0 | 0 | 0 | Packets are marked as bad but not dropped in the RX FIFO. These packets are sent to the SPI3 interface, and are not signaled with an RERR. |
| 0 | 1 | x | CRC errored packets are marked as bad, dropped in the RX FIFO, and never appear at the SPI3 interface.<br><br>*Note:* Packet sizes above the RX FIFO Transfer Threshold (see Table 128 through Table 131) cannot be dropped in the RX FIFO and are passed to the SPI3 interface. These packets can optionally be signaled with RERR on the SPI3 interface if the RERR Enable bit = 1. |

1. See Table 91, *RX Packet Filter Control ($ Port_Index + 0x19)*, on page 165.
2. See Table 123, *RX FIFO Errored Frame Drop Enable ($0x59F)*, on page 189.
3. See Table 147, *SPI3 Receive Configuration ($0x701)*, on page 207.
*Note:*    x = "DON'T CARE"

## 5.1.1.4      CRC Error Detection

Frames received by the MAC are checked for a correct CRC. When an incorrect CRC is detected on a received frame, the RX FCSError RMON statistic counter increments for each CRC errored frame. Received frames with CRC errors may optionally be dropped in the RX FIFO (refer to Section 5.1.1.3.6, *Filter CRC Error Packets*, on page 68). Otherwise, the frames are sent to the SPI3 interface and may be dropped by the switch or system controller.

Frames transmitted by the MAC are also checked for correct CRC. When an incorrect CRC is detected on a transmitted frame, the TX CRCError RMON statistic counter increments for each incorrect frame.

## 5.1.2      Flow Control

Flow Control is an IEEE 802.3x-defined mechanism for one network node to request that its link partner take a temporary "Pause" in packet transmission. This allows the requesting network node to prevent FIFO overruns and dropped packets, by managing incoming traffic to fit its available memory. The temporary pause allows the device to process packets already received or in transit, thus freeing up the FIFO space allocated to those packets.

The IXF1104 MAC MAC implements the IEEE 802.3x standard RX FIFO threshold-based Flow Control in copper and fiber modes. When appropriately programmed, the MAC can both generate and respond to IEEE standard pause frames in full-duplex operation. The IXF1104 MAC also supports externally triggered flow control through the Transmit Pause Control interface.

In half-duplex operation, the MAC generates collisions instead of sending pause frames to manage the incoming traffic from the link partner

### 5.1.2.1    802.3x Flow Control (Full-Duplex Operation)

The IEEE 802.3x standard identifies four options related to system flow control:

- No Pause
- Symmetric Pause (both directions)
- Asymmetric Pause (Receive direction only)
- Asymmetric Pause (Transmit direction only)

The IXF1104 MAC supports all four options on a per-port basis. Bits 2:0 of the Table 84 on page 160 provide programmable control for enabling or disabling flow control in each direction independently.

The IEEE 802.3x flow control mechanism is accomplished within the MAC sublayer, and is based on RX FIFO thresholds called watermarks. The RX FIFO level rises and falls as packets are received and processed. When the RX FIFO reaches a watermark (either exceeding a High or dropping below a Low after exceeding a High), the IXF1104 MAC control sublayer signals an internal state machine to transmit a PAUSE frame. The FIFOs automatically generate PAUSE frames (also called control frames) to initiate the following:

- Halt the link partner when the High watermark is reached.
- Restart the link partner when the data stored in the FIFO falls below the Low watermark.

Figure 7 illustrates the IEEE 802.3 FIFO flow control functions.

**Figure 7      Packet Buffering FIFO**
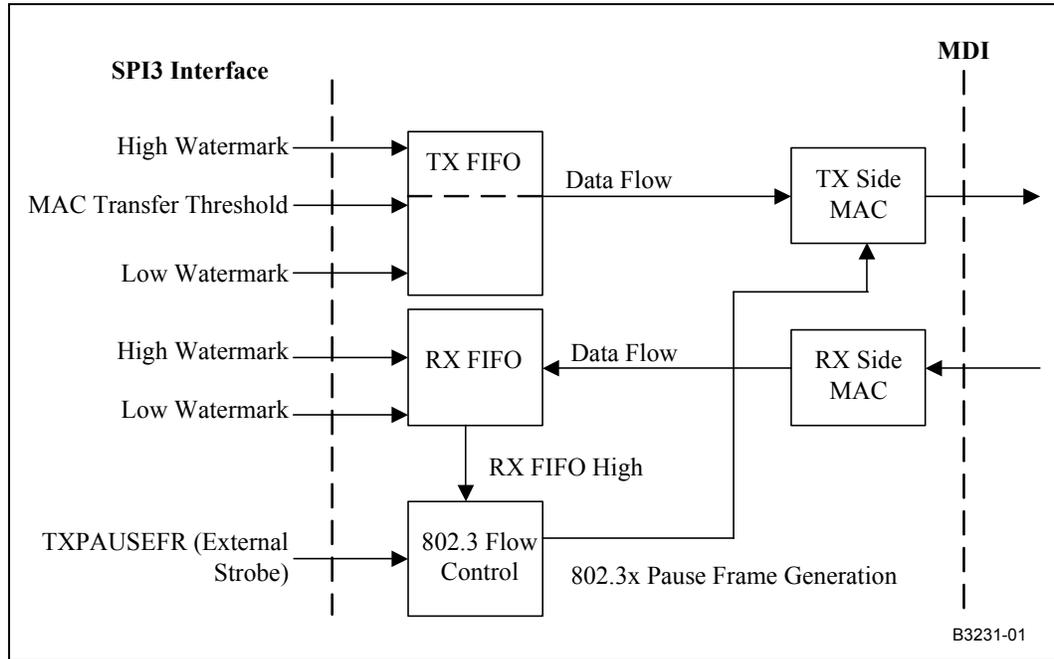


### 5.1.2.1.1      Pause Frame Format

PAUSE frames are MAC control frames that are padded to the minimum size (64 bytes). Figure 8 and Figure 9 illustrate the frame format and contents.

**Figure 8      Ethernet Frame Format**

**Figure 9     PAUSE Frame Format**



Note: In the IXF1104 architecture, the TX block of the MAC sets this as the pause multicast address.
The RX interface of the MAC will process this as the pause multicast or the MAC address.

B3218-03
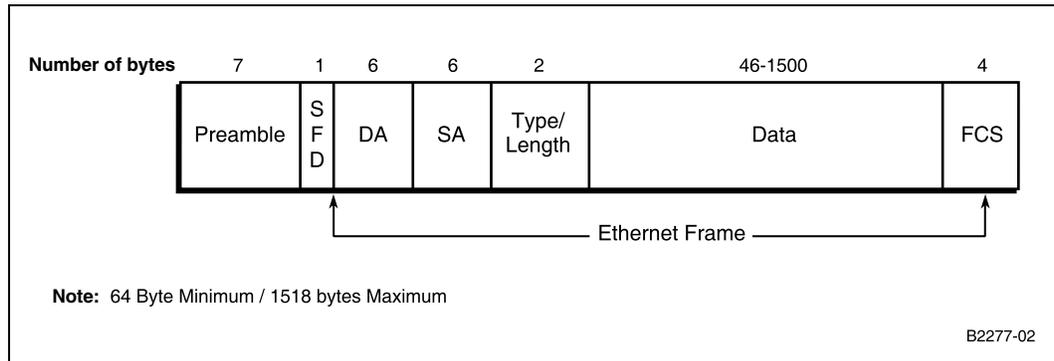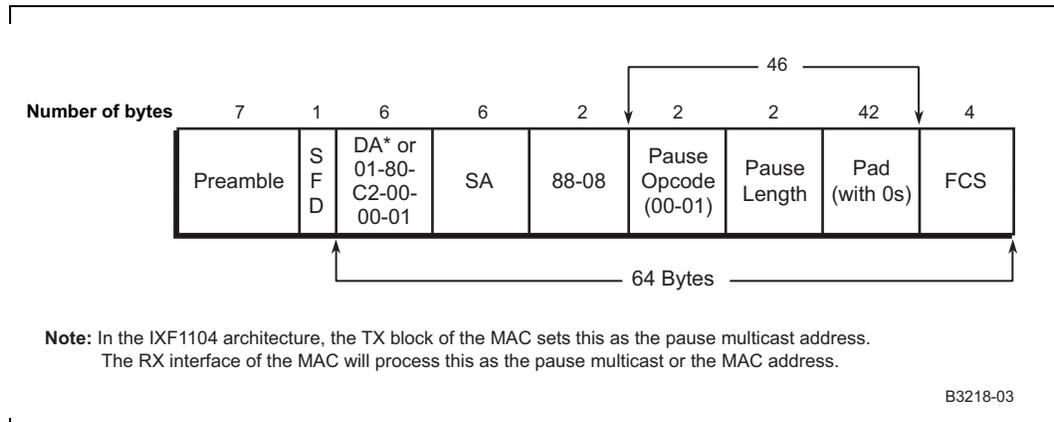
An IEEE 802.3 MAC PAUSE frame is identified by detecting all of the following:

- OpCode of 00-01
- Length/Type field of 88-08
- DA matching the unique multicast address (01-80-C2-00-00-01)

**XOFF.** A PAUSE frame informs the link partner to halt transmission for a specified length of time. The PauseLength octets specify the duration of the no-transmit period. If this time is greater than zero, the link partner must stop sending any further packets until this time has elapsed. This is referred to as XOFF.

**XON.** The MAC continues to transmit PAUSE frames with the specified Pause Length as long as the FIFO level exceeds the threshold. If the FIFO level falls below the threshold before the Pause Length time expires, the MAC sends another PAUSE frame with the Pause Length time specified as zero. This is referred to as XON and informs the link partner to resume normal transmission of packets.

### 5.1.2.1.2     Pause Settings

The MAC must send PAUSE frames repeatedly to maintain the link partner in a Pause state. The following two inter-related variables control this process:

- Pause Length is the amount of time, measured in multiples of 512 bit times, that the MAC requests the link partner to halt transmission for.
- Pause Threshold is the amount of time, measured in multiples of 512 bit times, prior to the expiration of the Pause Length that the MAC transmits another Pause frame to maintain the link partner in the pause state.

The transmitted Pause Length in the IXF1104 MAC is set by the Table 75 on page 157.

The IXF1104 MAC PAUSE frame transmission interval is set by the Table 80 on page 159.

### 5.1.2.1.3     Response to Received PAUSE Command Frames

When Flow Control is enabled in the receive direction (bit 0 in the *FC Enable ($ Port_Index + 0x12)*), the IXF1104 MAC responds to PAUSE Command frames received from the link partner as follows:

1. The IXF1104 MAC checks the entire frame to verify that it is a valid PAUSE control frame addressed to the Multicast Address 01-80-C2-00-00-01 (as specified in IEEE

802.3, Annex 31B) or has a Destinations Address matching the address programmed in the *Station Address ($ Port_Index +0x00 – +0x01)*.

2. If the PAUSE frame is valid, the transmit side of the IXF1104 MAC pauses for the required number of PAUSE Quanta, as specified in IEEE 802.3, Clause 31.

3. PAUSE does not begin until completion of the frame currently being transmitted.

The IXF1104 MAC response to valid received PAUSE frames is independent of the PAUSE frame filter settings. Refer to Section 5.1.1.3.5, *Filter Pause Packets*, on page 68 for additional details.

**Note:** Pause packets are not filtered if flow control is disabled in bit 0 of the Section 84, *FC Enable ($ Port_Index + 0x12)*.

#### 5.1.2.1.4 Half-Duplex Operation

Transmit flow control is implemented only in half-duplex operation. Upon entering the flow control state, the MAC generates a collision for all subsequent receive packets until exiting the flow control state. Any receive packet in progress when the MAC enters the flow control state will not be collided with but could be lost due if there is insufficient FIFO depth to complete packet reception. Bit 2 of the *FC Enable ($ Port_Index + 0x12)* enables the transmit flow control function.

#### 5.1.2.1.5 Transmit Pause Control Interface

The Transmit Pause Control interface allows an external device to trigger the generation of pause frames. The Transmit Pause Control interface is completely asynchronous. It consists of three address signals (TXPAUSEADD[2:0]) and a strobe signal (TXPAUSEFR). The required address for this interface operation is placed on the TXPAUSEADD[2:0] signals and the TXPAUSEFR is pulsed High and returned Low. Refer to Figure 10, *Transmit Pause Control Interface*, on page 74 and Table 55, *Transmit Pause Control Interface Timing Parameters*, on page 145. Table 23 shows the valid decodes for the TXPAUSEADD[2:0] signals. Figure 10 illustrates the transmit pause control interface.

**Note:** Flow control must be enabled in the Section 84, *FC Enable ($ Port_Index + 0x12)* for Transmit Pause Control interface operation.

**Note:** There are two additional decodes provided that allow the user to generate either an XOFF frame or XON frame from all ports simultaneously.

The default pause quanta for each port is held by the *FC TX Timer Value ($ Port_Index + 0x07)*). The default value of this register is 0x05E after reset is applied.

**Table 23    Valid Decodes for TXPAUSEADD[2:0] (Sheet 1 of 2)**

| TXPAUSEADD_2:0 | Operation of TX Pause Control Interface |
| --- | --- |
| 0x0 | Transmits a PAUSE frame on every port with a pause_time = ZERO (XON) (Cancels all previous pause commands). |
| 0x1 | Transmits a PAUSE frame on port 0 with pause_time equal to the value programmed in the port 0 *FC TX Timer Value ($ Port_Index + 0x07)* (XOFF). |
| 0x2 | Transmits a PAUSE frame on port 1 with pause_time equal to the value programmed in the port 1 *FC TX Timer Value ($ Port_Index + 0x07)* (XOFF). |
| 0x3 | Transmits a PAUSE frame on port 2 with pause_time equal to the value programmed in the port 2 *FC TX Timer Value ($ Port_Index + 0x07)* (XOFF). |

**Table 23      Valid Decodes for TXPAUSEADD[2:0] (Sheet 2 of 2)**

| TXPAUSEADD_2:0 | Operation of TX Pause Control Interface |
|---|---|
| 0x4 | Transmits a PAUSE frame on port 3 with pause_time equal to the value programmed in the port 3 *FC TX Timer Value ($ Port_Index + 0x07)* (XOFF). |
| 0x5 to 0x6 | Reserved. Do not use these addresses. The TX Pause Control interface will not operate under these conditions. |
| 0x7 | Transmits a PAUSE frame on every port with pause_time equal to the value programmed in the *FC TX Timer Value ($ Port_Index + 0x07)* for each port (XOFF). |

**Figure 10      Transmit Pause Control Interface**



**This example shows the following conditions:**

**Strobe 1:**
- Port 0: Transmit Pause Packet (XOFF)

**Strobe 2:**
- All Ports: Transmit Pause Packet with pause_time = 0 (XON)

**Strobe 3:**
- Port 3: Transmit Pause Packet (XOFF)

B3234-01
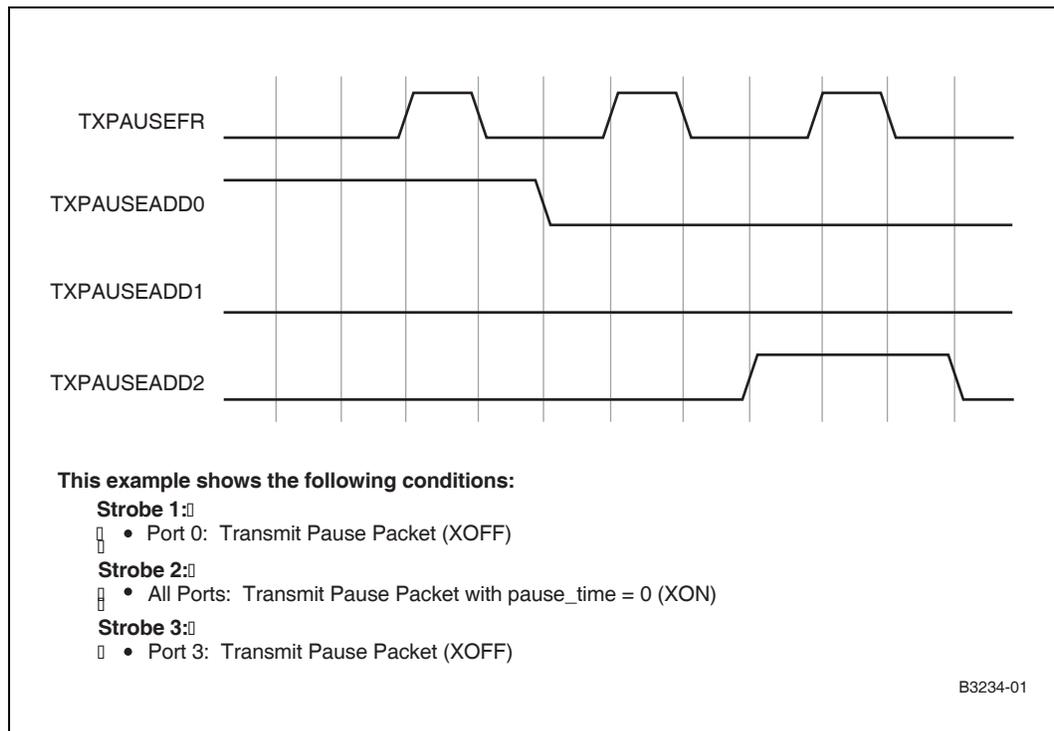
## 5.1.3      Mixed-Mode Operation

The IXF1104 MAC gives the user the option of configuring each port for 10/100 Mbps half-duplex copper, 10/100/1000 Mbps full-duplex copper, or 1000 Mbps full-duplex fiber operation. This gives the IXF1104 MAC the ability to support both copper and fiber operation line-side interfaces operating at the same time within a single device. (Refer to Table 16, *Line Side Interface Multiplexed Balls*, on page 59.)

The IXF1104 MAC provides complete flexibility in line-side connectivity by offering RGMII, integrated SerDes, and GMII.

### 5.1.3.1      Configuration

The memory maps (Table 59, *MAC Control Registers ($ Port Index + Offset)*, on page 149 through Table 69, *Optical Module Registers ($ 0x799 - 0x79F)*, on page 155) are logically split into the following two distinct regions:

- Per-Port Registers

- Global Registers

To achieve a desired configuration for a given port, the relevant per-port registers must be configured correctly by the user. The Table 59 through Table 69 also contain registers that affect the operation of all ports, such as the SPI3 interface configuration.

See Section 8.0, *Register Set*, on page 148 for a complete description of IXF1104 MAC configuration and status registers. The Register Maps (Table 59 through Table 69) present a summary of important configuration registers.

**Note:**      The initialization sequence provided in Section 6.1, *Change Port Mode Initialization Sequence*, on page 127 must be followed for proper configuration of the IXF1104 MAC.

### 5.1.3.2      Key Configuration Registers

The following key registers select the operational mode of a given port:

**Table 24      Operational Mode Configuration Registers**

| Register Name | Register Address | Description |
|---|---|---|
| *Desired Duplex ($ Port_Index + 0x02)* | 0x002 – Port 0<br>0x082 – Port 1<br>0x102 – Port 2<br>0x182 – Port 3 | The Table 71 on page 156 defines whether a port is to be configured for full-duplex or half-duplex operation.<br>***Note:***     Half-duplex operation is only valid for 10/100 speeds where the RGMII line interface has been selected. |
| *MAC IF Mode and RGMII Speed ($ Port_Index + 0x10)* | 0x010 – Port 0<br>0x090 – Port 1<br>0x110 – Port 2<br>0x190 – Port 3 | The Table 82 on page 160 determines the MAC operational frequency and mode for a given port.<br>***Note:***     Set the Table 152 on page 211 to 0x0 prior to any change in the register value. This ensures that a change in the MAC clock frequency is controlled correctly. If the *Clock and Interface Mode Change Enable Ports 0 - 3 ($0x794)* is not used correctly, the IXF1104 MAC may not be configured to the proper mode. |
| *Port Enable ($0x500)* | 0x500<br>Bit 0 – Port 0<br>Bit 1 – Port 1<br>Bit 2 – Port 2<br>Bit 3 – Port 3 | Each *Port Enable ($0x500)* bit relates to a port. Set the appropriate bit to 0x1 to enable a port. This should be the last step in the configuration process for a port. |
| *Interface Mode ($0x501)* | 0x501<br>Bit 0 – Port 0<br>Bit 1 – Port 1<br>Bit 2 – Port 2<br>Bit 3 – Port 3 | The *Interface Mode ($0x501)* selects whether a port operates with a copper (RGMII or GMII) line-side interface an integrated SerDes fiber line-side interface.<br>For copper operation for a given port, set the relevant bit to 0x1.<br>For fiber operation for a given port, set the relevant bit to 0x0.<br><br>***Note:***     All ports are configured for fiber operation in the IXF1104 MAC default mode of operation. |
| *Clock and Interface Mode Change Enable Ports 0 - 3 ($0x794)* | 0x794<br>Bit 0 – Port 0<br>Bit 1 – Port 1<br>Bit 2 – Port 2<br>Bit 3 – Port 3 | The *Clock and Interface Mode Change Enable Ports 0 - 3 ($0x794)* indicates to an internal clock generator when to sample the new value of the *MAC IF Mode and RGMII Speed ($ Port_Index + 0x10)* and the *Interface Mode ($0x501)* (copper/fiber).<br>When any of these two configuration values are changed for a port, the corresponding bits must be kept in this register under reset by writing 0x0 to the relevant bit. |
| ***Note:***     The initialization sequence provided in Section 6.1, *Change Port Mode Initialization Sequence*, on page 127 must be followed for proper configuration of the IXF1104 MAC. | | |

### 5.1.4　Fiber Mode

When the IXF1104 MAC is configured for fiber mode, the TX Data path from the MAC is an internal
10-bit interface as described in the IEEE 802.3z specification. It is connected directly to an internal SerDes block for serialization/deserialization and transmission/reception on the fiber medium to and from the link partner.

The MAC contains all of the PCS (8B/10B encoding and 10B/8B decoding) required to encode and decode the data. The MAC also supports auto-negotiation per the IEEE 802.3z specification via access to the *TX Config Word ($ Port_Index + 0x17)*, *RX Config Word ($ Port_Index + 0x16)*, and *Diverse Config Write ($ Port_Index + 0x18)*.

When configured for fiber mode, the full set of Optical Module interface control and status signals is presented through re-use of GMII signals on a per-port basis (see *4.5, Multiplexed Ball Connections*, on page 59). Fiber mode supports only full-duplex Gigabit operation.

#### 5.1.4.1　Fiber Auto-Negotiation

Auto-negotiation is performed by using the *TX Config Word ($ Port_Index + 0x17)*, *RX Config Word ($ Port_Index + 0x16)*, and *Diverse Config Write ($ Port_Index + 0x18)*. When autoneg_enable (*Diverse Config Write ($ Port_Index + 0x18)*) is set, the IXF1104 MAC performs hardware-defined auto-negotiation with the *TX Config Word ($ Port_Index + 0x17)* used as an *Auto-Negotiation Advertisement ($ Port Index + 0x64)* and the *RX Config Word ($ Port_Index + 0x16)* used as an *Auto-Negotiation Link Partner Base Page Ability ($ Port Index + 0x65)*.

**Note:**　While the MAC supports auto-negotiation functions, the IXF1104 MAC does not automatically configure the MAC or other device blocks to be consistent with the auto-negotiation results. This configuration is done by the user and system software.

#### 5.1.4.2　Determining If Link Is Established in Auto-Negotiation Mode

A valid link is established when the AN_complete bit is set and the RX_Sync bit reports that synchronization has occurred. Both register bits are located in the *RX Config Word ($ Port_Index + 0x16)*.

If the link goes down after auto-negotiation is completed, RX_Sync indicates that a loss of synchronization occurred. The IXF1104 MAC restarts auto-negotiation and attempts to reestablish a link. Once a link is reestablished, the AN_complete bit is set and the RX_Sync bit shows that synchronization has occurred.

To manually restart auto-negotiation, bit 5 of the Section 90, *Diverse Config Write ($ Port_Index + 0x18)* (AN_enable) must be de-asserted, then re-asserted.

#### 5.1.4.3　Fiber Forced Mode

The MAC fiber operation can be forced to operate at 1000 Mbps full-duplex without completion of the auto-negotiation function. In this mode, the MAC RX path must achieve synchronization with the link partner. Once achieved, the MAC TX path is enabled to allow data transmission. This forced mode is limited to operation with a link partner that operates with a full-duplex link at 1000 Mbps.

### 5.1.4.4    Determination of Link Establishment in Forced Mode

When the IXF1104 MAC is in forced mode operation, the Section 88, *RX Config Word ($ Port_Index + 0x16)* bit 20 RX Sync indicates when synchronization occurs and a valid link establishes.

**Note:**    The RX Sync bit indicates a loss of synchronization when the link is down.

## 5.1.5    Copper Mode

In copper mode, the IXF1104 MAC transmits data on the egress path of the RGMII or GMII interface, depending on the port configuration defined by the user. The copper MAC receives data on the ingress path of the RGMII or GMII interface, depending on the port configuration defined by the user. The RGMII interface supports operation at 10/100/1000 Mbps when a full-duplex link is established, and supports 10/100 Mbps when a half-duplex link is established. The GMII interface only supports a 1000 Mbps full-duplex link.

### 5.1.5.1    Speed

The copper MAC supports 10 Mbps, 100 Mbps, and 1000 Mbps. All required speed adjustments, clocks, etc., are supplied by the MAC. The operating speed of the MAC is programmable through the *MAC IF Mode and RGMII Speed ($ Port_Index + 0x10)* (MAC_IF_Mode). The IXF1104 MAC speed setting must be programmed by the system software to match the speed of the attached PHY for proper IXF1104 MAC operation.

**Note:**    When the IXF1104 MAC is configured to use the GMII interface, the only mode of operation that is supported is 1000 Mbps full-duplex.

If 10/100 Mbps operation is required in either full-duplex or half-duplex, the IXF1104 MAC must be configured to use the RGMII interface.

### 5.1.5.2    Duplex

The MAC supports full-duplex or half-duplex depending on the line-side interface that is configured by the *MAC IF Mode and RGMII Speed ($ Port_Index + 0x10)* (MAC_IF_Mode). The duplex of the MAC is set in the Section Table 71, *Desired Duplex ($ Port_Index + 0x02), on page 156*. The IXF1104 MAC duplex setting must be programmed by the system software to match the attached PHY duplex for proper IXF1104 MAC operation.

### 5.1.5.3    Copper Auto-Negotiation

In the copper MAC, auto-negotiation and all other controls of the PHY devices are achieved through the MDIO interface, and are independent of the MAC controller. See Section 5.5, *MDIO Control and Interface, on page 98* for further operation details.

**Note:**    In copper mode, auto-negotiation is accomplished by the attached PHY, not the IXF1104 MAC. Thus, the IXF1104 MAC does not automatically configure the MAC or other blocks in the device to be consistent with attached PHY auto-negotiation results. This must be accomplished by the user and system software.

## 5.1.6　Jumbo Packet Support

The IXF1104 MAC supports jumbo frames. The jumbo frame length is dependent on the application and the IXF1104 MAC design is optimized for a 9.6 KB jumbo frame length. Larger lengths can be programmed, but limited system performance may lead to data loss during certain flow-control conditions

The value programmed into the *Max Frame Size (Addr: Port_Index + 0x0F)* determines the maximum length frame size the MAC can receive or transmit without activating any error counters, and without truncation.

The *Max Frame Size (Addr: Port_Index + 0x0F)* bits 13:0 set the frame length. The default value programmed into this register is 0x05EE (1518). The value is internally adjusted by +4 if the frame has a VLAN tag. The overall programmable maximum is 0x3FFF or 16383 bytes.

The register should be programmed to 0x2667 for the 9.6 KB length jumbo frame, optimized for the IXF1104 MAC. The RMON counters are also implemented for jumbo frame support as follows:

### 5.1.6.1　Rx Statistics

- RxOctetsTotalOK (Addr: Port_Index + 0x20)
- RxPkts1519toMaxOctets (Addr: Port_Index + 0x2B)
- RxFCSErrors (Addr: Port_Index + 0x2C)
- RxDatatError (Addr: Port_Index = 0x02E)
- RxAlignErrors (Addr: Port_Index + 0x2F)
- RxLongErrors (Addr: Port_Index + 0x30)
- RxJabberErrors (Addr: Port_Index + 0x31)
- RxVeryLongErrors (Addr: Port_Index + 0x34)

### 5.1.6.2　TX Statistics

- OctetsTransmittedOK (Addr: Port_Index + 0x40)
- TxPkts1519toMaxOctets (Addr: Port_Index + 0x4B)
- TxExcessiveLengthDrop (Addr: Port_Index + 0x53)
- TxCRCError (Addr: Port_Index + 0x56)

The IXF1104 MAC checks the CRC for all legal-length jumbo frames (frames between 1519 and the Max Frame Size). On transmission, the MAC can be programmed to append the CRC to the frame or check the CRC and increment the appropriate counter. On reception, the MAC transmits these frames across the SPI3 interface (jumbo frames above the setting in the Section 128, *RX FIFO Transfer Threshold Port 0 ($0x5B8)* with a bad CRC cannot be dropped and are sent across the SPI3 interface). If the receive frame has a bad CRC, the appropriate counter increments and the RxERR flag is asserted on the SPI3 receive interface.

Jumbo frames also impact flow control. The maximum frame size needs to be taken into account when determining the FIFO watermarks. The current transmission must be completed before a Pause frame is transmitted (needed when the receiver FIFO High watermark is exceeded). If the current transmission is a jumbo frame, the delay may be significant and increase data loss due to insufficient available FIFO space.

### 5.1.6.3    Loss-less Flow Control

The IXF1104 MAC supports loss-less flow control when the size of a Jumbo packet is restricted to 9.6 k bytes. If this condition is met, the IXF1104 MAC has sufficient memory resources allocated to each MAC port to ensure that, if both the IXF1104 MAC and link partner are required to send Pause packets simultaneously during jumbo packet transfers across a medium of five kilometers of fiber, no packet data should be lost due to FIFO overflows.

## 5.1.7    Packet Buffer Dimensions

### 5.1.7.1    TX and RX FIFO Operation

#### 5.1.7.1.1    TX FIFO

The IXF1104 MAC TX FIFOs are implemented with 10 KB for each channel. This provides enough space for at least one maximum size (10 KB) packet per-port storage and ensures that no under-run conditions occur, assuming that the sending device can supply data at the required data rate.

A transfer to MAC Threshold parameter, which is user-programmable, determines when the FIFO signals to the MAC that it has data to send. This is configured for specific block sizes, and the user must ensure that an under-run does not occur. Also, the threshold can be set above the maximum size of a normal Ethernet packet. This causes the FIFO to send only data to the MAC when this threshold is exceeded or when the End-of-Packet marker is received. This second condition eliminates the possibility of under-run, except when the controlling switch device fails. It can, however, cause idle times on the media.

#### 5.1.7.1.2    RX FIFO

The IXF1104 MAC RX FIFOs are provisioned so that each port has its own 32 KB of memory space. This is enough memory to ensure that there is never an over-run on any channel while transferring normal Ethernet frame size data.

The FIFOs automatically generate Pause control frames to halt the link partner when the High watermark is reached and to restart the link partner when the data stored in the FIFO falls below the low-watermark. The RX and TX FIFOs have been sized to support lossless flow control with 9.6 KB packets. The RX FIFO has a programmable transfer threshold that sets the threshold at which packets become "cut through" and starts transitioning to the SPI3 interface before the EOP is received. Packets sizes below this threshold are treated as "store and forward." Once a packet size exceeds the RX FIFO transfer threshold, it can no longer be dropped by the RX FIFO even if it is marked to be dropped by the MAC.

## 5.1.8    RMON Statistics Support

The IXF1104 MAC supplies RMON statistics through the CPU interface. These statistics are available in the form of counter values that can be accessed at specific addresses in the register maps (Table 59 through Table 69). Once read, these counters automatically reset and begin counting from zero. A separate set of RMON statistics is available for each MAC device in the IXF1104 MAC.

Implementation of the RMON Statistics block is similar to the functionality provided by existing Cortina switch and router products. This implementation allows the IXF1104 MAC to provide all of the RMON Statistics group as defined by RFC2819. The IXF1104 MAC

supports the RMON RFC2819 Group 1 statistics counters. Table 25 notes the differences and additional statistics registers supported by the IXF1104 MAC that are outside the scope of the RMON RFC2819 document.

**Table 25      RMON Additional Statistics (Sheet 1 of 3)**

| RMON Ethernet Statistics Group 1 Statistics | Type | IXF1104 MAC-Equivalent Statistics | Type | Definition of RMON Versus IXF1104 MAC Documentation |
|---|---|---|---|---|
| etherStatsindex | Integer 32 | NA | NA | NA |
| etherStatsDataSource | Object identifier | NA | NA | NA |
| etherStatsDropEvents | Counter 32 | RX Number of Frames Removed/ TX Number of Frames Removed | Counter 32 | See table note 1 |
| etherStatsOctets | Counter 32 | RxOctetsTotalOK RxOctetsBad OctetsTransmittedOK OctetsTransmittedBad | Counter 32 | The IXF1104 MAC has two counters for receive and transmit that use different naming conventions for the total Octets and Octets Bad. These counters must be combined to meet the RMON definition for this statistic. |
| etherStatsPkts | Counter32 | RxUCPkts/TxUCPkts RxBCPkts/TxBCPkts RxMCPkts/TxMCPkts | Counter 32 | The IXF1104 MAC has three counters for the etherStatsPkts that must be combined to give the total packets as defined by the RMON specification. |
| etherStatsBroadcastPkts | Counter32 | RxBCPkts/TxBCPkts | Counter 32 | Same as RMON specification |
| etherStatsMulticastPkts | Counter32 | RxMCPkts/TxMCPkts | Counter 32 | See table note 2 |

*Note:* The RMON specification requires that this is, "The total number of events where packets were dropped by the probe due to a lack of resources. This number is not necessarily the number of packets dropped; it is the number of times this condition is detected. The *RX FIFO Overflow Frame Drop Counter Ports 0 - 3 ($0x594 – 0x597)* and *TX FIFO Overflow Frame Drop Counter Ports 0 - 3 ($0x621 – 0x624)* in the IXF1104 MAC support this and increment when either an RX FIFO or TX FIFO overflows. If any IXF1104 MAC programmable packet filtering is enabled, the *RX FIFO Errored Frame Drop Counter Ports 0 - 3 ($0x5A2 - 0x5A5)* and *TX FIFO Errored Frame Drop Counter Ports 0 - 3 ($0x625 – 0x629)* increment with every frame removed in addition to the existing frames counted due to FIFO overflow.

**Table 25      RMON Additional Statistics (Sheet 2 of 3)**

| RMON Ethernet Statistics Group 1 Statistics | Type | IXF1104 MAC-Equivalent Statistics | Type | Definition of RMON Versus IXF1104 MAC Documentation |
|---|---|---|---|---|
| etherStatsCRCAlignErrors | Counter32 | RxAlignErrors RxFCSErrors TxCRCError | Counter 32 | The IXF1104 MAC has two counters for the alignment and CRC errors for the RX side only. The IXF1104 MAC has a CRC Error counter for the TX side. |
| etherStatsUndersizedPkts | Counter32 | RxRuntErrors RxShortErrors Rx Statistics ONLY | Counter 32 | The IXF1104 MAC has two counters, one for Runt errors and one for ShortErrors. |
| etherStatsOversizePkts | Counter32 | RxLongErrors TxExcessiveLength Drop | Counter 32 | Same as RMON specification |
| etherStatsFragments | Counter32 | RuntErrors | Counter 32 | Same as RMON specification |
| etherStatsJabbers | Counter32 | JabberErrors | Counter 32 | Same as RMON specification |
| etherStatsCollisions | Counter32 | TxSingleCollision TxMultipleCollision TxLateCollision TxTotalCollision | Counter 32 | The TxTotalCollision count value is equivalent to the RMON specification minus the TxLateCollision |
| etherStatsPkts64Octets | Counter32 | RxPkts64Octets/ TxPkts64Octets | Counter 32 | Same as RMON specification |
| etherStatsPkts65to127Octets | Counter32 | RxPkts65to127Octets/ TxPkts65to127Octets | Counter 32 | Same a RMON specification |
| etherStatsPkts128to255Octets | Counter32 | RxPkts128to255Octets/ TxPkts128to255Octets | Counter32 | Same a RMON specification |
| etherStatsPkts256to511Octets | Counter32 | RxPkts256to511Octets/ TxPkts256to511Octets | Counter32 | Same a RMON specification |
| etherStatsPkts512to1023Octets | Counter32 | RxPkts512to1023Octets/ TxPkts512to1023Octets | Counter32 | Same a RMON specification |

*Note:*   The RMON specification requires that this is, "The total number of events where packets were dropped by the probe due to a lack of resources. This number is not necessarily the number of packets dropped; it is the number of times this condition is detected. The *RX FIFO Overflow Frame Drop Counter Ports 0 - 3 ($0x594 – 0x597)* and *TX FIFO Overflow Frame Drop Counter Ports 0 - 3 ($0x621 – 0x624)* in the IXF1104 MAC support this and increment when either an RX FIFO or TX FIFO overflows. If any IXF1104 MAC programmable packet filtering is enabled, the *RX FIFO Errored Frame Drop Counter Ports 0 - 3 ($0x5A2 - 0x5A5)* and *TX FIFO Errored Frame Drop Counter Ports 0 - 3 ($0x625 – 0x629)* increment with every frame removed in addition to the existing frames counted due to FIFO overflow.

**Table 25     RMON Additional Statistics (Sheet 3 of 3)**

| RMON Ethernet Statistics Group 1 Statistics | Type | IXF1104 MAC-Equivalent Statistics | Type | Definition of RMON Versus IXF1104 MAC Documentation |
|---|---|---|---|---|
| etherStatsPkts1023to1518 Octets | Counter 32 | RxPkts1023to1518Octets/ TxPkts1023to1518Octets | Counter32 | Same as RMON specification |
| etherStatOwner | Owner String | NA | NA | NA |
| etherStatsStatus | Entry Status | NA | NA | NA |

*Note:*   The RMON specification requires that this is, "The total number of events where packets were dropped by the probe due to a lack of resources. This number is not necessarily the number of packets dropped; it is the number of times this condition is detected. The *RX FIFO Overflow Frame Drop Counter Ports 0 - 3 ($0x594 – 0x597)* and *TX FIFO Overflow Frame Drop Counter Ports 0 - 3 ($0x621 – 0x624)* in the IXF1104 MAC support this and increment when either an RX FIFO or TX FIFO overflows. If any IXF1104 MAC programmable packet filtering is enabled, the *RX FIFO Errored Frame Drop Counter Ports 0 - 3 ($0x5A2 - 0x5A5)* and *TX FIFO Errored Frame Drop Counter Ports 0 - 3 ($0x625 – 0x629)* increment with every frame removed in addition to the existing frames counted due to FIFO overflow.

### 5.1.8.1     Conventions

The following conventions are used throughout the RMON Management Information Base (MIB) and its companion documents.

- **Good Packets**: Error-free packets that have a valid frame length. For example, on Ethernet, good packets are error-free packets that are between 64 and 1518 octets long. They follow the form defined in IEEE 802.3, Section 3.2.

- **Bad Packets**: Bad packets are packets that have proper framing and recognized as packets, but contain errors within the packet or have an invalid length. For example, on Ethernet, bad packets have a valid preamble and SFD, but have a bad CRC, or are either shorter than 64 octets or longer than 1518 octets.

### 5.1.8.2     Advantages

The following lists additional IXF1104 MAC registers that support features not documented in RMON:

- MAC (flow) control frames
- VLAN Tagged
- Sequence Errors
- Symbol Errors
- CRC Error

These additional counters allow for differentiation beyond standard RMON probes.

**Note:**     In fiber mode, a packet transfer with an invalid 10-bit symbol does not always update the statistics registers correctly.

- **Behavior:** The IXF1104 MAC 8B10B decoder substitutes a valid code word octet in its place. The packet transfer is aborted and marked as bad. The new internal length of the packet is equal to the byte position where the invalid symbol was. No packet fragments are seen at the next packet transfer.

- **Issue:** If the invalid 10-bit code is inserted in a byte position of 64 or greater, expected RX statistics are reported. However, if the invalid code is inserted in a byte position of less than 64, expected RX statistics are not stored.

## 5.2 SPI3 Interface

The IXF1104 MAC SPI3 Interface is implemented to the System Packet Interface Level 3 (SPI3) Physical Layer Interface standard. The interface function allows the IXF1104 MAC blocks to interface to higher-layer network processors or switch fabric.

The IXF1104 MAC transmit interface allows data flows from a network processor or switch fabric device to the IXF1104 MAC. The receive interface allows data to flow from the IXF1104 MAC to the network processor or switch fabric device.

This interface receives and transmits data between the MAC and the Network Processor with compliant SPI3 interfaces. The SPI3 interface operation is defined in the OIF-SPI3-01.0 (available from the Optical Internet Working Forum [www.oiforum.com]). The OIF specification defines operation for the transfer of data at data rates of up to 3.2 Gbps when operating at a frequency of 104 MHz. The IXF1104 MAC defines operation for the transfer of data at data rates of up to 4.256 Gbps when operating at a maximum frequency of 133 MHz in MPHY mode and 125 MHz in SPHY Mode.

There is no guarantee of the number of bytes available since the size of packets is variable. An IXF1104 MAC port-transmit packet available status is provided on signals DTPA, STPA or PTPA, indicating the TX FIFO is nearly full.

In the receive direction, RVAL indicates if valid data is available on the receive data bus and is defined so that data transfers can be aligned with packet boundaries.

The SPI3 interface supports the following two modes of operation:

- MPHY or 32 bit mode (one 32-bit data bus)
- SPHY or 4 x 8 mode (four individual 8-bit data buses)

### 5.2.1 MPHY Operation

The MPHY operation mode is selected when bit 21 of the Section 146, *SPI3 Transmit and Global Configuration ($0x700)* is set to 0 and bit 7 of the *SPI3 Receive Configuration ($0x701)* is set to 1.

#### 5.2.1.1 Data Path

The IXF1104 MAC SPI3 interface has a single 32-bit data path in the MPHY configuration mode (see Figure 13). The bus interface is point-to-point (one output driving only one input load), so a 32-bit data bus would support only one IXF1104 MAC.

To support variable-length packets, the RMOD[1:0]/TMOD[1:0] signals are defined to specify valid bytes in the 32-bit data bus structure. Each double-word must contain four valid bytes of packet data until the last double-word of the packet transfer, which is marked with the end of packet REOP/TEOP signal. This last double-word of the transfer contains up to four valid bytes specified by the RMOD[1:0]/TMOD[1:0] signals.

The IXF1104 MAC port selection is performed using in-band addressing. In the transmit direction, the network processor device selects an IXF1104 MAC port by sending the address on the TDAT[1:0] bus marked with the TSX signal active and TENB signal inactive. All subsequent TDAT[1:0] bus operations marked with the TSX signal inactive and the TENB active are packet data for the specified port.

In the receive direction, the IXF1104 MAC specifies the selected port by sending the address on the RDAT[1:0] bus marked with the RSX signal active and RVAL signal inactive. All subsequent RDAT[1:0] bus operations marked with RSX inactive and RVAL active are packet data from the specified port.

**Note:** See Table 17, *SPI3 MPHY/SPHY Interface,* on page 60 for a complete list of the MPHY mode signals. The control signals with the port designator for Port 0 are the only ones used in MPHY mode and they apply to all 4 ports. Table 3, *SPI3 Interface Signal Descriptions,* on page 40 provides a comprehensive list of SPI3 signal descriptions.

### 5.2.1.2 SPI3 RX Round Robin Data Transmission

The IXF1104 MAC uses a round-robin protocol to service each of the 4 ports dependent upon the enable status of the port and if there is data available to be taken from the RX FIFO. The round robin order goes from port 0, port 1, port 2, port 3, and back to port 0. A port is skipped and the next port is serviced if it has no available transmit data. The data transfer bursts are user-configurable burst lengths of 64, 128, or 256 bytes. The IXF1104 MAC also has a configurable pause interval between data transfer bursts on the receive side of the interface. The RX SPI3 burst lengths and the pause interval can be set in the *SPI3 Receive Configuration ($0x701)*).

## 5.2.2 MPHY Logical Timing

The SPI3 interface AC timing for MPHY can be found in Section 7.2, *SPI3 AC Timing Specifications,* on page 133. Logical timing in the following diagrams illustrates all signals associated with MPHY mode.

### 5.2.2.1 Transmit Timing

In MPHY mode a packet transmission starts with the TSX signal indicating port address information is on the data bus. The next clock cycle TENB and TSOP indicate present data on the bus is the first word in the packet and all subsequent clocks will contain valid data as long as TENB is active or until TEOP is asserted. Data transmission can be temporally halted when TENB goes high then resumed when TENB is low. The valid bytes in the final word, during an active TEOP, are indicated by state of TMOD [1:0].

**Figure 11     MPHY Transmit Logical Timing**



1.  Applies to all transmit packet available signals (STPA, PTPA, DTPA_0:3).
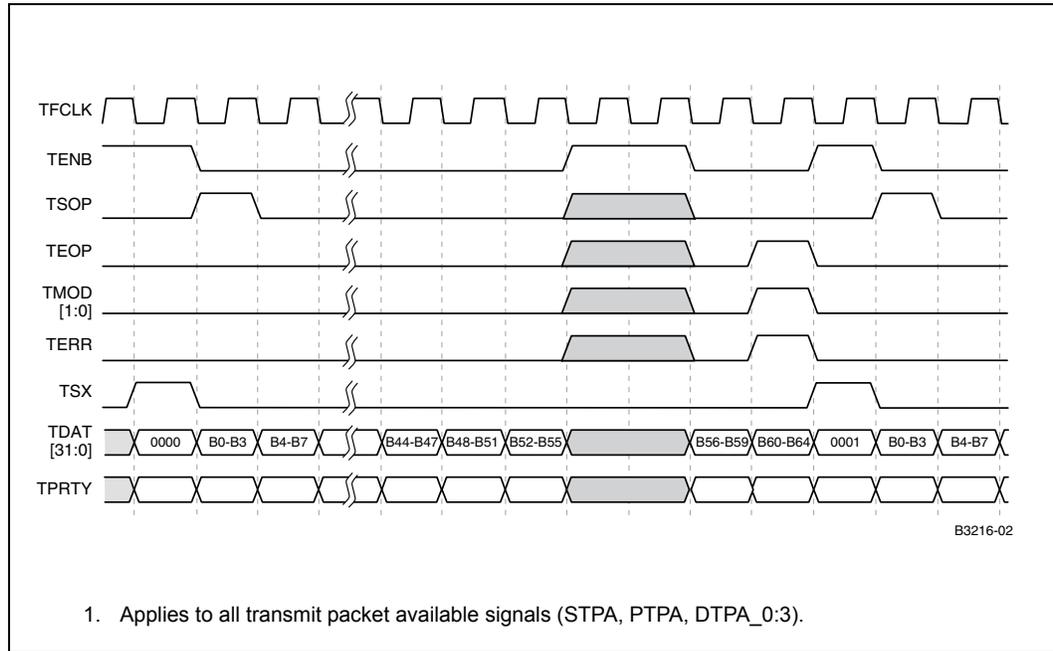
## 5.2.2.2     Receive Timing

A packet is received when RSX indicates port address information on the data bus followed by RSOP to indicate the data bus contains the first word of a packet. All subsequent data is valid only while RVAL is High and until REOP is asserted. Receive data can be temporarily halted when RENB is de-asserted and starts again on the second rising edge of RFCLK following the assertion of RENB. RMOD indicates the number of valid bytes in the last transfer when REOP is asserted.

**Figure 12     MPHY Receive Logical Timing**
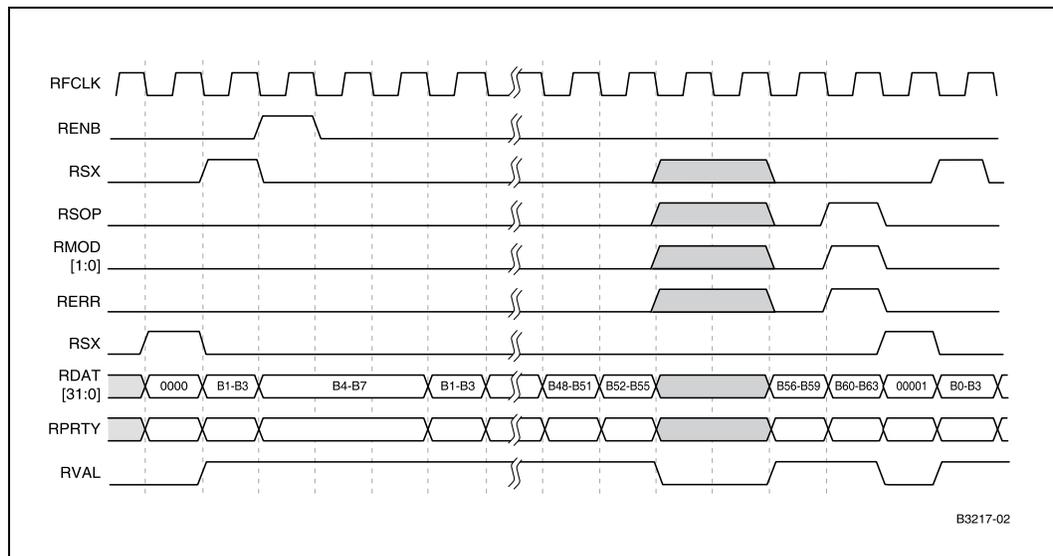
**Figure 13    MPHY 32-Bit Interface**
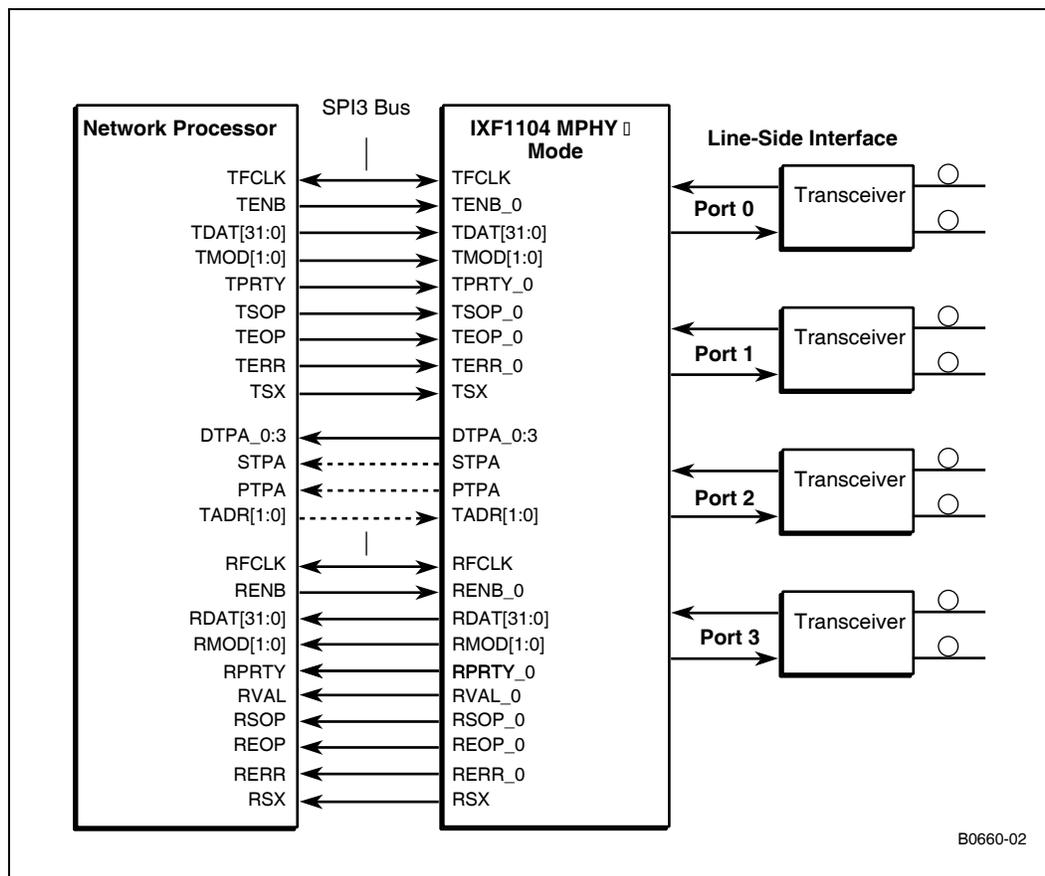


### 5.2.2.3    Clock Rates

In MPHY mode, the TFCLK and RFCLK can be independent of each other. TFCLK and RFCLK should be common to the IXF1104 MAC and the Network Processor. The IXF1104 MAC requires a single clock source for the transmit path and a single clock source for the receive path.

To allow all four IXF1104 MAC ports to operate at 1 Gbps, the IXF1104 MAC is designed to allow this interface to be overclocked. This allows operation for data transfer at data rates of up to 4.256 Gbps when operating at an overclocked frequency of 133 MHz.

**Note:**    MPHY mode operates at a maximum clock frequency of 133 MHz (TFCLK and RFCLK).

### 5.2.2.4    Parity

The IXF1104 MAC can be odd or even (the IXF1104 MAC is odd by default) when calculating parity on the data bus. This can be changed to accommodate even parity if desired, and can be set for transmit and receive independently. The RX Parity is set in bit 12 of the Section 147, *SPI3 Receive Configuration ($0x701)* and the TX Parity is set in bit 4 of the Section 146, *SPI3 Transmit and Global Configuration ($0x700)*.

### 5.2.2.5 SPHY Mode

The SPHY operation mode is selected when bit 21 of the Table 146, *SPI3 Transmit and Global Configuration ($0x700), on page 204* is set to 1. The SPHY mode is the default operation for the IXF1104 MAC SPI3 interface.

#### 5.2.2.5.1 Data Path

The IXF1104 MAC SPI3 interface has four 8-bit data paths that can support four independent 8-bit point-to-point connections in SPHY mode (see Figure 16). Since each MAC port has its own dedicated 8-bit SPI3 data bus, each port has it own status signal (unlike MPHY). See the For a detailed list of all the signals refer to the SPI3 pin multiplexing table....

Furthermore since each port has it own dedicated bus the in band port addressing is not needed. The 8 bit data bus eliminates the need to have separate control signals determine the number of valid bytes on an EOP.Therefore TSX, RSX, TMOD[1:0] RMOD[1:0] are not used in SPHY mode.

**Note:** See Table 17, *SPI3 MPHY/SPHY Interface, on page 60* for a complete list of the SPHY mode signals. Unlike MPHY mode, each port has a dedicated control signal associated with each of the per-port 8-bit data buses. Table 3, *SPI3 Interface Signal Descriptions, on page 40* provides signal descriptions for all SPI3 signals.

#### 5.2.2.5.2 Receive Data Transmission

Packets are transmitted on each port as they become available from the RX FIFO. The burst length is determined by the setting of per port burst size and the B2B pause settings in the *SPI3 Receive Configuration ($0x701)*. If the B2B pause setting is zero pause cycles inserted, then the entire packet will be burst without any pauses unless the Network Processor de-asserts RENB. If the B2B_Pause setting calls for the insertion of two pause cycles on a port, these are inserted after each data burst for that port. The data bursts are user configurable for each port in the *SPI3 Receive Configuration ($0x701)*.

### 5.2.2.6 SPHY Logical Timing

SPI3 interface AC timing for SPHY can be found in Section 7.2, *SPI3 AC Timing Specifications, on page 133*. Logical timing in the following diagrams illustrates all signals associated with SPHY mode. SPHY mode is similar to MPHY mode except the following signals are not used:

- TMOD[1:0]
- RMOD[1:0]
- TSX
- RSX
- Address Data appearing on the data bus

### 5.2.2.7 Transmit Timing (SPHY)

Packet transmission starts when TENB and TSOP indicate present data on the bus is the first word in the packet. All subsequent clocks will contain valid data as long as TENB is active or until TEOP is asserted. Data transmission can be temporally halted when TENB goes high then resumed when TENB is low.

**Figure 14    SPHY Transmit Logical Timing**
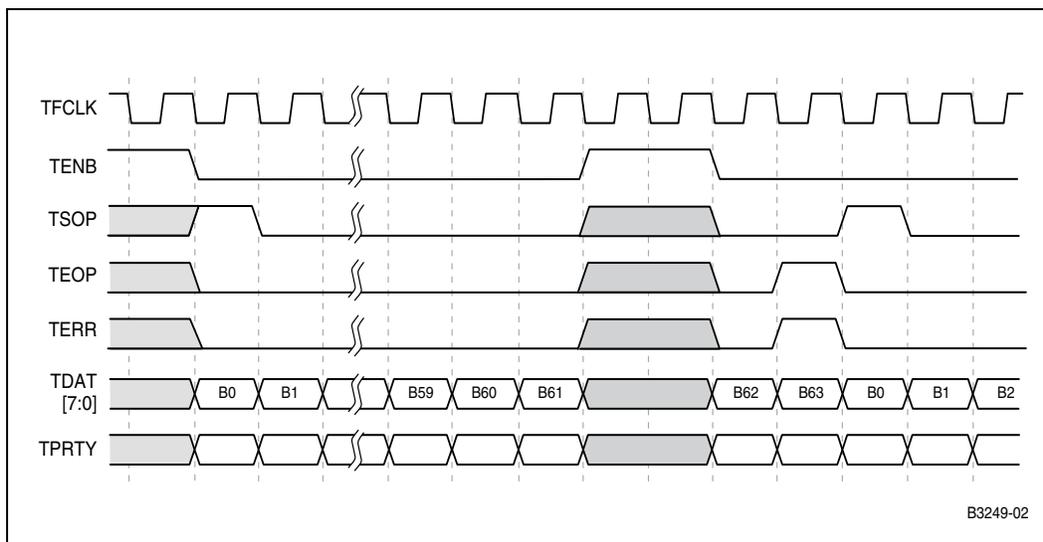


### 5.2.2.8    Receive Timing (SPHY)

A packet is received when RSOP is asserted to indicate the data bus contains the first word of the packet. All subsequent data is valid only while RVAL is high and until REOP is asserted. Receive data can be temporarily halted when RENB is de-asserted and starts again on the second rising edge of RFCLK following the assertion of RENB. When REOP is asserted RMOD indicates the number of valid bytes in the last transfer.

**Figure 15    SPHY Receive Logical Timing**
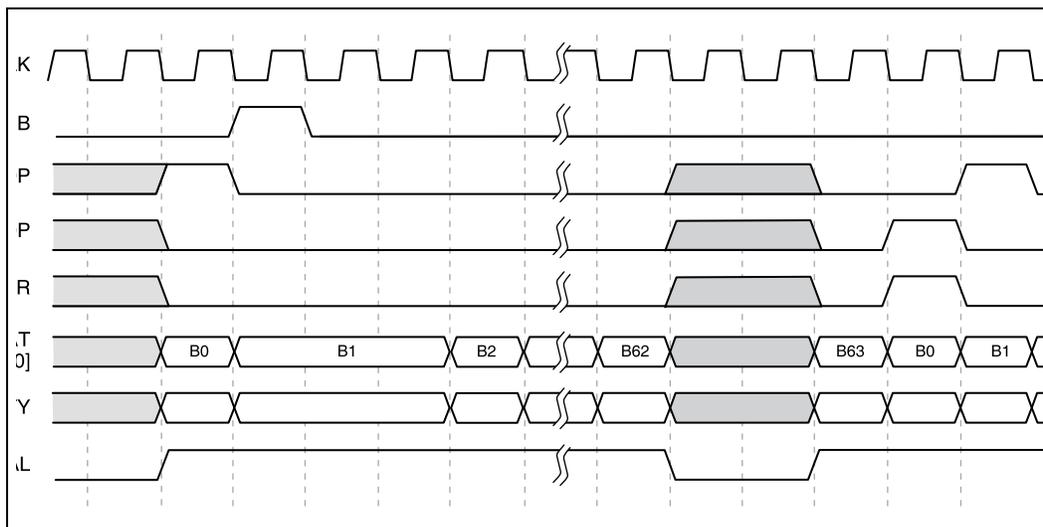
**Figure 16    SPHY Connection for Two IXF1104 MAC Ports (8-Bit Interface)**
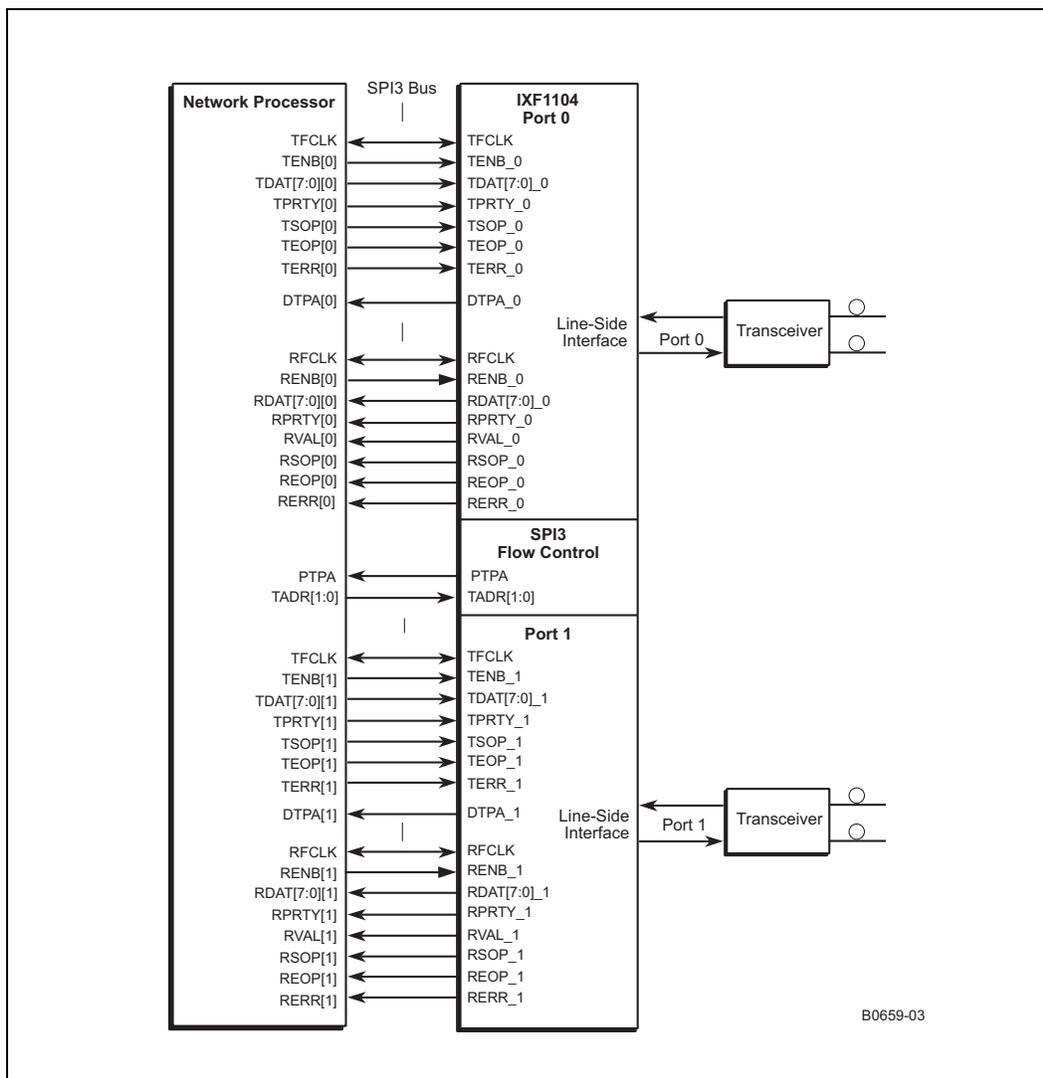


#### 5.2.2.8.1    Clock Rates

The TFCLK and RFCLK can be independent of each other in SPHY mode operation. TFCLK and RFCLK should be common to all the Network Processor devices. The IXF1104 MAC requires an individual single clock source for the device transmit path and a single clock source for the device receive path.

The IXF1104 MAC allows this interface to be overclocked so that all four IXF1104 MAC ports can operate at 1 Gbps. This allows data transfer at data rates of up to 4.0 Gbps when operating at an overclocked frequency of 125 MHz.

**Note:**    SPHY operates at a maximum frequency of 125 MHz.

#### 5.2.2.8.2   Parity

The IXF1104 MAC can be odd or even (the IXF1104 MAC defaults to odd) when calculating parity on the data bus. This can be changed to accommodate even parity if desired, and can be set for transmit and receive ports independently. The RX and TX parity sense bits have a direct relationship to the port parity in SPHY mode.

The per port RX parity is set in the *SPI3 Receive Configuration ($0x701)* and the per port TX Parity is set in the *SPI3 Transmit and Global Configuration ($0x700)*.

### 5.2.2.9   SPI3 Flow Control

The SPI3 packet interface supports transmit and receive data transfers at clock rates independent of the line bit rate. As a result, the IXF1104 MAC supports packet rate decoupling using internal FIFOs. These FIFOs are 10 KB per port in the transmit direction (egress from the IXF1104 MAC to the line interfaces) and 32 KB per port in the receive direction (ingress to the IXF1104 MAC from the line interfaces).

Control signals are provided to the network processor and the IXF1104 MAC to allow either one to exercise flow control. Since the bus interface is point-to-point, the receive interface of the
IXF1104 MAC pushes data to the link-layer device. For the transmit interface, the packet available status granularity is byte-based.

#### 5.2.2.9.1   RX SPI3 Flow Control

In the receive direction, when the IXF1104 MAC has stored an end-of-packet (a complete small packet or the end of a larger packet) or some predefined number of bytes in its receive FIFO, it sends the in-band address followed by FIFO data to the link-layer device (in MPHY mode). The data on the interface bus is marked with the valid signal (RVAL) asserted. The network processor device can pause the data flow by de-asserting the Receive Read Enable (RENB) signal.

**RENB_0:3**

RENB_0:3 controls the flow of data from the IXF1104 MAC RX FIFOs. In SPHY mode, there is a dedicated RENB for each port. In MPHY mode, RENB_0 is used as the global signal covering all ports. When RENB is sampled Low, the network processor can accept data. A read is performed from the RX FIFO and the RDAT, RPRTY, RMOD[1:0], RSOP, REOP, RERR, RSX, and RVAL signals are updated on the following rising edge of RFCLK.

RENB can be asserted High by the Network Processor at any time if it is unable to accept any more data. When the RENB is sampled High by the IXF1104 MAC, a read of the RX FIFO is not performed, and the RDAT, RPRTY, RMOD[1:0], RSOP, REOP, RERR, RSX and RVAL signals remain unchanged on the following rising edge of RFCLK.

#### 5.2.2.9.2   TX SPI3 Flow Control

In the transmit direction, when the IXF1104 MAC has space for some predefined number of bytes in its transmit FIFO, it informs the Network Processor device by asserting one of the Transmit Packet Available (TPA) signals. The Network Processor device writes the in-band address followed by packet data to the IXF1104 MAC using an enable signal (TENB). The network processor device monitors the TPA signals for a High-to-Low transition, which indicates that the transmit FIFO is almost full (the number of bytes left in the FIFO is user-selectable by setting the *TX FIFO High Watermark Ports 0 - 3 ($0x600 – 0x603)*, and suspends data transfer to avoid an overflow. The Network Processor device can pause the data flow by de-asserting the enable signal (TENB).

The IXF1104 MAC provides the following three types of TPA signals:

- Dedicated per port Direct Transmit Packet Available (DTPA)
- Selected-PHY Transmit Packet Available (STPA), which is based on the current in-band port address in MPHY mode.
- Polled-PHY Transmit packet Available (PTPA), which provides FIFO information on the port selected by the TADR[1:0] signals.

The following three TPA signals (DTPA_0:3, STPA, and PTPA) provide flow control based on the programmable TX FIFO High and Low watermarks. Refer to Table 132, *TX FIFO High Watermark Ports 0 - 3 ($0x600 – 0x603), on page 195* and Table 133, *TX FIFO Low Watermark Register Ports 0 - 3 ($0x60A – 0x60D), on page 196* for more information.

### DTPA_0:3:

A direct status indication for the TX FIFOs of ports [0:3]. When DTPA is High, it indicates the amount of data in the TX FIFO is below the TX FIFO High watermark. When the High watermark is crossed, DTPA transitions Low to indicate the TX FIFO is almost full. It stays low until the amount data in the TX FIFO goes back below the TX FIFO Low watermark. At this point, DTPA transitions High to indicate the programmed number of bytes are now available for data transfers.

DTPA_0:3 is updated on the rising edge of the TFCLK.

### STPA:

STPA provides TX FIFO status for the currently selected port in MPHY mode. When High, STPA indicates that the amount of data in the TX FIFO for the port selected, specified by the latest in-band address, is below the TX FIFO High watermark. When the High watermark is crossed, STPA transitions Low to indicate the TX FIFO is almost full. It stays Low until the amount of data in the TX FIFO goes back below the TX FIFO Low watermark. At this point, STPA transitions High to indicate the programmed number of bytes are now available for data transfers.

The port reported by STPA is updated on the rising edge of TFCLK after TSX is sampled as asserted. STPA is updated on the rising edge of TFCLK.

**Note:** STPA is only used when the IXF1104 MAC is configured for MPHY mode of operation.

### PTPA:

PTPA provides status of the TX FIFO based on the port selected by the TADR[1:0] address bus.

When High, PTPA indicates that the amount of data in the TX FIFO for the port selected is below the TX FIFO High watermark. When the High watermark is crossed, PTPA transitions Low to indicate the TX FIFO is almost full. It stays Low until the amount of data in the TX FIFO goes back below the TX FIFO Low watermark. PTPA then transitions High to indicate the programmed number of bytes are now available for data transfers.

The port reported by PTPA is updated on the rising edge of TFCLK after the TADR{1:0] port address is sampled.

PTPA is updated on the rising edge of TFCLK.

### 5.2.3      Pre-Pending Function

The IXF1104 MAC implements a pre-pending feature to allow 1518-byte Ethernet packets to be pre-padded with two additional bytes of data so that the packet becomes low-word aligned. The 2-byte pre-pend value is all zeros and is inserted before the destination address of the packet being pre-pended. This value is fixed and cannot be changed.

This function is enabled by writing the appropriate data to the *RX FIFO Padding and CRC Strip Enable ($0x5B3)* for each port.

A standard 1518-byte Ethernet packet occupies 379 long words (four bytes) with two additional bytes left over (1518/4 = 379.5). To eliminate the memory-management problems for a network processor or switch fabric, the two remaining bytes are dealt with by the addition of two bytes to the start of a packet. This results in a standard 1518-byte Ethernet packet received by the IXF1104 MAC being forwarded to the higher-layer device as a 380-long-word packet. The upper-layer device is responsible for stripping the additional two bytes.

This feature was added to the IXF1104 MAC to assist in the design of higher-layer memory management. The addition of the two extra bytes is not the default operation of the IXF1104 MAC and must be enabled by the user. The default operation of the IXF1104 MAC SPI3 receive interface forwards data exactly as it is received by the IXF1104 MAC line interface.

## 5.3      Gigabit Media Independent Interface (GMII)

The IXF1104 MAC supports a subset of the GMII interface standard as defined in IEEE 802.3 2000 Edition for 1 Gbps operation only. This subset is limited to operation at 1000 Mbps full-duplex.

The GMII Interface operates as a source synchronous interface only and does not accept a TXC clock provided by a PHY device when operating at 10/100 Mbps speeds.

**Note:**    The RGMII interface must be used for applications that require 10/100/1000 Mbps operation.

The IXF1104 MAC does NOT support 10/100 Mbps copper PHY devices that are implemented using the MII Interface.

**Note:**    MII operation is not supported by the IXF1104 MAC.

The user can select GMII, RGMII, or Optical Module/SerDes functionality on a per-port basis. This mode of operation is controlled through a configuration register.

While IEEE 802.3 specifies 3.3 V operation of GMII devices, most PHYs use 2.5 V signaling. The IXF1104 MAC provides a 2.5 V drive and is 3.3 V-tolerant on inputs.

**Figure 17    MAC GMII Interconnect**
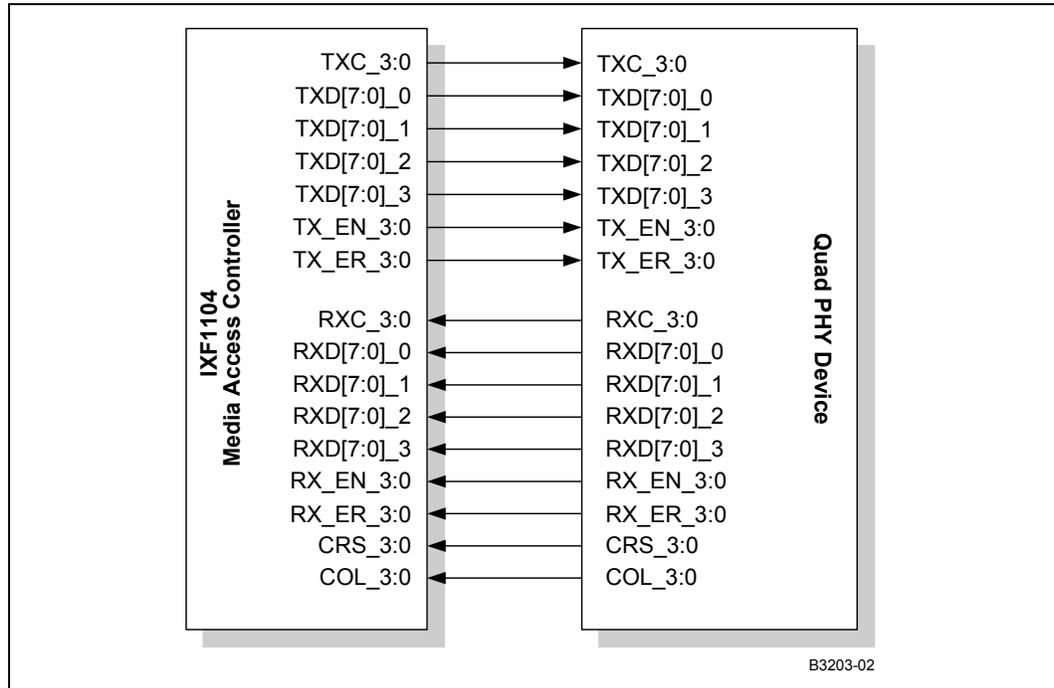


## 5.3.1    GMII Signal Multiplexing

The GMII balls are reassigned when using the RGMII mode or fiber mode. Table 16, *Line Side Interface Multiplexed Balls*, on page 59 specifies the multiplexing of GMII balls in these modes. See Section 5.1.3, *Mixed-Mode Operation*, on page 74 for proper configuration of the IXF1104 MAC in GMII mode.

## 5.3.2    GMII Interface Signal Definition

Table 26, *GMII Interface Signal Definitions*, on page 94 provides the GMII interface signal definitions. For information on 1000BASE-T GMII transmit and receive timing diagrams and tables, please refer to Table 49, *GMII 1000BASE-T Transmit Signal Parameters*, on page 138, Figure 37, *1000BASE-T Transmit Interface Timing*, on page 138, Figure 38, *1000BASE-T Receive Interface Timing*, on page 138, and Table 50, *GMII 1000BASE-T Receive Signal Parameters*, on page 139

**Table 26** **GMII Interface Signal Definitions**

| IXF1104 MAC Signal | GMII Standard Signal | Source | Description |
|---|---|---|---|
| TXC_0<br>TXC_1<br>TXC_2<br>TXC_3 | GTX_CLK | IXF1104 MAC | **Transmit Reference Clock:**<br>125 MHz for Gigabit operation.<br>MII operation for 10/100 Mbps operation is not supported. |
| TXD[7:0]_0<br>TXD[7:0]_1<br>TXD[7:0]_2<br>TXD[7:0]_3 | TXD[7:0] | IXF1104 MAC | **Transmit Data Bus:**<br>Width of this synchronous output bus varies with the speed/mode of operation. In 1000 Mbps mode, all 8 bits are used. |
| TX_EN_0<br>TX_EN_1<br>TX_EN_2<br>TX_EN_3 | TX_EN | IXF1104 MAC | **Transmit Enable:**<br>Synchronous input that indicates Valid data is being driven on the TXD[7:0] data bus. |
| TX_ER_0<br>TX_ER_1<br>TX_ER_2<br>TX_ER_3 | TX_ER | IXF1104 MAC | **Transmit Error:**<br>Synchronous input to PHY causes the transmission of error symbols in 1000 Mbps links. |
| RXC_0<br>RXC_1<br>RXC_2<br>RXC_3 | RX_CLK | PHY | **Receive Clock:**<br>Continuous reference clock is 125 MHz +/– 100 ppm. |
| RXD[7:0]_0<br>RXD[7:0]_1<br>RXD[7:0]_2<br>RXD[7:0]_3 | RXD<3:0> | PHY | **Receive Data Bus:**<br>Width of the bus varies with the speed and mode of operation. In 1000 Mbps mode, all 8 bits are driven by the PHY device.<br>**Note:** MII operation at 10/100 Mbps is not supported. |
| RX_DV_0<br>RX_DV_1<br>RX_DV_2<br>RX_DV_3 | RX_DV | PHY | **Receive Data Valid:**<br>This signal is asserted when valid data is present on the corresponding RXD bus. |
| RX_ER_0<br>RX_ER_1<br>RX_ER_2<br>RX_ER_3 | RX_ER | PHY | **Receive Error:**<br>In 1000 Mbps mode, asserted when error symbols or carrier extension symbols are received.<br>Always synchronous to RX_CLK. |
| CRS_0<br>CRS_1<br>CRS_2<br>CRS_3 | CRS | PHY | **Carrier Sense:**<br>Asserted when valid activity is detected at the line-side interface. |
| COL_0<br>COL_1<br>COL_2<br>COL_3 | COL | PHY | **Collision:**<br>Asserted when a collision is detected and remains asserted for the duration of the collision event. In full-duplex mode, the PHY should force this signal Low. |

## 5.4 Reduced Gigabit Media Independent Interface (RGMII)

The IXF1104 MAC supports the RGMII interface standard as defined in the RGMII Version 1.2 specification. The RGMII interface is an alternative to the IEEE 802.3u MII interface.

The RGMII interface is intended as an alternative to the IEEE 802.3u MII and the IEEE 802.3z GMII. The principle objective of the RGMII is to reduce the number of balls (from a maximum of 28 balls to 12 balls) required to interconnect the MAC and the PHY. This reduction is both cost-effective and technology-independent. To accomplish this objective, the data paths and all associated control signals are reduced, control signals are multiplexed together, and both edges of the clock are used.

- 1000 Mbps operation – clocks operate at 125 MHz
- 100 Mbps operation – clocks operate at 25 MHz
- 10 Mbps operation – clocks operate at 2.5 MHz.

**Note:** The IXF1104 MAC RGMII interface is multiplexed with signals from the GMII interface. See Table 16, *Line Side Interface Multiplexed Balls*, on page 59 for detailed information.

**Figure 18    RGMII Interface**
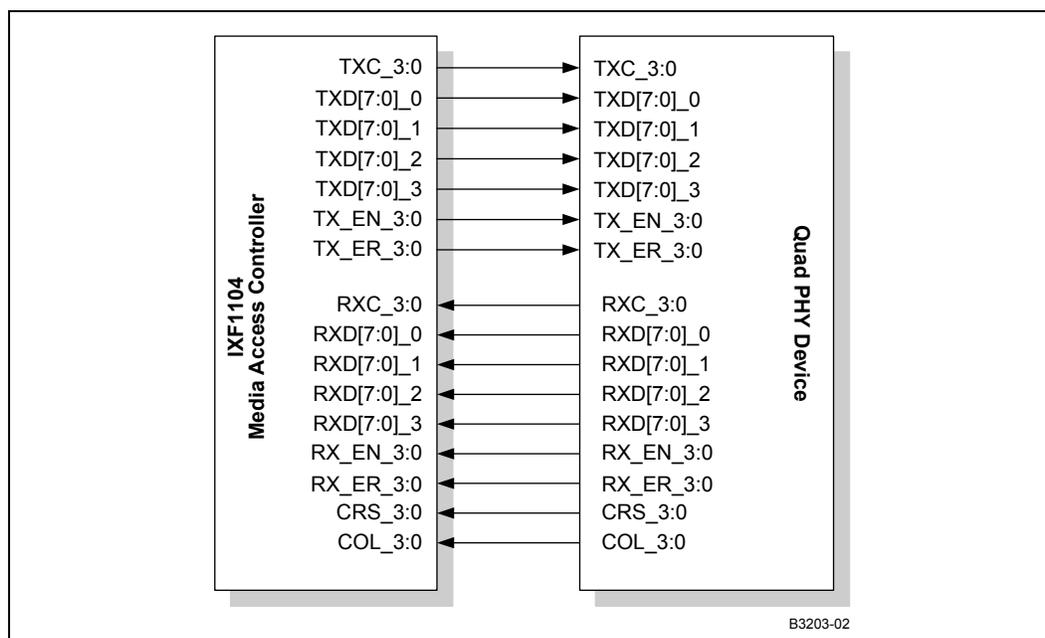


## 5.4.1 Multiplexing of Data and Control

Multiplexing of data and control information is achieved by utilizing both edges of the reference clocks and sending the lower four bits on the rising edge and the upper four bits on the falling edge. Control signals are multiplexed into a single clock cycle using the same technique. For further information on timing parameters, see Figure 36, *RGMII Interface Timing*, on page 137 and Table 48, *RGMII Interface Timing Parameters*, on page 137.

## 5.4.2 Timing Specifics

The IXF1104 MAC RGMII complies with RGMII Rev1.2a requirements. Table 27 provides the timing specifics.

## 5.4.3 TX_ER and RX_ER Coding

To reduce interface power, the transmit error condition (TX_ER) and the receive error condition (RX_ER) are encoded on the RGMII interface to minimize transitions during normal network operation (refer to Table 28 on page 96 for the encoding method). Table 27 provides signal definitions for RGMII.

**Table 27    RGMII Signal Definitions**

| IXF1104 MAC Signal | RGMII Standard Signal | Source | Description |
| --- | --- | --- | --- |
| TXC_0:3 | TXC | MAC | Depending on speed, the transmit reference clock is 125 MHz, 25 MHz, or 2.5 MHz +/– 50ppm. |
| TD[3:0]_n | TD<3:0> | MAC | Contains register bits 3:0 on the rising edge of TXC and register bits 7:4 on the falling edge of TXC. |
| TX_EN | TX_CTL | MAC | TXEN is on the leading edge of TXC.<br>TX_EN xor TX_ER is on the falling edge of TXC. |
| RXC_0:3 | RXC | PHY | Continuous reference clock is 125 MHz, 25 MHz, or 2.5 MHz +/– 50 ppm. |
| RD[3:0]_n | RD<3:0> | PHY | Contains register bits 3:0 on the leading edge of RXC and register bits 7:4 on the trailing edge of RXC. |
| RX_DV | RX_CTL | PHY | RX_DV is on the leading edge of RXC.<br>RX_DV or RXERR is the falling edge of RXC. |

The value of RGMII_TX_ER and RGMII_TX_EN are valid at the rising edge of the clock while TX_ER is presented on the falling edge of the clock. RX_ER coding behaves in the same way (see Table 28, Figure 19, and Figure 20).

**Table 28    TX_ER and RX_ER Coding Description**

| Condition | Description | |
| --- | --- | --- |
| Receiving valid frame, no errors | RX_DV = true<br>Logic High on rising edge of RXC | RX_ER = false<br>Logic High on the falling edge of RXC |
| Receiving valid frame, with errors | RX_DV = true<br>Logic High on rising edge of RXC | RX_ER = true<br>Logic Low on the falling edge of RXC |
| Receiving invalid frame (or no frame) | RX_DV = false<br>Logic Low on rising edge of RXC | RX_ER = false<br>Logic Low on the falling edge of RXC |
| Transmitting valid frame, no errors | TX_EN = true<br>Logic High on rising edge of TXC | TX_ER =false<br>Logic High on the falling edge of TXC |
| Transmitting valid frame with errors | TX_EN = true<br>Logic High on rising edge of TXC | TX_ER = true<br>Logic Low on the falling edge of TXC |
| Transmitting invalid frame (or no frame) | TX_EN = false<br>Logic Low on rising edge of TXC | TX_ER = false<br>Logic low on the falling edge of TXC |
| **Note:**    Refer to Figure 19 for TX_CTL behavior, and Figure 20 for RX_CTL behavior. | | |

**Figure 19    TX_CTL Behavior**
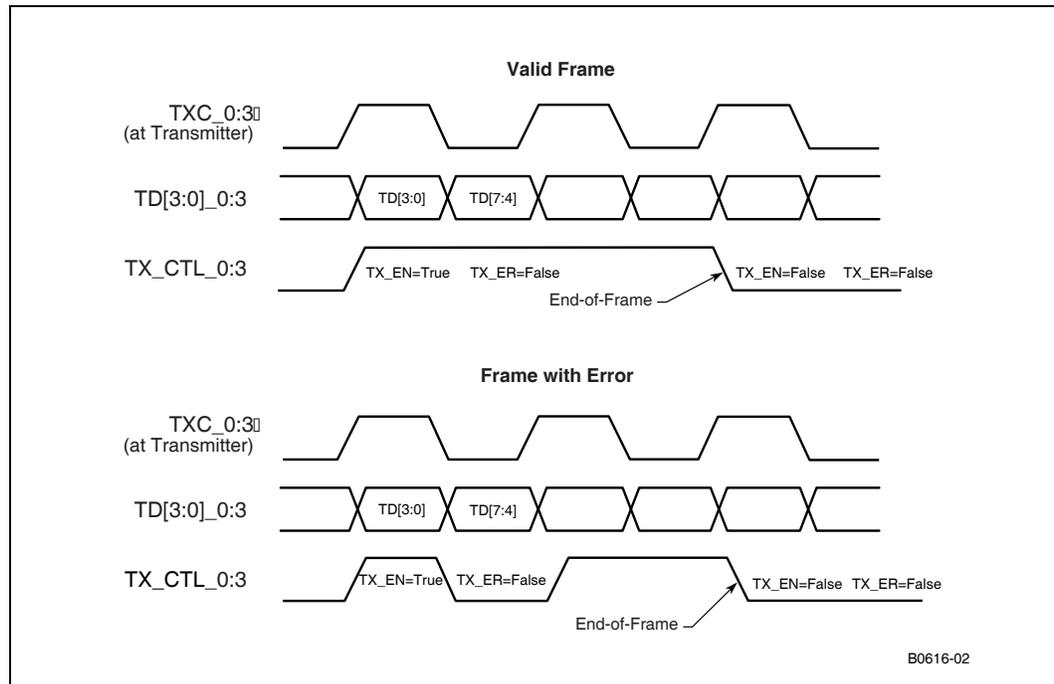


**Figure 20    RX_CTL Behavior**
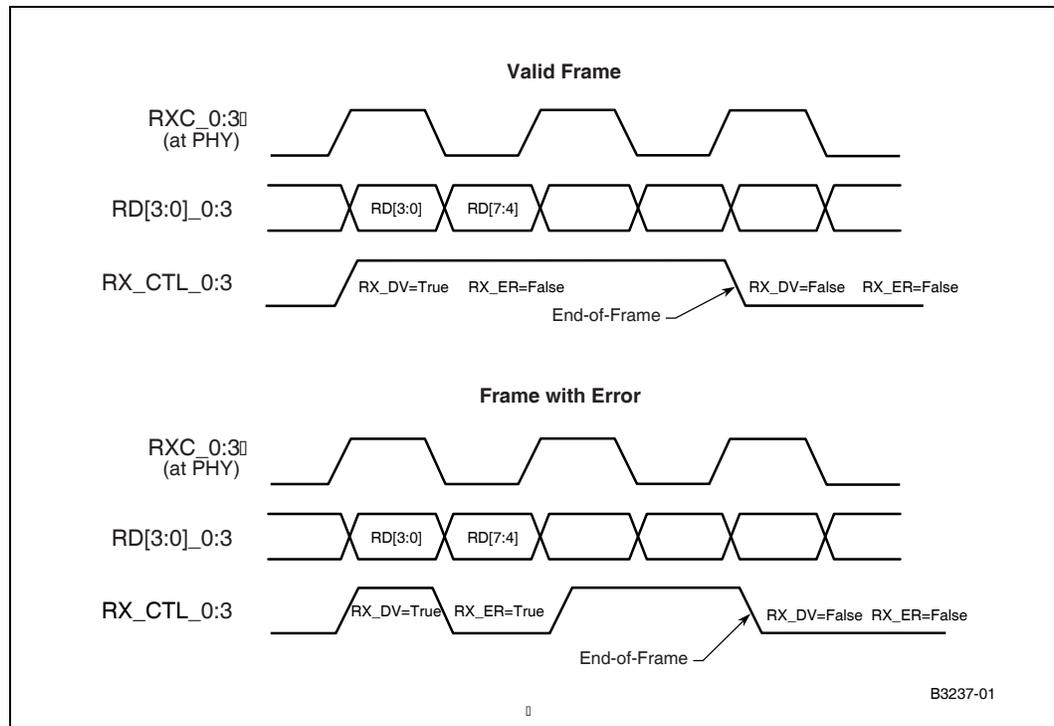
### 5.4.3.1 In-Band Status

Carrier Sense (CRS) is generated by the PHY when a packet is received from the network interface. CRS is indicated when:

- RXDV = true.

- RXDV = false, RXERR = true, and a value of FF exists on the RXD[7:0] bits simultaneously.

- Carrier Extend, Carrier Extend Error, or False Carrier occurs (please reference the Hewlett-Packard* Version 1.2a RGMII Specification for details.).

Carrier Extend and Carrier Extend Error are applicable to Gigabit speeds only. Collision is determined at the MAC by the assertion of TXEN being true while either CRS or RXDV are true. The PHY will not assert CRS as a result of TXEN being true.

## 5.4.4 10/100 Mbps Functionality

The RGMII interface implements the 10/100 Mbps Ethernet Media Independent Interface (MII) by reducing the clock rate to 25 MHz for 100 Mbps operation and 2.5 MHz for 10 Mbps. The TXC is generated by the MAC and the RXC is generated by the PHY. During packet reception, the RXC is stretched on either the positive or negative pulse to accommodate transition from the free-running clock to a data-synchronous clock domain. When the speed of the PHY changes, a similar stretching of the positive or negative pulses is allowed. No glitching of the clocks is allowed during speed transitions.

This interface operates at 10 Mbps and 100 Mbps speeds in the same manner as 1000 Mbps speed, although the data may be duplicated on the falling edge of the appropriate clock. The MAC holds TX_CTL Low until it is operating at the same speed as the PHY.

**Note:** The IXF1104 MAC does not support 10/100 Mbps operation when configured in GMII mode

## 5.5 MDIO Control and Interface

The IXF1104 MAC supports the IEEE 802.3 MII Management Interface, also known as the Management Data Input/Output (MDIO) Interface. This interface allows the IXF1104 MAC to monitor and control each of the PHY devices that are connected to the four ports of IXF1104 MAC when those ports are in copper mode.

The MDIO Master Interface block is implemented once in the IXF1104 MAC. The MDIO Interface block contains the logic through which the user accesses the registers in PHY devices connected to the MDIO/MDC interface, which is controlled by each port.

The MDIO Master Interface block supports the management frame format, specified by IEEE 802.3, clause 22.2.4.5. This block also supports single MDI access through the CPU interface and an autoscan mode. Autoscan allows the IXF1104 MAC MDIO master to read all 32 registers of the per-port copper PHYs and store the contents in the IXF1104 MAC. This provides external-CPU-ready access to the PHY register contents through a single CPU read without the latency of waiting on the low-speed serial MDIO data bus for each register access.

Scan of a single register with low-frequency operation takes approximately 25.6 µs. Scan of a 32-register block takes approximately 820 µs, or 3.3 ms for all four ports. Autoscan data is not valid until approximately 19.2 µs after enabling scan. These numbers scale by 7/50 for high-frequency operation.

### 5.5.1   MDIO Address

The 5-bit PHY address for the MDIO transactions can be set in the *MDIO Single Command ($0x680)*. Bits 5:2 of the PHY address are fixed to a value of 0. Bits 1 and 0 are programmable in bits 9 and 8 of *MDIO Single Command ($0x680)*.

### 5.5.2   MDIO Register Descriptions

For complete information on the MDI registers, refer to the Table 142, *MDIO Single Command ($0x680)*, on page 202, Table 143, *MDIO Single Read and Write Data ($0x681)*, on page 203, Table 144, *Autoscan PHY Address Enable ($0x682)*, on page 203, and Table 145, *MDIO Control ($0x683)*, on page 204.

### 5.5.3   Clear When Done

The MDI Command register bit, in the *MDIO Single Command ($0x680)*, clears upon command completion and is set by the user to start the requested single MDIO Read or Write operation. This bit is cleared automatically upon operation completion.

### 5.5.4   MDC Generation

The MDC clock is used for the MDIO/MDC interface. The frequency of the MDC clock is selectable by setting bit 0, MDC Speed, in an IXF1104 MAC configuration register (see Table 145, *MDIO Control ($0x683)*, on page 204).

### 5.5.4.1   MDC High-Frequency Operation

The high-frequency MDC is 18 MHz, derived from the 125-MHz system clock by dividing the frequency by 7.

The duty cycle is as follows:

- MDC High duration: 3 x (1/125 MHz) = 3 x 8 ns = 24 ns
- MDC Low duration: 4 x (1/125 MHz) = 4 x 8 ns = 32 ns
- MDC runs continuously after reset

Refer to Figure 40, *MDC High-Speed Operation Timing*, on page 140 for the high-frequency MDC timing diagram.

### 5.5.4.2   MDC Low-Frequency Operation

The low-frequency MDC is 2.5 MHz, which is derived from the 125-MHz system clock by dividing the frequency by 50.

The duty cycle is as follows:

- MDC High duration: 25 x (1/125 MHz) = 25 x 8 ns = 200 ns
- MDC Low duration: 25 x (1/125 MHz) = 25 x 8 ns = 200 ns
- MDC runs continuously after reset

Refer to Figure 41, *MDC Low-Speed Operation Timing*, on page 140 for the low frequency MDC timing diagram.

## 5.5.5 Management Frames

The Management Interface serializes the external register access information into the format specified by IEEE 802.3, Section 22.2.4.5 (see Figure 21).

**Figure 21   Management Frame Structure (Single-Frame Format)**

| Preamble 32 Bits | Start 2 Bits | Op Code 2 Bits | PHY Addr 5 Bits | REG Addr 5 Bits | Turnaround 2 Bits | Data 16 Bits |
|---|---|---|---|---|---|---|

First Bit Transmitted                    Last Bit Transmitted

## 5.5.6 Single MDI Command Operation

The Management Data Interface is accessed through the *MDIO Single Command ($0x680)* and the *MDIO Single Read and Write Data ($0x681)*. A single management frame is sent by setting Register 0, bit 20 to logic 1, and is automatically cleared when the frame is completed.

The Write data is first set up in Register 1, bits 15:0 for Write operation. Register 0 is initialized with the appropriate control information (start, op code, etc.) and Register 0, bit 20 is set to logic 1. Register 0, bit 20 is reset to logic 0 when the frame is complete.

The steps are identical for Read operation except that in Register 1, bits 15:0, the data is ignored. The data received from the MDIO is read by the CPU interface from Register 1, bits 31:16.

## 5.5.7 MDI State Machine

The MDI State Machine sequences the information sent to it by the MDIO control registers and keeps track of the current sequence bit count, enabling or disabling the MDIO driver output (see Figure 22.

**Figure 22    MDI State**

### 5.5.8 Autoscan Operation

The autoscan function allows the 32 registers in each external PHY (up to four) to be stored internally in the IXF1104 MAC. Autoscan is enabled by setting bit 1 of the MDI Control register. When enabled, autoscan runs continuously, reading each PHY register. When a PHY register access is instigated through the CPU interface, the current autoscan register Read is completed before the CPU register access starts. Upon completion of the CPU-induced access, the autoscan functionality restarts from the last autoscan register access.

The *Autoscan PHY Address Enable ($0x682)* determines which PHY addresses are being occupied for each IXF1104 MAC port. The least significant bit (LSB) that is set in the register is Port 0, the next significant bit that is set is assumed to be port 1, and so on. If more than four bits are set, the bits beyond the fourth bit are ignored. If less than four bits are set, the round-robin process returns to the port identified by the LSB being set.

## 5.6 SerDes Interface

The IXF1104 MAC integrates four integrated Serializer/Deserializer (SerDes) devices that allow direct connection to optical modules and remove the requirement for external SerDes devices. This increases integration, which reduces the size of the PCB area required to implement this function, reduces total power, reduces silicon and manufacturing costs, and improves reliability. Each SerDes interface is identical and fully compliant with the relevant IEEE 802.3 Specifications, including auto-negotiation. Each port is also compliant with and supports the requirements of the Small Form Factor Pluggable (SFP) Multi-Source Agreement (MSA), see Section 5.7, *Optical Module Interface*, on page 106.

The following sections describe the operations supported by each interface, the configurable options, and the register bits that control these options. A full list of the register addresses and full bit definitions are found in the register maps (Table 59 through Table 69).

### 5.6.1 Features

The SerDes cores are designed to operate in point-to-point data transmission applications. While the core can be used across various media types, such as PCB or backplanes, it is configured specifically for use in 1000BASE-X Ethernet fiber applications in the IXF1104 MAC. The following features are supported.

- 10-bit data path, which connects to the output/input of the 8B/10B encoder/decoder PCS that resides in the MAC controller

- Data frequency of 1.25 GHz

- Low power: <200 mW per SerDes port

- Asynchronous clock data recovery

### 5.6.2 Functional Description

The SerDes transmit interface sends serialized data at 1.25 GHz. The interface is differential with two signals for transmit operation. The transmit interface is designed to operate in a 100 $\Omega$ differential environment and all the terminations are included on the device. The outputs are high-speed SerDes and are capable of operating in either an AC- or DC-coupled environment. AC coupling is recommended for this interface to ensure that the correct input bias current is supplied at the receiver.

The SerDes receive interface receives serialized data at 1.25 GHz. The interface is differential with two signals for the receive operation. The equalizer receives a differential signal that is equalized for the assumed media channel. The SerDes transmit and receive interfaces are designed to operate within a 100 Ω differential environment and all terminations are included on the device. The SerDes is capable of operating in either AC- or DC-coupled environments.

### 5.6.2.1 Transmitter Operational Overview

The transmit section of the IXF1104 MAC has to serialize the Ten Bit Interface (TBI) data from the IXF1104 MAC MAC section and outputs this data at 1.25 GHz differential signal levels. The 1.25 GHz differential SerDes signals are compliant with the Small Form Factor Pluggable (SFP) Multi-Source Agreement (MSA).

The transmitter section takes the contents of the data register within the MAC and synchronously transfers the data out, ten bits at a time – Least Significant Bit (LSB) first, followed by the next Most Significant Bit (MSB). When these ten bits have been serialized and transmitted, the next word of 10-bit data from the MAC is ready to be serialized for transmission.

The data is transmitted by the high-speed current mode differential SerDes output stage using an internal 1.25 GHz clock generated from the 125 MHz clock input.

### 5.6.2.2 Transmitter Programmable Driver-Power Levels

The IXF1104 MAC SerDes core has programmable transmitter power levels to enhance usability in any given application.The SerDes Registers are programmable to allow adjustment of the transmit core driver output power. When driving a 100 Ω differential terminated network, these output power settings effectively establish the differential voltage swings at the driver output.

The *TX Driver Power Level Ports 0 - 3 ($0x784)* allows the selection of four discrete power settings. The selected power setting of these inputs is applied to each of the transmit core drivers on a per-port basis. Table 29, *SerDes Driver TX Power Levels* lists the normalized power settings of the transmit drivers as a function of the Driver Power Control inputs. The normalized current setting is 10 mA, which corresponds to the normalized power setting of 1.0. This is the default setting of the IXF1104 MAC SerDes interface. Other values listed in the Normalized Driver Power Setting column are multiples of 10 mA. For example, with inputs at 1110, the driver power is the following:

**Equation 1**    .5 x 10 mA = 5 mA.

**Table 29    SerDes Driver TX Power Levels**

| DRVPWRx[3] | DRVPWRx[2] | DRVPWRx[1] | DRVPWRx[0] | Normalized Driver Power Setting | Driver Power |
|---|---|---|---|---|---|
| 0 | 0 | 1 | 1 | 1.33 | 13.3 mA |
| 1 | 0 | 1 | 1 | 2.0 | 20 mA |
| 1 | 1 | 0 | 1 | 1.0 | 10 mA |
| 1 | 1 | 1 | 0 | 0.5 | 5 mA |
| *Note:*   All other values are reserved. | | | | | |

### 5.6.2.3     Receiver Operational Overview

The receiver structure performs Clock and Data Recovery (CDR) on the incoming serial data stream. The quality of this operation is a dominant factor for the Bit Error Rate (BER) system performance. Feed forward and feedback controls are combined in one receiver architecture for enhanced performance. The data is over-sampled and a digital circuit detects the edge position in the data stream. A signal is not generated if an edge is not found. A feedback loop takes care of low-frequency jitter phenomenon of unlimited amplitude, while a feed forward section suppresses high-frequency jitter having limited amplitude. The static edge position is held at a constant position in the over-sampled by a constant adjustment of the sampling phases with the early and late signals.

### 5.6.2.4     Selective Power-Down

The IXF1104 MAC offers the ability to selectively power-down any of the SerDes TX or RX ports that are not being used. This is done via Section Table 150, *TX and RX Power-Down ($0x787), on page 210*.

### 5.6.2.5     Receiver Jitter Tolerance

The SerDes receiver architecture is designed to track frequency mismatch, recover phase, and is tolerant of low-frequency data jitter. Figure 23 specifies the SerDes core receiver sinusoidal jitter tracking capabilities.

**Figure 23**     **SerDes Receiver Jitter Tolerance**



Note:          UI = Unit interval.
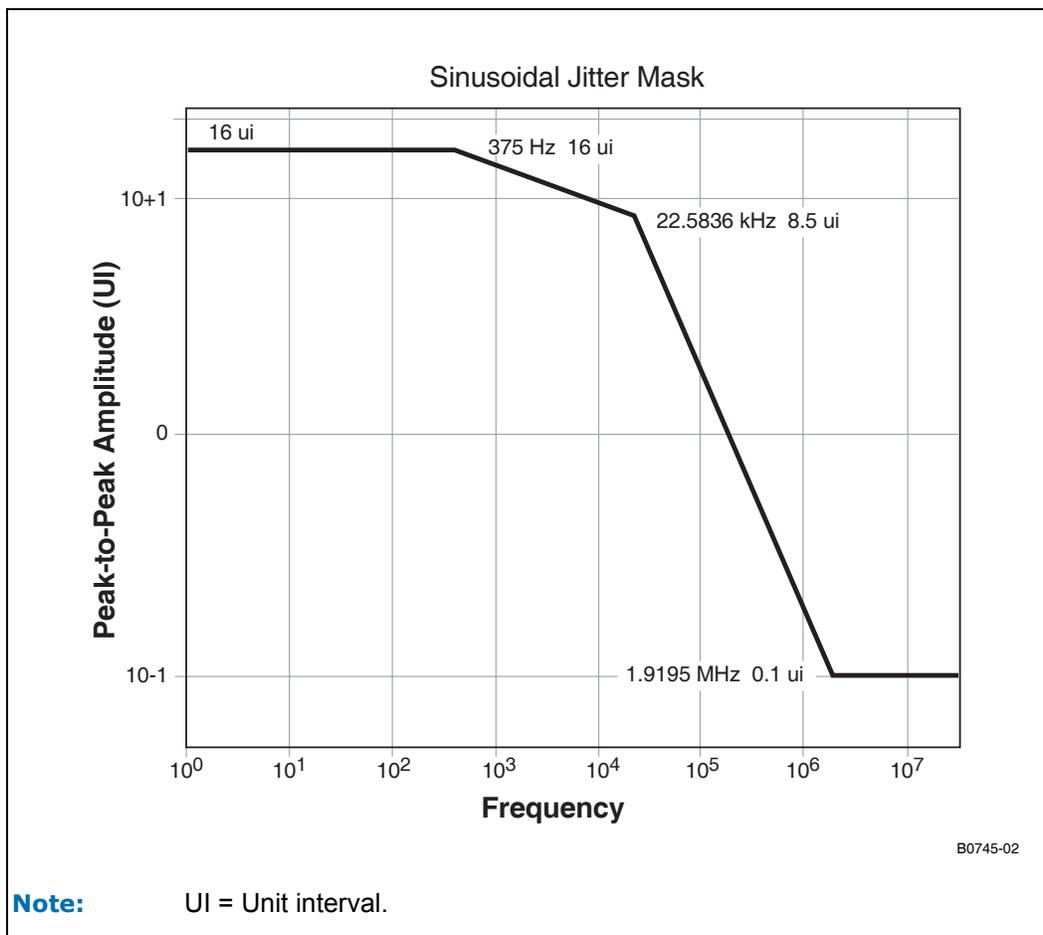
### 5.6.2.6     Transmit Jitter

The SerDes core total transmit jitter, including contributions from the intermediate frequency PLL, is comprised of the following two components:

- A deterministic component attributed to the SerDes core's architectural characteristics
- A random component attributed to random thermal noise effects

Since the thermal noise component is random and statistical in nature, the SerDes core total transmit jitter must be specified as a function of BER.

### 5.6.2.7     Receive Jitter

The SerDes core total receiver jitter, including contributions from the intermediate frequency PLL, is comprised of the following two components:

- A deterministic component attributed to the SerDes core architectural characteristics
- A random component attributed to random thermal noise effects.

## 5.7 Optical Module Interface

This section describes the connection of the IXF1104 MAC ports to an Optical Module Interface and details the minimal connections that are supported for correct operation. The registers used for write control and read status information are documented.

The Optical Module Interface allows the IXF1104 MAC a seamless connection to the Small Form Factor Optical Modules (SFP) that form the system's physical media connection, eliminating the need for any FPGAs or CPUs to process data. All required optical module information is available to the system CPU through the IXF1104 MAC CPU interface, leading to a more integrated, reliable, and cost-effective system.

The IXF1104 MAC supports all the functions required for the Small Form Factor pluggable Multi-Source Agreement (MSA).

There are specific mechanical and electrical requirements for the size, form factor, and connections supported on all Optical Module Interfaces. There are also specific requirements for each Optical Module Interface that supports a particular media requirement or interface configuration. These requirements are detailed in the relevant specifications or manufacturers' datasheets.IXF1104 MAC

### 5.7.1 IXF1104 MAC-Supported Optical Module Interface Signals

To describe the Optical Module Interface operation, three supported signal subgroups are required, allowing a more explicit definition of each function and implementation. The three subgroups are as follows:

- High-Speed Serial Interface
- Low-Speed Status Signaling Interface
- I²C Module Configuration Interface

Table 30 provides descriptions for IXF1104 MAC-to-SFP optical module connection signals.

**Table 30    IXF1104 MAC-to-SFP Optical Module Interface Connections**

| IXF1104 MAC Signal Names | SFP Signal Names | Description | Notes |
|---|---|---|---|
| TX_P_0:3 | TD+ | Transmit Data, Differential LVDS | Output from the IXF1104 MAC |
| TX_N_0:3 | TD- | Transmit Data, Differential LVDS | Output from the IXF1104 MAC |
| RX_P_0:3 | RD+ | Receive Data, Differential LVDS | Input to the IXF1104 MAC |
| RX_N_0:3 | RD- | Receive Data, Differential LVDS | Input to the IXF1104 MAC |
| I²C_CLK | MOD-DEF1 | I²C_CLK output from the IXF1104 MAC (SCL) | Output from the IXF1104 MAC |
| I²C_DATA_0:3 | MOD-DEF2 | I²C_DATA I/O (SDA) | Input/Output |
| MOD_DEF_0:3 | MOD-DEF0 | MOD_DEF_0 is TTL Low level during normal operation. | Input to the IXF1104 MAC |
| TX_DISABLE_0:3 | TX DISABLE | Transmitter disable, logic High, open collector compatible | Output from the IXF1104 MAC |
| TX_FAULT_0:3 | TX FAULT | Transmitter fault, logic High, open collector compatible | Input to the IXF1104 MAC |
| RX_LOS_0:3 | LOS | Receiver loss-of-signal, logic High, open collector compatible | Input to the IXF1104 MAC |

## 5.7.2　Functional Descriptions

### 5.7.2.1　High-Speed Serial Interface

These signals are responsible for transfer of the actual data at 1.25 Gbps. Table 41, *DC Specifications*, on page 131 shows the data is 8B/10B encoded and transmitted differentially.

The following signals are required to implement the high-speed serial interface:

- TX_P_0:3
- TX_N_0:3
- RX_P_0:3
- RX_N_0:3

### 5.7.2.2　Low-Speed Status Signaling Interface

The following Low-Speed signals indicate the state of the line through the Optical Module Interface:

- MOD_DEF_0:3
- TX_FAULT_0:3
- RX_LOS_0:3
- TX_DISABLE_0:3
- MOD_DEF_INT
- TX_FAULT_INT
- RX_LOS_INT

#### 5.7.2.2.1　MOD_DEF_0:3

MOD_DEF_0:3 are direct inputs to the IXF1104 MAC and are pulled to a logic Low level during normal operation, indicating that a module is present for each channel respectively. If a module is not present, a logic High is received, which is achieved by an external pull-up resistor at the IXF1104 MAC device pad.

The status of each bit (one for each port) is found in bits [3:0] of the Table 153 on page 212). Any change in the state of these bits causes a logic Low level on the MOD_DEF_INT output if this operation is enabled.

#### 5.7.2.2.2　TX_FAULT_0:3

TX_FAULT_0:3 are inputs to the IXF1104 MAC. These signals are pulled to a logic Low level by the optical module during normal operation. A logic Low level on these signals indicates no fault condition exists. If a fault is present, a logic High is received through the use of an external pull-up resistor at the IXF1104 MAC pad.

The status of each bit (one for each port) can be found in bits [13:10] of the Table 153 on page 212. Any change in the state of these bits causes a logic Low level on the TX_FAULT_INT output if this operation is enabled.

### 5.7.2.2.3    RX_LOS_0:3

RX_LOS_0:3 are inputs to the IXF1104 MAC. These signals are pulled to a logic Low level by the optical module during normal operation, which indicates that no loss-of-signal exists. If a loss-of-signal occurs, a logic High is received on these inputs through the use of an external pull-up resistor at the IXF1104 MAC device pad.

The status of each bit (one for each port) is found in *Optical Module Status Ports 0-3 ($0x799)* bits [23:20]. Any change in the state of these bits causes a logic Low level on the RX_LOS_INT output if this operation is enabled.

### 5.7.2.2.4    TX_DISABLE_0:3

TX_DISABLE_0:3 are outputs from the IXF1104 MAC. These signals are driven to a logic Low level by the IXF1104 MAC during normal operation. This indicates that the optical module transmitter is enabled. If the optical module transmitter is disabled, this signal is switched to a logic High level. On the IXF1104 MAC, these outputs are open drain types and pulled up by the 4.7 k to 10 k pull-up resistor at the Optical Module Interface. Each of these signals is controlled through bits 3:0 respectively of the *Optical Module Control Ports 0 - 3 ($0x79A)*.

### 5.7.2.2.5    MOD_DEF_INT

MOD_DEF_INT is a single output, open-drain type signal and is active Low. A change in state of any MOD_DEF_0:3 inputs causes this signal to switch Low and remain in this state until a read of the *Optical Module Status Ports 0-3 ($0x799)*. The signal then returns to an inactive state.

### 5.7.2.2.6    TX_FAULT_INT

TX_FAULT_INT is a single output, open-drain type signal and is active Low. A change in state of any TX_FAULT_0:3 inputs causes this signal to switch Low and remain in this state until a read of the *Optical Module Status Ports 0-3 ($0x799)*. The signal then returns to an inactive state.

### 5.7.2.2.7    RX_LOS_INT

RX_LOS_INT is a single output, open-drain type signal and is active low. A change in state of any of the RX_LOS_3:0 inputs causes this signal to switch low and remain in this state until a Read of the *Optical Module Status Ports 0-3 ($0x799)* has taken place. The signal returns to an inactive state.

**Note:**    MOD_DEF_INT, TX_FAULT_INT, and RX_LOS_INT are open-drain type outputs. With the three signals on the device, the system can decide which *Optical Module Status Ports 0-3 ($0x799)* bits to look at to identify the interrupt condition source port. However, this is achieved at the expense of the three device signals.

## 5.7.3    I²C Module Configuration Interface

The I²C interface is supported on SFP optical modules. Details of the operation are found in the SFP Multi-Source Agreement, which details the contents of the registers and addresses accessible on a given Optical Module Interface supporting this interface.

The SFP MSA identifies up to 512 8-bit registers that are accessible in each optical module. The Optical Module Interface is read-only and supports either sequential or random access to the 8-bit parameters. The maximum clock rate of the interface is 100 kHz. All address-select signals on the internal E²PROM are tied Low to give a device address equal to zero (00h).

Several PHY vendors may offer copper/CAT5-based SFP optical compliant modules. To program the internal configuration registers of these modules, the IXF1104 MAC I$^2$C interface needs to provide the capability to write data to the SFP modules.

The IXF1104 MAC I$^2$C interface is designed to allow individual writes of byte-wide data to the SFP.

The specific interface in the IXF1104 MAC supports only a subset of the full I²C interface, and only the features required to support the Optical Module Interfaces are implemented. This leads to the following support features.

- Single I$^2$C_CLK pin connected to all optical modules and implemented to save unnecessary signals use.

- Four per-port I$^2$C_DATA signals (I²C Data[3:0]) are required because of the optical module requirement that all modules must be addressed as 00h.

- The interface has both read and write functionality.

- Due to the single internal optical module controller, only one optical module may be accessed at any one time. Each access contains a single register Read. Since these register accesses will most likely be done during power-up or discovery of a new module, these restrictions should not affect normal operation.

- The I$^2$C interface supports byte write accesses to the full address range.

**Note:** The I$^2$C interface only supports random single-byte reads and does not guarantee coherency when reading two-byte registers.

## 5.7.3.1 I$^2$C Control and Data Registers

In the IXF1104 MAC, the entire I²C interface is controlled through the following two registers:

- Section Table 155, *I$^2$C Control Ports 0 - 3 ($0x79B)*, on page 214
- Section Table 156, *I$^2$C Data Ports 0 - 3 ($0x79F)*, on page 214

These registers can be programmed by system software using the CPU interface.

## 5.7.3.2 I$^2$C Read Operation

To perform a read operation using the I$^2$C interface, use the following sequence:

1. Initialize the Control register by setting the following values:

   a. Enable the I$^2$C Controller by setting bit [25] to 0x1.

   b. Initiate the I$^2$C transfer by setting bit [24] of the control register to 0x1.

   c. Select the port by using bits [17:16].

   d. Select the Read mode of operation by setting bit [15] to 0x1.

   e. Select the Device ID by setting bits [14:11].

   f. Select the register address by setting bits [10:0].

2. Set the Device ID field to 0xA and the register address (bits 10:8) to 0x0 to access the fiber module serial E2PROM. Setting the Device ID field to 0xA and the Register Address [10:8] to 0x0 permits read-only access.

3. Set the Device ID field to 0xA and the Register Address [10:8] between the values of 0x1 and 0x7 to access the PHY registers.

4. Poll the Read_Valid field, bit 20. The read data is available when this bit is set to 0x1.

Figure 24 shows an 8-bit read access.

**Note:** The user software ensures the order of the contiguous accesses required to read the High and Low bytes of 16-bit-wide PHY registers.

**Figure 24** **I$^2$C Random Read Transaction**
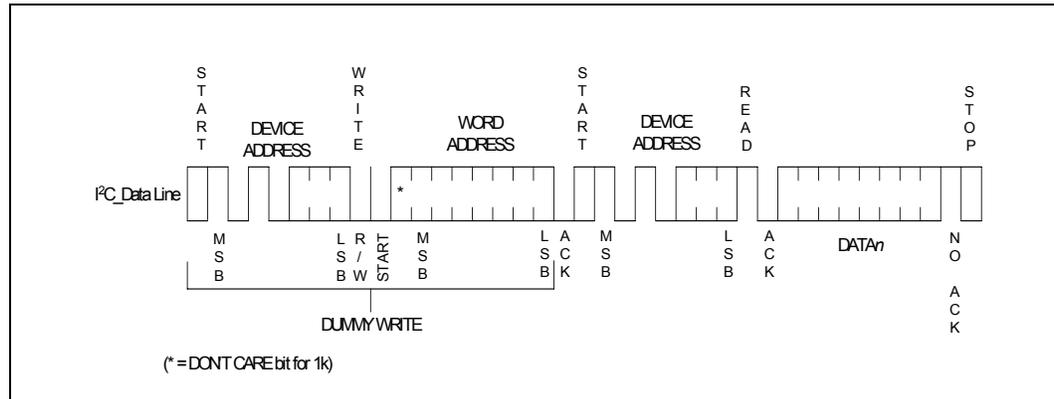


**Note:** Only one optical module I²C access sequence can be run at any given time. If a second write is carried out to the *I$^2$C Control Ports 0 - 3 ($0x79B)* and *I$^2$C Data Ports 0 - 3 ($0x79F)* before a result is returned for the previous write, the data for the first write is lost. An internal state machine completes the Optical Module Interface register access for the first write. It attempts to place the data in the DataRead field and checks to see if the WriteCommand bit is 00h. If it is not 00h, it discards the data and signals the I²C access state machine to begin a new cycle using the data from the second write.

### 5.7.3.3 I$^2$C Write Operation

The following sequence provides an example of writing data to Register Address 0xFF for Port 3:

1. Program the *I$^2$C Control Ports 0 - 3 ($0x79B)* with the following information:

    a. Enable the I$^2$C block by setting Register bit 25 to 0x1.

    b. Set the port to be accessed by setting Register bits 17:16 to 0x3.

    c. Select a Write access by setting Register bit 15 to 0x0.

    d. Set the Device ID Register bits 14:11 to Ah (Atmel compatible).

    e. Set the 11-bit register address (Register bits 10:0) to 0FFh.

    f. Enable the I$^2$C controller by setting Register bit 2 to 0x1.

    g. Initiate the I$^2$C transfer by setting Register bit 24 to 0x1.

All other bits in this register should be set to 0x0.

This data is written into the *I$^2$C Control Ports 0 - 3 ($0x79B)* in a single cycle via the CPU interface.

2. When this register is written and the I$^2$C Start bit is at a Logic 1, the I$^2$C access state machine examines the Port Address Select and enables the I$^2$C_DATA_0:3 output for the selected port.

3. The state machines uses the data in the Device ID and Register Address fields to build the data frame to be sent to the optical module

4. The I²C_DATA_WRITE_FSM internal state machine takes over the task of transferring the actual data between the IXF1104 MAC and the selected optical module (refer to the details in Section 5.7.3.4, *I²C Protocol Specifics*, on page 111).

5. The I²C_DATA_WRITE_FSM internal state machine uses the data from the Write_Data field bits [23:16] of the Table 156 on page 214 and sets the Write_Complete Register bit 22 of the *I²C Control Ports 0 - 3 ($0x79B)* to 0x1 to signify that the Write Access is complete.

6. The data is written through the CPU interface. The CPU must poll the Write_Complete bit until it is set to 0x1. It is safe to request a new access *only* when this bit is set.

**Note:**    Only one optical module I²C access sequence can be run at any given time. The data for the first Write is lost if a second Write is carried out to the *I²C Control Ports 0 - 3 ($0x79B)* before a result is returned for the previous Write. Make sure Write complete = 0x1 before starting the next Write sequence to ensure that no data is lost.

### 5.7.3.4    I²C Protocol Specifics

Section 5.7.3.4 describes the IXF1104 MAC I²C Protocol behavior, which is controlled by an internal state machine. Specific protocol states are defined below, with an additional description of the hardware signals used on the interface.

The Serial Clock Line (I²C_CLK) is an output from the IXF1104 MAC. The serial data is synchronous with this clock and is driven off the rising edge by the IXF1104 MAC and off the falling edge by the optical module. The IXF1104 MAC has only one I²C_CLK line that drives all of the optical modules. I²C_CLK runs continuously when enabled (I²C Enable = 01h0).

The Serial Data (I²C_DATA_3:0) signals (one per port) are bi-directional for serial data transfer. These signals are open drain.

### 5.7.3.5    Port Protocol Operation

### 5.7.3.6    Clock and Data Transitions

The I²C_DATA is normally pulled High with an extra device. Data on the I²C_DATA pin changes only during the I²C_CLK Low time periods (see Figure 25). Data changes during I²C_CLK High periods indicate a start or stop condition.

**Figure 25    Data Validity Timing**
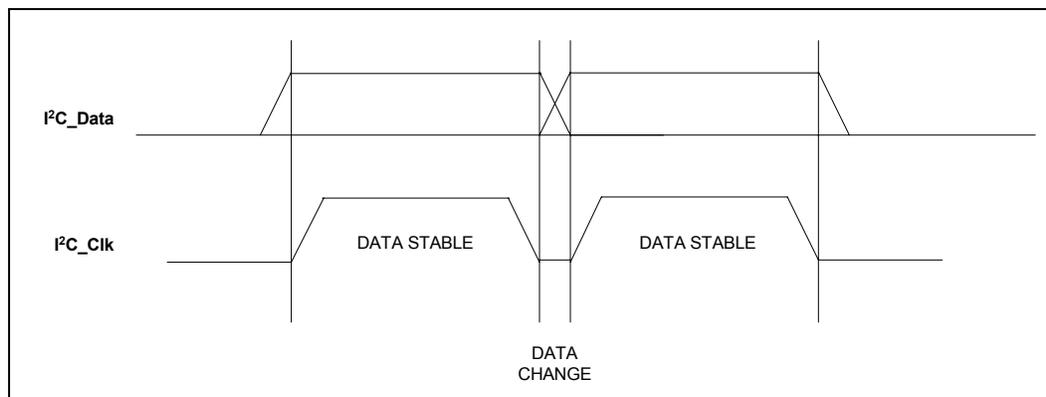
#### 5.7.3.6.1 Start Condition

A High-to-Low transition of I$^2$C_DATA, with I$^2$C_CLK High, is a start condition that must precede any other command (see Figure 26).

#### 5.7.3.6.2 Stop Condition

A Low-to-High transition of the I$^2$C_DATA with I$^2$C_CLK High is a stop condition. After a Read sequence, the stop command places the E²PROM and the optical module in a standby power mode (see Figure 26).

**Figure 26    Start and Stop Definition Timing**
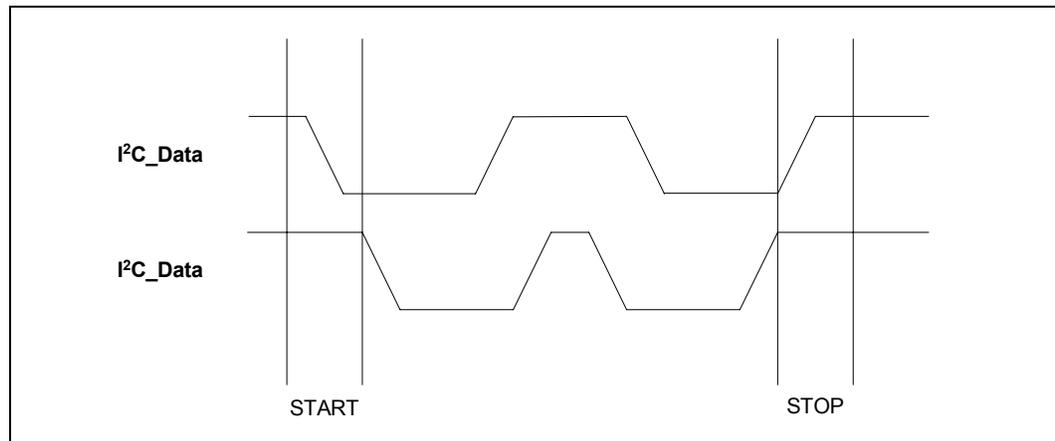


#### 5.7.3.6.3 Acknowledge

All addresses and data words are serially transmitted to and from the optical module in 8-bit words. The optical module E²PROM sends a zero to acknowledge that it has received each word, which happens during the ninth clock cycle (see Figure 27).

**Figure 27    Acknowledge Timing**
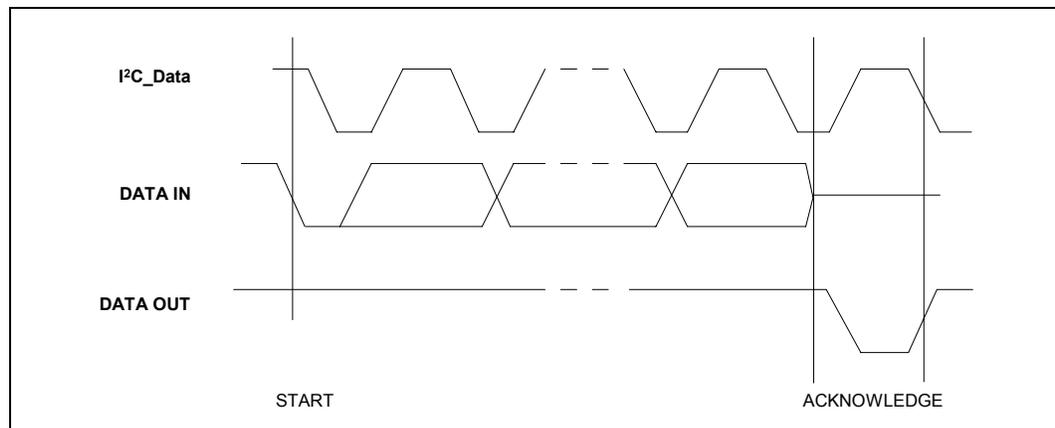


#### 5.7.3.6.4 Memory Reset

After an interruption in protocol, power loss, or system reset, any 2-wire optical module can be reset by following three steps:

1. Clock up to 9 cycles

2. Wait for I$^2$C_DATA High in each cycle while I$^2$C_CLK is High

3. Initiate a start condition.

### 5.7.3.6.5 Device Addressing

All E²PROMs in SFP optical module devices require an 8-bit device address word following a start condition to enable the chip to read or write. The device address word consists of a mandatory one, zero sequence for the four most-significant bits. This is common to all devices. The next three bits are the A2, A1, and A0 device address bits that are tied to zero in an optical module. The eighth bit of the device address is the Read/ Write operation select bit. A Read operation is initiated if this bit is High and a Write operation is initiated if this bit is Low.

Upon comparison of the device address, the optical module outputs a zero. If a comparison is not made, the optical module E²PROM returns to a standby state.

### 5.7.3.6.6 Random Read Operation

A random Read requires a "dummy" Byte/Write sequence to load the data word address. The "dummy" write is achieved by first sending the device address word with the Read/ Write bit cleared to Low, which signals a Write operation. The optical module acknowledges receipt of the device address word. The IXF1104 MAC sends the data word address, which is again acknowledged by the optical module. The IXF1104 MAC generates another start condition. This completes the "dummy" write and sets the optical module E²PROM pointers to the desired location.

The IXF1104 MAC initiates a current address read by sending a device address with the Read/Write bit set High. The optical module acknowledges the device address and serially clocks out the data word. The IXF1104 MAC does not respond with a zero but generates a stop condition (see Figure 28).

### Figure 28 Random Read
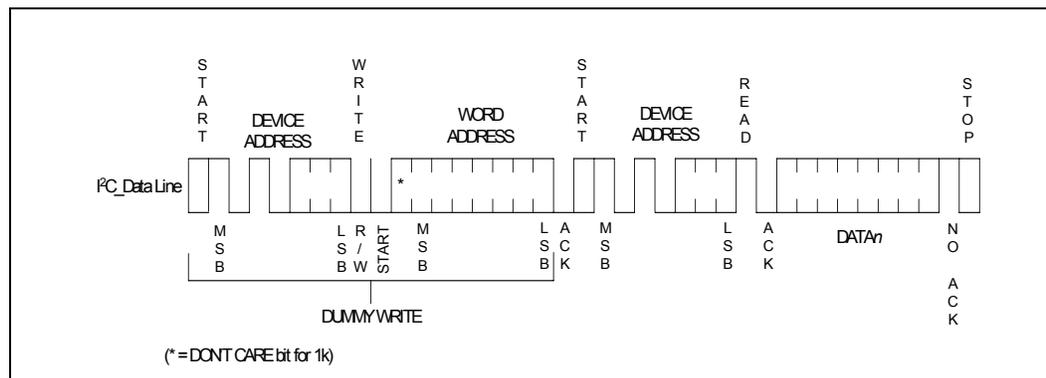


## 5.8 LED Interface

The IXF1104 MAC uses a Serial interface, consisting of three signals, to provide LED data to some form of external driver. This provides the data for 12 separate direct drive LEDs and allows three LEDs per MAC port.

There are two modes of operation, each with its own separate LED decode mapping. Modes of operation and LEDs are detailed in the following sections.

## 5.8.1 Modes of Operation

There are two modes of operation: Mode 0 and Mode 1. Mode selection is accomplished by using the LED_SEL_MODE bit. This bit is globally selected and controls the operation of all ports (see Table 109, *LED Control ($0x509),* on page 183).

**Mode 0: (LED_SEL_MODE = 0 [Default]):** This mode selects operations compatible with the SGS Thompson M5450 LED Display Driver device. This device converts the serial data stream, output by the IXF1104 MAC, into 30 direct-drive LED outputs. Although the LED interface is capable of driving all 30 LEDs, only twelve will be driven in the four-port IXF1104 MAC, three LEDs per port.

**Mode 1: (LED_SEL_MODE = 1):** This mode is used with standard TTL (74LS599) or HCMOS (74HC599) octal shift registers with latches, providing the most general and cost-effective implementation of the serial data stream conversion.

In addition to these physical modes of operation, there are two types of specific LED data decodes available for fiber and copper modes. This option is a global selection and controls the operation of all ports (see Table 109, *LED Control ($0x509),* on page 183).

## 5.8.2 LED Interface Signal Description

The IXF1104 MAC LED interface consists of three output signal signals that are 2.5 V CMOS level pads. Table 31 provides LED signal names, pin numbers, and descriptions.

**Table 31**    **LED Interface Signal Descriptions**

| Pin Name | Pin # | Pin Description |
|---|---|---|
| LED_CLK | K24 | This signal is an output that provides a continuous clock synchronous to the serial data stream output on the LED_DATA pin. This clock has a maximum speed of 720 Hz.<br>The behavior of this signal remains constant in all modes of operation. |
| LED_DATA | M22 | This signal provides the data, in various formats, as a serial bit stream. The data must be valid on the rising edge of the LED_CLK signal.<br>In Mode 0, the data presented on this pin is TRUE (Logic 1 = High).<br>In Mode 1, the data presented on this pin is INVERTED (Logic 1 = Low). |
| LED_LATCH | L22 | This is an output pin, and the signal is used only in Mode 1 as the Latch enable for the shift register chain.<br>This signal is not used in Mode 0, and should be left unconnected. |

## 5.8.3 Mode 0: Detailed Operation

**Note:** Please refer to the SGS Thompson* M5450 datasheet for device-operation information.

The operation of the LED Interface in Mode 0 is based on a 36-bit counter loop. The data for each LED is placed in turn on the serial data line and clocked out by the LED_CLK. Figure 29 shows the basic timing relationship and relative positioning in the data stream of each bit.

Figure 29 shows the 36 clocks that are output on the LED_CLK pin. The data is changed on the falling edge of the clock and is valid for almost the entire clock cycle. This ensures that the data is valid during the rising edge of the LED_CLK, which clocks the data into the M5450 device.

The actual data shown in Figure 29 consists of a chain of 36 bits, 12 of which are valid LED DATA. The 36-bit data chain is built up as follows:

**Figure 29**    **Mode 0 Timing**
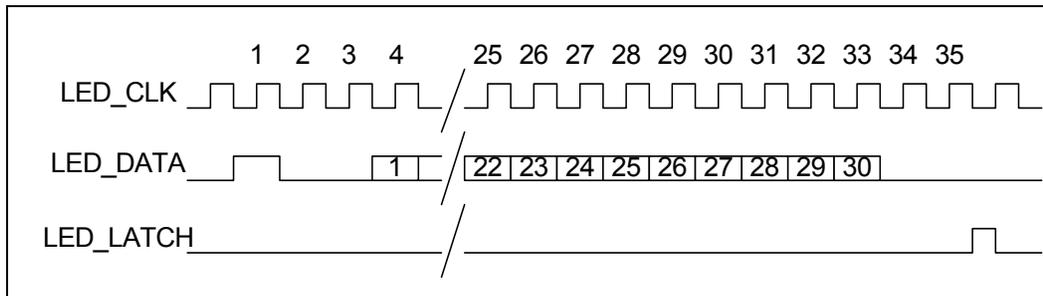


**Table 32**    **Mode 0 Clock Cycle to Data Bit Relationship**

| LED_CLK Cycle | LED_DATA Name | LED_DATA Description |
|---|---|---|
| 1 | START BIT | This bit synchronizes the M5450 device to expect 35 bits of data to follow. |
| 2:3 | PAD BITS | These bits are used only as fillers in the data stream to extend the length from the actual 12-bit LED DATA to the required 18-bit frame length. These bits should always be a logic 0. |
| 4:15 | LED DATA 1-12 | These bits are the actual data transmitted to the M5450 device. The decode for each individual bit in each mode is defined in Table 34 on page 117.<br><br>The data is TRUE. Logic 1 (LED ON) = High |
| 36:38 | PAD BITS | These bits are used as fillers in the data stream to extend the length from the actual 30-bit LED DATA to the required 36-bit frame length. These bits should always be a logic 0. |

When implemented on the board with the M5450 device, the LED DATA bit 1 appears on Output bit 3 of the M5450 and the LED DATA bit 2 appears on Output bit 4, etc. This means that Output bits 1, 2, and 15 through 35 will never have valid data and should not be used.

## 5.8.4    Mode 1: Detailed Operation

**Note:**    Please refer to generic specifications for 74LS/HC599 for information on device operation.

The operation of the LED Interface in Mode 1 is based on a 36-bit counter loop. The data for each LED is placed in turn on the serial data line and clocked out by the LED_CLK. Figure 30 on page 116 shows the basic timing relationship and relative positioning in the data stream of each bit.

Figure 30 on page 116 shows the 36 clocks which are output on the LED_CLK pin. The data is changed on the falling edge of the clock and is valid for the almost the entire clock cycle. This ensures that the data is valid during the rising edge of the LED_CLK, which clocks the data into the shift register chain devices.

The LED_LATCH signal is required in Mode 1, and latches the data shifted into the shift register chain into the output latches of the 74HC599 device. Figure 30 shows that the LED_LATCH signal is active High during the Low period on the 35th LED_CLK cycle. This avoids any possibility of trying to latch data as it is shifting through the register.

When this operation mode is implemented on a board with a shift register chain containing three 74HC599 devices, the LED DATA bit 1 is output on Shift register bit 1, and so on up the chain. Only Shift register bits 31 and 32 do not contain valid data.

The actual data shown in Figure 30 consists of a 36-bit chain, of which 12 bits are valid LED DATA. The 36-bit data chain is built up as shown in Figure 30.

**Note:** The LED_DATA signal is now inverted from the state in Mode 0.

**Figure 30     Mode 1 Timing**
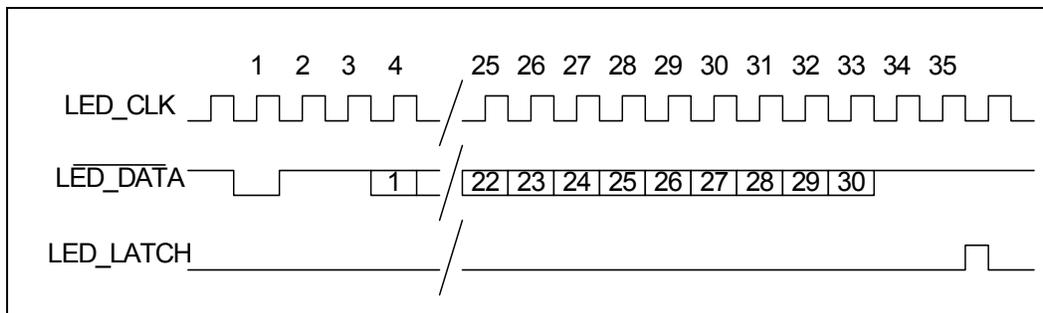


**Table 33     Mode 1 Clock Cycle to Data Bit Relationship**

| LED_CLK Cycle | LED_DATA Name | LED_DATA Description |
|---|---|---|
| 1 | START BIT | This bit has no meaning in Mode 1 operation and is shifted out of the 16-stage shift register chain before the LED_LATCH signal is asserted. |
| 2:3 | PAD BITS | These bits have no meaning in Mode 1 operation and are shifted out of the 16-stage shift register chain before the LED_LATCH signal is asserted. |
| 4:15 | LED DATA 1-12 | These bits are the actual data to be transmitted to the 16-stage shift register chain. The decode for each bit in each mode is defined in Table 34 on page 117.<br>The data is INVERTD. Logic 1 (LED ON) = Low. |
| 36:38 | PAD BITS | These bits have no meaning in Mode 1 operation and are latched into positions 31 and 32 in the shift register chain. These bits are not considered as valid data and should be ignored. They should always be a Logic 0 = High. |

## 5.8.5     Power-On, Reset, Initialization

The LED interface is disabled at power-on or reset. The system software controller must enable the LED interface. The internal state machines and output signals are held in reset until the full Cortina Systems® IXF1104 4-Port Gigabit Ethernet Media Access Controller device configuration is completed. This is done by setting the LED_ENABLE bit to a logic 1 (see Table 109, *LED Control ($0x509),* on page 183). The power-on default for this bit is logic 0.

## 5.8.6     LED DATA Decodes

The data transmitted on the LED_DATA line is determined by programming the global operation mode as either fiber or copper. Table 34 shows the data decode of the data for both fiber and copper MACs.

**Note:** The data decode of the LED bits is independent of the Physical mode selection.

**Table 34    LED_DATA# Decodes**

| LED_DATA# | MAC Port # | Fiber Designation | Copper Designation |
|---|---|---|---|
| 1 | 0 | Rx LED—Amber | Link LED—Amber |
| 2 | | Rx LED—Green | Link LED—Green |
| 3 | | TX LED—Green | Activity LED—Green |
| 4 | 1 | Rx LED—Amber | Link LED—Amber |
| 5 | | Rx LED—Green | Link LED—Green |
| 6 | | TX LED—Green | Activity LED—Green |
| 7 | 2 | Rx LED—Amber | Link LED—Amber |
| 8 | | Rx LED—Green | Link LED—Green |
| 9 | | TX LED—Green | Activity LED—Green |
| 10 | 3 | Rx LED—Amber | Link LED—Amber |
| 11 | | Rx LED—Green | Link LED—Green |
| 12 | | TX LED—Green | Activity LED—Green |

## 5.8.6.1    LED Signaling Behavior

Operation in each mode for the decoded LED data in Table 34 is detailed in Table 35 and Table 36.

### 5.8.6.1.1    Fiber LED Behavior

**Table 35    LED Behavior (Fiber Mode)**

| Type | Status | Description |
|---|---|---|
| RXLED | Off | Synchronization occurs but no packets are received and the Section 105, *Link LED Enable ($0x502)* is not set. |
| | Amber On | RX Synchronization has not occurred or no optical signal exists. |
| | Amber Blinking | The port has remote fault and the Section 105, *Link LED Enable ($0x502)* is not set (based on remote fault bit setting received in Rx_Config word). |
| | Green On | RX Synchronization occurs and the Section 105, *Link LED Enable ($0x502)* bit is set. |
| | Green Blinking | RX Synchronization occurs and the port is receiving data. |
| TXLED | Off | The port is not transmitting data or the Section 105, *Link LED Enable ($0x502)* is not set. |
| | Green Blinking | The port is transmitting data and the Section 105, *Link LED Enable ($0x502)* bit is set |

*Note:* Table 35 assumes the port is enabled in the Section 103, *Port Enable ($0x500)* and the LEDs are enabled in the Section 109, *LED Control ($0x509)*. If a port is not enabled, all the LEDs for that port will be off. If the LEDs are not enabled, all of the LEDs will be off.

### 5.8.6.1.2    Copper LED Behavior

**Table 36      LED Behavior (Copper Mode)**

| Type | Status | Description |
|---|---|---|
| Link LED | Off | Port does not have a remote fault and Section Table 109, *LED Control ($0x509)*, on page 183 bit is not set. |
| | Amber On | Port has an RGMII RXERR condition detected and Section Table 109, *LED Control ($0x509)*, on page 183 bit is set |
| | Amber Blinking | Port has a remote fault and Section Table 111, *LED Fault Disable ($0x50B)*, on page 184 is not set. |
| | Green On | Section Table 109, *LED Control ($0x509)*, on page 183 bit is set and port does not have an RGMII RXERR error or remote fault condition present. |
| Activity LED - Green | Off | Port is not transmitting and receiving data. |
| | Blinking | Section Table 109, *LED Control ($0x509)*, on page 183 set: Port is transmitting and/or receiving. |
| | | Section Table 109, *LED Control ($0x509)*, on page 183 not set: Port is receiving data. |

*Note:*    Table 34, *LED_DATA# Decodes* assumes the port is enabled in the Section Table 103, *Port Enable ($0x500)*, on page 181 and the LEDs are enabled in the Section Table 109, *LED Control ($0x509)*, on page 183. If a port is not enabled, all the LEDs for that port are off. If the LEDs are not enabled, all of the LEDs are off.

## 5.9        CPU Interface

The CPU interface block provides access to registers and statistics in the IXF1104 MAC. The interface is asynchronous externally and operates within the 125 MHz clock domain internally. The interface provides access to the following:

- Receive statistics registers
- Transmit statistics registers
- Receive FIFO registers
- Transmit FIFO registers
- Global configuration and control registers
- MAC_0 to MAC_3 registers

The CPU interface width can be configured with the two strap signals (UPX_WIDTH[1:0]) to operate as an 8-bit, 16-bit, or 32-bit bus. All internal accesses to registers are 32-bit (4, 2, or 1 data cycles respectively are required to fully access a register). When operating in 8-bit or 16-bit mode, read data for bytes [3:1] is strobed into read holding registers when byte [0] is read. Subsequent reads of bytes {1, 2, 3} in byte mode or of bytes {2,3} in 16-bit mode are supplied from the holding register independent of the upper address bits. On write accesses in 8-bit mode, the data of bytes {0, 1, 2} is similarly captured in internal write holding registers and the complete 32-bit write is committed when byte[3] is written to the IXF1104 MAC. When writing in 16-bit mode, bytes [1:0] are captured, and the double-word is committed when bytes [3:2] are written. The complete address for write is ignored (except for the write which causes the commit operation).

## 5.9.1 Functional Description

### 5.9.1.1 Read Access

Read access involves the following:

- Detect assertion of asynchronous Read control signal and latch address
- Generate internal Read strobe
- Drive valid data onto processor bus
- Assert asynchronous Ready signal for required length of time

Comment: shows the timing of the asynchronous interface for Read access.

**Comment:**  **Read Timing Diagram - Asynchronous Interface**



B5103-01
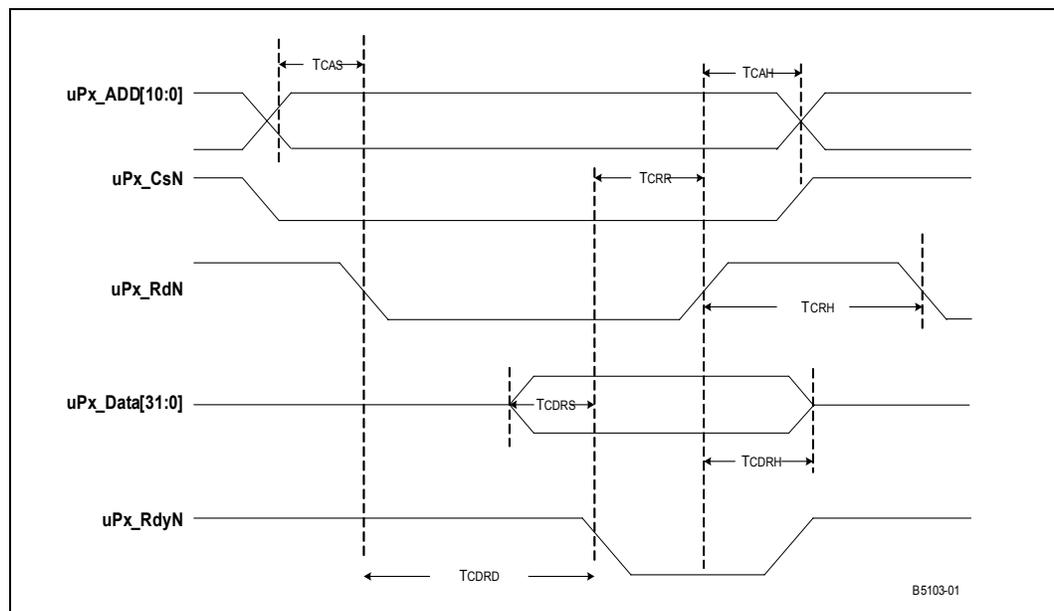
### 5.9.1.2 Write Access

Write process involves the following:

- Detect assertion of asynchronous Write control signal and latch address
- Detect de-assertion of asynchronous Write control signal and latch data
- Generate internal Write strobe
- Assert asynchronous Ready signal for required length of time

Figure 31 shows the timing of the asynchronous interface for Write accesses.

**Figure 31      Write Timing Diagram - Asynchronous Interface**
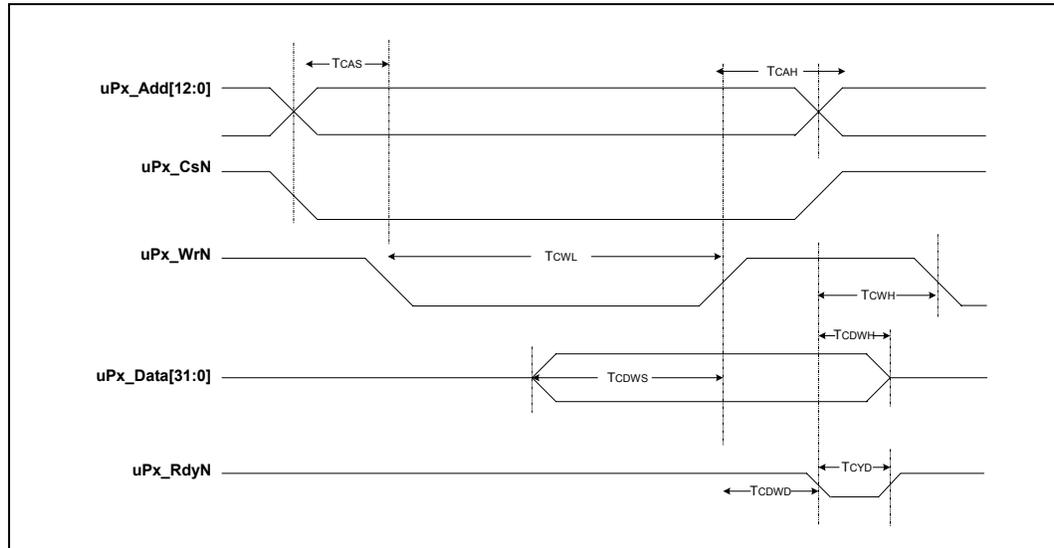


### 5.9.1.3      CPU Timing Parameters

For information on the CPU interface Read and Write cycle AC timing parameters, refer to Figure 46, *CPU Interface Read Cycle AC Timing*, on page 143, Figure 47, *CPU Interface Write Cycle AC Timing*, on page 144, and Table 54, *CPU Interface Write Cycle AC Signal Parameters*, on page 144.

### 5.9.2      Endian

The Endian of the CPU interface may be changed to allow connection of various CPUs to the Cortina Systems® IXF1104 4-Port Gigabit Ethernet Media Access Controller. The Endian selection is determined by setting the Endian bit in the *CPU Interface ($0x508)*.

The following describes Endianness control:

• There is a byte swapper between the internal 32-bit bus and the external 32-bit bus.

• In 8-bit or 16-bit mode operation, the byte packer/byte unpacker holding registers sink and source data just like the 32-bit external bus in 32-bit mode.

• The *CPU Interface ($0x508)* selects Big-Endian or Little-Endian mode.

• The byte swapper causes the behavior seen in Table 37 for accessing a register with data bits data[31:0].

**Table 37     Byte Swapper Behavior**

| UPX_BADD [1:0] | Little Endian | | | Big Endian | | |
|---|---|---|---|---|---|---|
| | 32-bit | 16-bit | 8-bit[1] | 32-bit | 16-bit | 8-bit[1] |
| | UPX_DATA_ [31:0] | UPX_DATA [15:0] | UPX_DATA [7:0] | UPX_DATA [31:0] | UPX_DATA [15:0] | UPX_DATA [7:0] |
| 00 | [31:0] | [15:0] | [7:0] | [7:0] [15:8] [23:16 [31:24] | [7:0] [15:8] | [7:0] |
| 01 | – | – | [15:8] | – | – | [15:8] |
| 10 | – | [31:16] | [23:16] | – | [23:16] [31:24] | [23:16] |
| 11 | – | – | [31:24] | – | – | [31:24] |

1.  In 8-bit mode, data is output in Little Endian format regardless of the IXF1104 MAC Endian setting.

## 5.10     TAP Interface (JTAG)

The IXF1104 MAC includes an IEEE 1149.1 compliant Test Access Port (TAP) interface used during boundary scan testing. The interface consists of the following five signals:

- TDI – Serial Data Input
- TMS – Test Mode Select
- TCLK – TAP Clock
- TRST_L – Active Low asynchronous reset for the TAP
- TDO – Serial Data Output

TDI and TMS require external pull-up resistors to float the signals High per the IEEE 1149.1 specification. Pull-ups are recommended on TCK and TDO. For normal operation, TRST_L can be pulled Low, permanently disabling the JTAG interface. If the JTAG interface is used, the TAP controller must be reset as described in Section 5.10.1, *TAP State Machine*, on page 121 and returned to a logic High.

### 5.10.1     TAP State Machine

The TAP signals drive a TAP controller, which implements the 16-state state machine specified by the IEEE 1149.1 specification. Following power-up, the TAP controller must be reset by one of following two mechanisms:

- Asynchronous reset
- Synchronous reset

Asynchronous reset is achieved by pulsing or holding TRST_L Low. Synchronous reset is achieved by clocking TCLK with five clock pulses while TMS is held or floats High. This ensures that the boundary scan cells do not block the pin to core connections in the IXF1104 MAC.

## 5.10.2 Instruction Register and Supported Instructions

The instruction register is a 4-bit register that enacts the boundary scan instructions. After the state machine resets, the default instruction is IDCODE. The decode logic in the TAP controller selects the appropriate data register and configures the boundary scan cells for the current instruction. Table 38 shows the supported boundary-scan instructions.

**Table 38    Instruction Register Description**

| Instruction | Code | Description | Data Register |
|---|---|---|---|
| BYPASS | 1111 | 1-bit Bypass | Bypass |
| EXTEST | 0000 | External Test | Boundary Scan |
| SAMPLE | 0001 | Sample Boundary | Boundary Scan |
| IDCODE | 0110 | ID Code Inspection | ID |
| HIGHZ | 0101 | Float Boundary | Bypass |
| CLAMP | 0111 | Clamp Boundary | Bypass |

## 5.10.3 ID Register

The ID register is a 32-bit register. The IDCODE instruction connects this register between TDI and TDO. See Table 112, *JTAG ID ($0x50C)*, on page 185 for detailed information.

## 5.10.4 Boundary Scan Register

The Boundary Scan register is a shift register made up of all the boundary scan cells associated with the device signals. The number, type, and order of the boundary scan cells are specified in the IXF1104 MAC BSDL file. The EXTEST and SAMPLE instructions connect this register between TDI and TDO.

## 5.10.5 Bypass Register

The Bypass register is a 1-bit register that bypasses the IXF1104 MAC to reduce the JTAG chain length when accessing other devices on the chain besides the IXF1104 MAC. The BYPASS, HIGHZ, and CLAMP instructions connect this register between TDI and TDO.

## 5.11 Loopback Modes

The IXF1104 MAC provides two loopback modes for device diagnostic testing when it has been integrated into a user system. A line-side loopback allows the line-side receive interface to be looped back to the transmit line-side interface. A SPI3 loopback mode allows the SPI3 transmit interface to be looped back to the SPI3 receive interface.

The IXF1104 MAC line-side and SPI3 loopback modes are effective diagnostic tools for validation of system level connectivity and interface compatibility.

In loopback-mode operation, the data path is internally redirected to allow for the data flow return path. Redirection requires the data path to circumvent resources that are required during normal traffic flow. For example, while operating in SPI3 loopback mode, the data path does not pass through the MAC or TX FIFO and those resource features are not used. The result is a possible degradation of throughput performance and statistical data accuracy. Cortina recommends that loopback modes be used for diagnostic purposes only.

### 5.11.1    SPI3 Interface Loopback

To provide a diagnostic loopback feature on the SPI3 interface, it is possible to configure the IXF1104 MAC to loop back any data written to the IXF1104 MAC through the SPI3 transmit interface back to the SPI3 receive interface. This is accomplished using the data path shown in Figure 32.

**Note:**        Loopback packets also appear on the line side TX interface.

**Figure 32      SPI3 Interface Loopback Path**
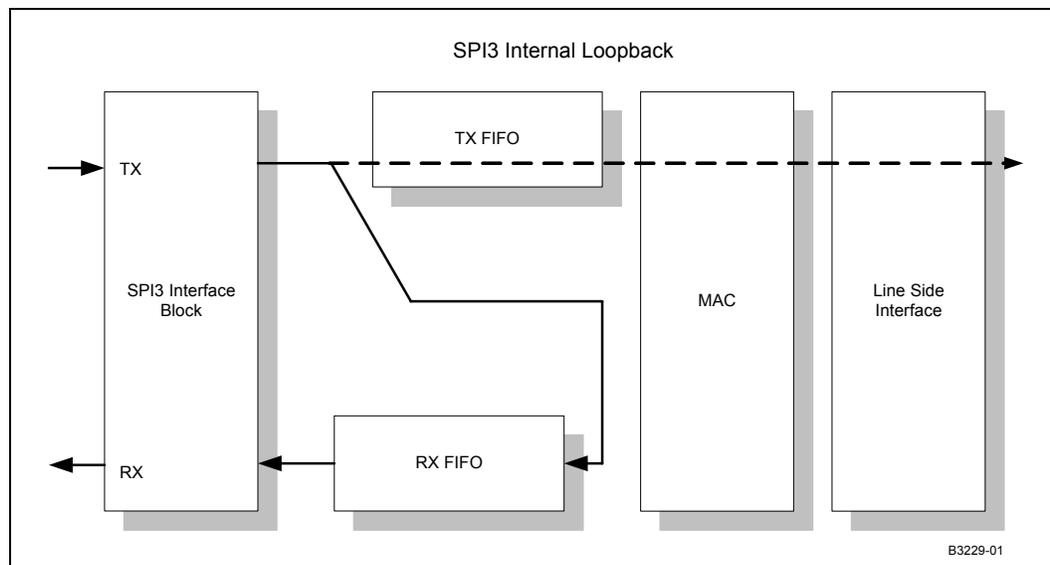


**Note:**        There is a restriction when using this loopback mode. At least one clock cycle is required between a TEOP assertion and a TSOP assertion. This is required when the pre-pend feature of the receive FIFO is enabled to allow the addition of the extra two bytes to the data sent on the transmit interface. Where the pre-pend feature has not been enabled, data can be sent back-to-back on the transmit SPI3 interface with TSOP following TEOP on the next cycle.

To configure the IXF1104 MAC to use the SPI3 loopback mode, the *RX FIFO SPI3 Loopback Enable for Ports 0 - 3 ($0x5B2)* must be configured. Each IXF1104 MAC port has a unique bit in this register designated to control loopback. It is possible to have individual ports in a loopback mode while other ports continue to operate in a normal mode.

### 5.11.2    Line Side Interface Loopback

To provide a diagnostic loopback feature on the line-side interfaces, the IXF1104 MAC can be configured to loop back any data received by the IXF1104 MAC through one of the line interfaces back to the corresponding transmit line interface. This is done by using the data path shown in Figure 33. The line-side interface can be either SerDes, RGMII or GMII. Please note that it is not possible to loop one line-side interface back to a different one (for example, Rx SerDes looped back to transmit RGMII).

**Figure 33   Line Side Interface Loopback Path**
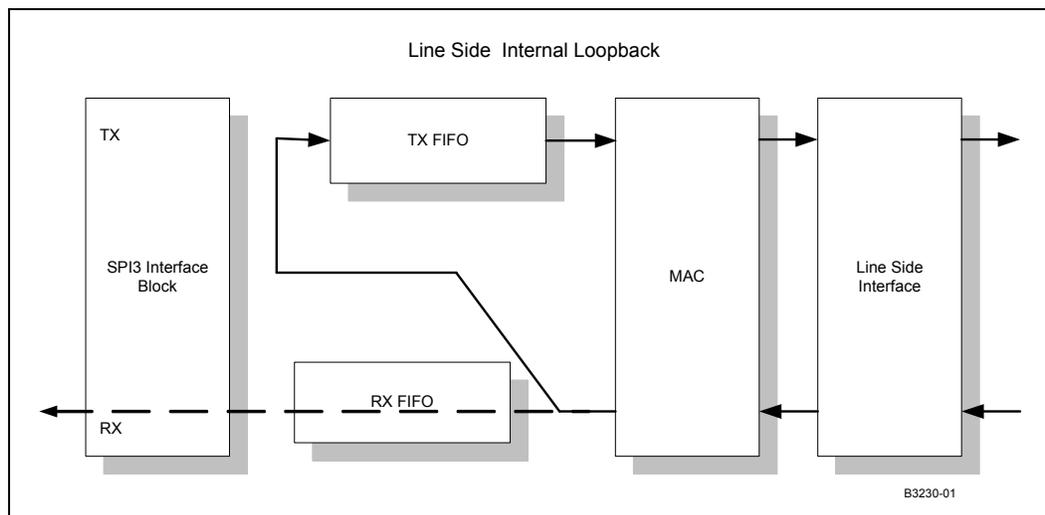


When the IXF1104 MAC is configured in this loopback mode, all of the MAC functions and features are available, including flow control and pause-packet generation.

To configure the IXF1104 MAC to use the line-side loopback mode, the *Loop RX Data to TX FIFO (Line-Side Loopback) Ports 0 - 3 ($0x61F)* must be configured. Each IXF1104 MAC port has a unique bit in this register designated to control the loopback. It is possible to have individual ports in a loopback mode while other ports continue to operate in a normal mode.

**Note:**   Line side interface loopback packets also appear at the SPI3 interface.

## 5.12   Clocks

The IXF1104 MAC system interface has several reference clocks, including the following:

- SPI3 data path input clocks
- RGMII input and output clocks
- MDIO output clock
- JTAG input clock
- $I^2C$ clock
- LED output clock.

This section details the unique clock source requirements.

### 5.12.1   System Interface Reference Clocks

The following system interface clock is required by the IXF1104 MAC:

- CLK125

### 5.12.1.1 CLK125

The system interface clock, which supplies the clock to the majority of the internal circuitry, is the 125 MHz clock. The source of this clock must meet the following specifications:

- 2.5 V CMOS drive
- +/- 50 ppm
- Maximum duty cycle distortion 40/60

## 5.12.2 SPI3 Receive and Transmit Clocks

The IXF1104 MAC transmit clock requirements include the following:

- 3.3 V LVTTL drive
- +/- 50 ppm
- Maximum frequency of 133 MHz in MPHY mode
- Maximum frequency of 125 MHz in SPHY mode
- Maximum duty cycle distortion 45/55

The IXF1104 MAC meets the following specifications for the receive clock:

- 3.3 V LVTTL drive
- +/- 50 ppm
- Maximum frequency of 133 MHz in MPHY mode
- Maximum frequency of 125 MHz in SPHY mode
- Maximum duty cycle distortion 45/55

## 5.12.3 RGMII Clocks

The RGMII interface is governed by the Hewlett-Packard* 1.2a specification. The IXF1104 MAC compliant to this specification with the following:

- 2.5 V CMOS drive
- Maximum duty cycle distortion 40/60
- +/- 100 ppm
- 125 MHz for 1000 Mbps, 25 MHz for 100 Mbps and 2.5 MHz for 10 Mbps

## 5.12.4 MDC Clock

The IXF1104 MAC supports the IEEE 802.3 MII Management Interface, also known as the Management Data Input/Output (MDIO) Interface. The IXF1104 MAC meets the following specifications for this clock:

- 2.5 V CMOS drive
- 2.5/18 MHz operation (selectable by the MDC speed bit in the *MDIO Control ($0x683)*)
- 50/50 duty cycle for 2.5 MHz operation
- 43/57 duty cycle for 18 MHz operation

### 5.12.5 JTAG Clock

The IXF1104 MAC supports JTAG. The source of this clock must meet the following specifications:

- 2.5 V CMOS drive
- Maximum clock frequency 11 MHz
- Maximum duty cycle distortion 40/60

### 5.12.6 I$^2$C Clock

The IXF1104 MAC supports a single-output I$^2$C clock to support all ten Optical Module interfaces. The IXF1104 MAC meets the following specifications for this clock:

- 2.5 V CMOS drive
- Maximum clock frequency of 100 KHz

### 5.12.7 LED Clock

The IXF1104 MAC supports a serial LED data stream and meets the following specifications for this clock:

- 2.5 V CMOS drive
- Maximum frequency of 720 Hz
- Maximum duty cycle distortion 50/50

# 6.0 Applications

## 6.1 Change Port Mode Initialization Sequence

Use the change port mode initialization sequence after power-up and anytime a port is configured into or switching between fiber or copper mode, switching to/from RGMII and GMII modes, or switching speeds and duplex in RGMII mode.

The following sequence applies to all four ports and can be done simultaneously for all ports or as a subset of the ports.

1. Place the MAC in reset for the port(s) which require a change by asserting (set to 1) the Section 106, *MAC Soft Reset ($0x505)*.

2. Place the TX FIFO in reset for the port(s) which require a change by asserting (set to 1) the Section 137, *TX FIFO Port Reset ($0x620)*.

3. Disable the port(s) which require change by de-asserting (set to 0) the appropriate bits in the Section 103, *Port Enable ($0x500)*.

4. Wait 1 $\mu$s.

5. De-assert (set to 0) Section 152, *Clock and Interface Mode Change Enable Ports 0 - 3 ($0x794)* for the ports being changed.

6. Set the speed, mode, and duplex as follows for the ports being changed:

    a. Copper mode:

    Select copper mode for the Section 104, *Interface Mode ($0x501)* ports.
    Set the per-port Section 82, *MAC IF Mode and RGMII Speed ($ Port_Index + 0x10)* to the appropriate speed and RGMII/GMII interface setting.
    Set the per-port Section 71, *Desired Duplex ($ Port_Index + 0x02)*.

    Note: Half-duplex is supported only when RGMII 10 Mbps or 100 Mbps is selected in the Section 82, *MAC IF Mode and RGMII Speed ($ Port_Index + 0x10)*.

    b. Fiber mode:
    Select fiber mode by setting the appropriate bit to 0 in the Section 104, *Interface Mode ($0x501)* ports.

7. Assert (set to 1) Section 152, *Clock and Interface Mode Change Enable Ports 0 - 3 ($0x794)* for the ports being changed.

8. Wait 1 $\mu$s.

9. De-assert (set to 0) Section 106, *MAC Soft Reset ($0x505)* for the ports being changed.

10. De-assert (set to 0) Section 137, *TX FIFO Port Reset ($0x620)* for the ports being changed.

11. Wait 1 to 2 $\mu$s.

12. Set the Section 90, *Diverse Config Write ($ Port_Index + 0x18)* to the appropriate value as follows:

    a. Copper mode:

    Write the reserved bits to the default value.

Enable packet padding and CRC appending on transmitted packets in bits 6 and 7, as needed.

Set bit 5 to 0x0.

b. Fiber Mode:

Write the reserved bits to the default value.

Enable Packet padding and CRC Appending on transmitted packets in bits 6 and 7, as needed.

Set bit 5 to 1 to enable auto-negotiation.

Set bit 5 to 0 to enable forced mode operation.

13. Assert (set to 1) Section 103, *Port Enable ($0x500)*.

14. Wait 1 to 2 μs.

15. Perform additional device configurations, as needed.

## 6.2 Disable and Enable Port Sequences

Cortina recommends the following sequences to disable and enable individual ports, and for dropped links. When a link is dropped, Cortina recommends the port be completely reset and flushed to remove packet fragments that may interfere with the auto-negotiation process on link recovery.

### 6.2.1 Disable Port Sequence

Use the following sequence to disable an individual port:

1. Disable the port using MAC port enable/disable bits [Section 103, *Port Enable ($0x500)* Bits (3-0)].

2. Apply TX FIFO soft reset [Section 137, *TX FIFO Port Reset ($0x620)* Bits(3-0)].

3. Introduce some delay to allow completion of packet transmission (not necessary if link is dropped).

4. Flush TX [Section 83, *Flush TX ($ Port_Index + 0x11)* Bit 0].

5. Apply MAC soft reset [Section 106, *MAC Soft Reset ($0x505)* Bits(3-0)].

6. Apply RX FIFO soft reset [Section 122, *RX FIFO Port Reset ($0x59E)* Bits(3:0)].

### 6.2.2 Enable Port Sequence

Use the following sequence to enable an individual port:

1. Enable the port(s) using MAC port enable/disable bits [Section 103, *Port Enable ($0x500)* Bits (3-0)].

2. Disable TX FIFO soft reset [Section 137, *TX FIFO Port Reset ($0x620)* Bits(3-0)].

3. Reset flush TX [Section 83, *Flush TX ($ Port_Index + 0x11)* Bit 0].

4. Disable MAC soft reset [Section 106, *MAC Soft Reset ($0x505)* Bits(3-0)].

5. Disable RX FIFO soft reset [Section 122, *RX FIFO Port Reset ($0x59E)* Bits(3:0)].

# 7.0 Electrical Specifications

Table 39 through Table 58, *LED Interface AC Timing Parameters*, on page 147 and Figure 34, *SPI3 Receive Interface Timing*, on page 133 through Figure 51, *LED AC Interface Timing*, on page 147 represent the target specifications of the following IXF1104 MAC interfaces:

- — SPI3
- — JTAG
- — MDIO
- — Pause Control
- — CPU
- — LED
- — System
- — GMII and RGMII
- — SerDes
- — Optical Module

These specifications are not guaranteed and are subject to change without notice. Minimum and maximum values listed in Table 41, *DC Specifications*, on page 131 through Table 58, *LED Interface AC Timing Parameters*, on page 147 apply over the recommended operating conditions specified in Table 40.

## Table 39    Absolute Maximum Ratings

| Parameter | | Symbol | Min | Max | Units | Comments |
|---|---|---|---|---|---|---|
| Supply voltage | | VDD | -0.3 | 2.2 | volts | Core digital power |
| | | VDD2, VDD3 | -0.3 | 4.25 | volts | I/O digital power |
| | | VDD4, VDD5 | -0.3 | 4.25 | volts | I/O digital power |
| | | AVDD1P8_1/2 | -0.3 | 2.2 | volts | Analog power |
| | | AVDD2P5_1/2 | -0.3 | 4.25 | volts | Analog power |
| Operating temperature | Ambient | TOPA | -40 | +85 | °C | Copper mode |
| | Ambient | TOPA | 0.0 | +70 | °C | Fiber mode |
| Storage temperature | | TST | -40 | +150 | °C | – |

**Caution:** Exceeding these values may cause permanent damage to the device. Functional operation under these conditions is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

**Table 40     Recommended Operating Conditions**

| Parameter | | Symbol | Min | Typ | Max | Units |
|---|---|---|---|---|---|---|
| Recommended supply voltage | | VDD | 1.65 | – | 1.95 | Volts |
| | | VDD2, VDD3 | 3.0 | – | 3.6 | Volts |
| | | VDD4, VDD5 | 2.3 | – | 2.7 | Volts |
| | | AVDD1P8_1 AVDD1P8_2 | 1.65 | – | 1.95 | Volts |
| | | AVDD2P5_1 AVDD2P5_2 | 2.3 | – | 2.7 | Volts |
| Operating Current | SerDes Operation Transmitting and receiving in 1000 Mbps mode | VDD AVDD1P8_1 AVDD1P8_2 | – | 0.780 | – | Amps |
| | | VDD4 VDD5 AVDD2P5_1 AVDD2P5_2 | – | 0.050 | – | Amps |
| | | VDD2, VDD3 | – | 0.246 | – | Amps |
| Operating Current | RGMII Operation Transmitting and receiving in 1000 Mbps mode | VDD AVDD1P8_1 AVDD1P8_2 | – | 0.757 | – | Amps |
| | | VDD4 VDD5 AVDD2P5_1 AVDD2P5_2 | – | 0.224 | – | Amps |
| | | VDD2, VDD3 | – | 0.208 | 0.235 | Amps |
| Recommended operating temperature | Ambient | TOPA | 0 | – | 70 | °C |
| | Case with heat sink | TOPC-HS | 0 | – | 122 | °C |
| | Case without heat sink | TOPC-NHS | 0 | – | 121 | °C |
| Power consumption | SerDes Operation Transmitting and receiving in 1000 Mbps mode | – | – | 2.23 | 2.72 | Watts |
| | RGMII Operation Transmitting and receiving in 1000 Mbps mode | – | – | 2.84 | 3.4 | Watts |

## 7.1     DC Specifications

The IXF1104 MAC supports the following I/O buffer types:

- 2.5 V CMOS
- 3.3 V LVTTL
- SerDes

See Section 5.1.7, *Packet Buffer Dimensions*, on page 79 for additional information regarding I/O buffer types. The related driver characteristics are described in this section.

**Caution:** IXF1104 MAC input signals are not 5 V tolerant. Devices driving the IXF1104 MAC must provide 3.3 V signal levels or use level-shifting buffers to provide 3.3 V-compatible levels. Otherwise, damage to the IXF1104 MAC will occur.

**Table 41    DC Specifications**

| Parameter | Symbol | Min | Typ | Max | Units | Comments |
|---|---|---|---|---|---|---|
| **2.5 V CMOS I/O Cells** | | | | | | |
| Input High voltage | VIH | 1.7 | – | – | V | 2.5 V I/Os |
| Input low voltage | VIL | – | – | 0.7 | V | 2.5 V I/Os |
| Output High voltage | VOH | 2.0 | – | – | V | 2.5 V I/Os |
| Output low voltage | VOL | – | – | 0.4 | V | 2.5 V I/Os |
| **3.3 V I/O Cells** | | | | | | |
| Input High voltage | VIH | 1.7 | – | – | V | 3.3 V LVTTL I/Os |
| Input low voltage | VIL | – | – | 0.7 | V | 3.3 V LVTTL I/Os |
| Output High voltage | VOH | 2.4 | – | – | V | 3.3 V LVTTL I/Os |
| Output low voltage | VOL | – | – | 0.4 | V | 3.3 V LVTTL I/Os |

**Table 42    SerDes Transmit Characteristics (Sheet 1 of 2)**

| Parameter | Symbol | Normalized Power Drive Settings[1] | Min | Typ | Max | Units | Comments |
|---|---|---|---|---|---|---|---|
| Transmit differential signal level | TxDfPP | 0.50 | 180 | 230 | 325 | mVpp diff | AVDD1P8_2 terminated to 1.8V; Rload = 50 Ω |
| | | 1.00 | 350 | 440 | 700 | | |
| | | 1.33 | 425 | 580 | 900 | | |
| | | 2.00 | 600 | 770 | 1050 | | |
| Transmit common mode voltage range | TxCMV | 0.50 | 1300 | 1600 | 1940 | mV | AVDD1P8_2 terminated to 1.8V; RLoad = 50 ohms; FIR coeffs = 0 |
| | | 1.00 | 1000 | 1400 | 1870 | | |
| | | 1.33 | 800 | 1300 | 1825 | | |
| | | 2.00 | 700 | 1100 | 1760 | | |
| Differential signal rise/fall time | Diff rise/fall | 1.00 | 60 | 96 | 132 | ps | Rload = 50 Ω; 20% to 80% max |
| Differential output impedance | TxDiffZ | – | 60 | 105 | 150 | Ω diff | Nominal value = 100 Ω differential |
| Receiver differential voltage requirement at center of receive eye | RxDiffV | – | 200 | – | – | mVp-p diff | – |

1.  Refer to Section 5.6.2.2, *Transmitter Programmable Driver-Power Levels*, on page 103.

**Table 42    SerDes Transmit Characteristics (Sheet 2 of 2)**

| Parameter | Symbol | Normalized Power Drive Settings[1] | Min | Typ | Max | Units | Comments |
|---|---|---|---|---|---|---|---|
| Receiver common mode voltage range | RxCMV | – | 900 | 1275 | 1650 | mV | – |
| Receiver termination impedance | RxZ | – | 40 | 51 | 62.5 | Ω | – |
| Signal detect level | RxSigDet | – | 50 | 125 | 200 | mVp-pdiff | – |
| 1.   Refer to Section 5.6.2.2, *Transmitter Programmable Driver-Power Levels*, on page 103. | | | | | | | |

**Table 43    SerDes Receive Characteristics**

| Parameter | Symbol | Normalized Power Drive Settings | Min | Typ | Max | Units | Comments |
|---|---|---|---|---|---|---|---|
| Receiver differential voltage requirement at center of receive eye | RxDiffV | – | 200 | – | – | mVp-p diff | – |
| Receiver common mode voltage range | RxCMV | – | 900 | 1275 | 1650 | mV | – |
| Receiver termination impedance | RxZ | – | 40 | 51 | 62.5 | Ω | – |
| Signal detect level | RxSigDet | – | 50 | 125 | 200 | mVp-pdiff | – |

## 7.1.1    Undershoot / Overshoot Specifications

The overshoot figures given in this section represent the maximum voltage that can be applied without affecting the reliability of the device (see Table 44).

**Caution:**    If these limits are exceeded, damage to the device will occur.

**Table 44    Undershoot / Overshoot Limits**

| Pin Type | Undershoot | Overshoot |
|---|---|---|
| 2.5 V CMOS | -0.60 V | 3.9 V |
| 3.3 V LVTTL | -0.60 V | 3.9 V |

## 7.1.2    RGMII Electrical Characteristics

The RGMII signals (including MDIO/MDC) are based on 2.5V CMOS interface voltages, as defined by JEDEC EIA/JESD8-5 (see Table 45).

**Table 45    RGMII Power**

| Symbol | Parameter | Conditions | Min | Max | Units |
|--------|-----------|------------|-----|-----|-------|
| $V_{OH}$ | Output High Voltage | $I_{OH}$ = -1.0 MA; $V_{DD}$ = MIN | 2.0 | VDD +.3 | V |
| $V_{OL}$ | Output Low Voltage | $I_{OL}$ = 1.0 MA; $V_{DD}$ = MIN | GND -.3 | 0.40 | V |
| $V_{IH}$ | Input High Voltage | $V_{IH}$ > $V_{IH\_MIN}$; $V_{DD}$ = MIN | – | VDD +.3 | V |
| $V_{IL}$ | Input Low Voltage | $V_{IL}$ < $V_{IL\_MAX}$; $V_{DD}$ = MIN | – | .70 | V |
| $I_{IH}$ | Input High Current | $V_{DD}$ = MAX; $V_{IN}$ = 2.5V | – | 15 | µA |
| $I_{IL}$ | Input Low Current | $V_{DD}$ = MAX; $V_{IN}$ = 0.4V | -15 | – | µA |

## 7.2        SPI3 AC Timing Specifications

### 7.2.1      Receive Interface Timing

Figure 34 and Table 46 illustrate and provide SPI3 receive interface timing information.

**Figure 34    SPI3 Receive Interface Timing**

**Table 46      SPI3 Receive Interface Signal Parameters**

| Symbol | Parameter | Min | Max | Units |
|---|---|---|---|---|
| – | RFCLK frequency | 90 | 133 | MHz |
| – | RFCLK duty cycle | 45 | 55 | % |
| Tsrenb | RENB setup time to RFCLK | 1.8 | – | ns |
| Threnb | RENB hold time to RFCLK | 0.5 | – | ns |
| TPrdat | RFCLK High to RDAT valid | 1.5 | 3.7 | ns |
| TPrprty | RFCLK High to RPRTY valid | 1.5 | 3.7 | ns |
| TPrsop | RFCLK High to RSOP valid | 1.5 | 3.7 | ns |
| TPreop | RFCLK High to REOP valid | 1.5 | 3.7 | ns |
| TPrmod | RFCLK High to RMOD valid | 1.5 | 3.7 | ns |
| TPrerr | RFCLK High to RERR valid | 1.5 | 3.7 | ns |
| TPrval | RFCLK High to RVAL valid | 1.5 | 3.7 | ns |
| TPrsx | RFCLK High to RSX valid | 1.5 | 3.7 | ns |

*Notes:*  Receive I/O Timing
1.  When a setup time is specified between an input and a clock, the setup time is the time in nanoseconds from the 1.4-volt point of the input to the 1.4-volt point of the clock.
2.  When a hold time is specified between an input and a clock, the hold time is the time in nanoseconds from the 1.4-volt point of the clock to the 1.4-volt point of the input.
3.  Output propagation time is the time in nanoseconds from the 1.4-volt point of the reference signal to the 1.4-volt point of the output.
4.  Maximum propagation delays are measured with a 30 pF load when operating OIF-SPI3 standard 104 MHz. Over-clocked rates of 125 MHz or higher are measured using a load of 20 pF.

## 7.2.2 Transmit Interface Timing

Figure 35 and Table 47 illustrate and provide SPI3 transmit interface timing information.

**Figure 35   SPI3 Transmit Interface Timing**
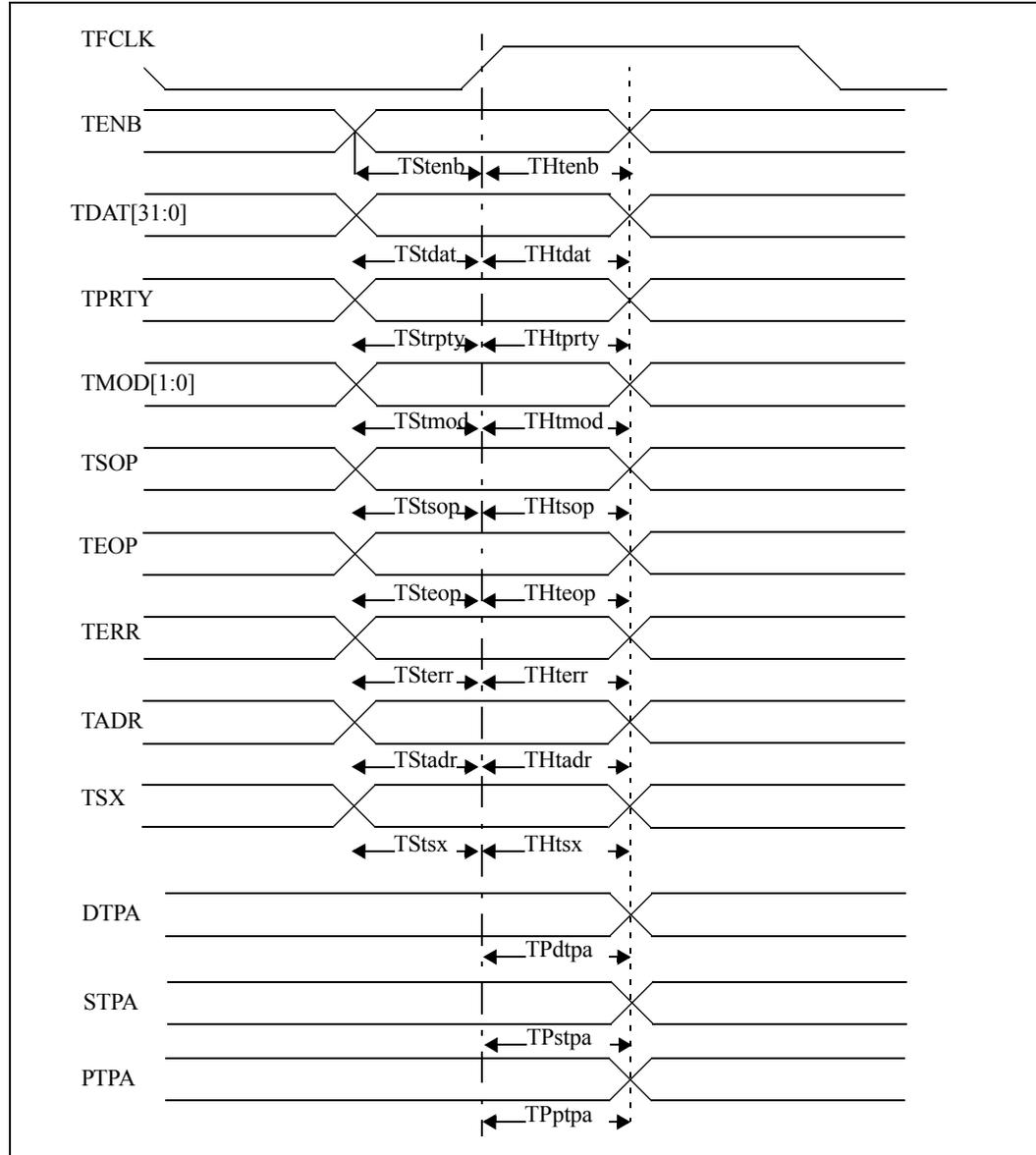
**Table 47      SPI3 Transmit Interface Signal Parameters**

| Symbol | Parameter | Min | Max | Units |
|--------|-----------|-----|-----|-------|
| – | TFCLK frequency | – | 133 | MHz |
| – | TFCLK duty cycle | 45 | 55 | % |
| TStenb | TENB setup time to TFCLK | 1.8 | – | ns |
| THtenb | TENB hold time to TFCLK | 0.5 | – | ns |
| TStdat | TDAT[31:0] setup time to TFCLK | 1.8 | – | ns |
| THtdat | TDAT[31:0} hold time to TFCLK | 0.5 | – | ns |
| TStprty | TRPTY setup time to TFCLK | 1.8 | – | ns |
| THtprty | TPRTY hold time to TFCLK | 0.5 | – | ns |
| TStsop | TSOP setup time to TFCLK | 1.8 | – | ns |
| THtsop | TSOP hold time to TFCLK | 0.5 | – | ns |
| TSteop | TEOP setup time to TFCLK | 1.8 | – | ns |
| THteop | TEOP hold time to TFCLK | 0.5 | – | ns |
| TStmod | TMOD setup time to TFCLK | 1.8 | – | ns |
| THtmod | TMOD hold time to TFCLK | 0.5 | – | ns |
| TSterr | TERR setup time to TFCLK | 1.8 | – | ns |
| THterr | TERR hold time to TFCLK | 0.5 | – | ns |
| TStsx | TSX setup time to TFCLK | 1.8 | – | ns |
| THtsx | TSX hold time to TFCLK | 0.5 | – | ns |
| TStadr | TADR setup time to TFCLK | 1.8 | – | ns |
| THtadr | TADR hold time to TFCLK | 0.5 | – | ns |
| TPdtpa | TFCLK High to DTPA valid | 1.5 | 3.7 | ns |
| TPstpa | TFCLK High to STPA valid | 1.5 | 3.7 | ns |
| TPptpa | TFCLK High to PTPA valid | 1.5 | 3.7 | ns |

*Notes:* Transmit I/O Timing:
1.  When a setup time is specified between an input and a clock, the setup time is the time in nanoseconds from the 1.4 V point of the input to the 1.4-volt point of the clock.
2.  When a hold time is specified between an input and clock, the hold time is the time in nanoseconds from the 1.4 V point of the clock to the 1.4-volt point of the input.
3.  Output propagation delay time is the time in nanoseconds from the 1.4 V point of the reference signal to the 1.4 V point of the output.

## 7.3 RGMII AC Timing Specification

Figure 36 and Table 48 provide RGMII interface timing parameters.

**Figure 36    RGMII Interface Timing**



B3251-01
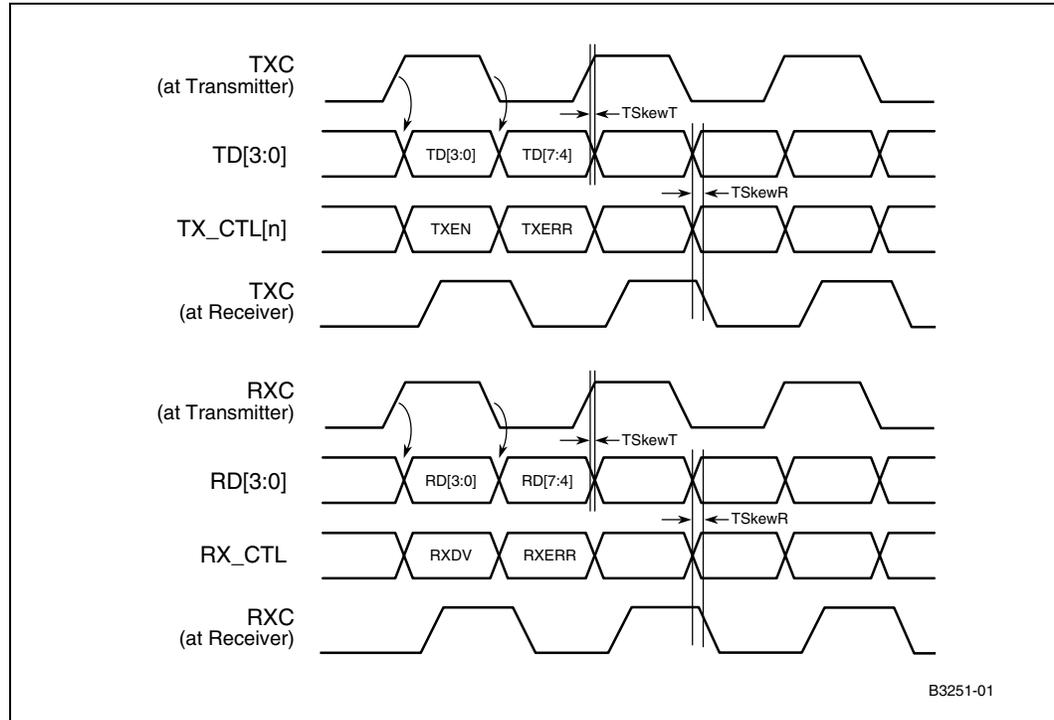
**Table 48    RGMII Interface Timing Parameters**

| Symbol | Parameter | Min | Typ | Max | Unit |
|--------|-----------|-----|-----|-----|------|
| TskewT | Data-to-Clock Output Skew (at Transmitter) | -500 | 0 | 500 | ps |
| TskewR | Data-to-Clock Input Skew (at Receiver)[1] | 1 | – | 2.8 | ns |
| Tcyc | Clock Cycle Duration[2] | 7.2 | 8 | 8.8 | ns |
| Duty_T | Duty Cycle for Gigabit[2] | 45 | 50 | 55 | % |
| Duty_G | Duty Cycle for 10/100T[3] | 40 | 50 | 60 | % |
| Tr/Tf | Rise/Fall Time (20–80%) | – | – | .75 | ns |

1. This implies that PC board design requires clocks to be routed so that an additional trace delay of greater than 1.5 ns is added to the associated clock signal.
2. For 10 Mbps and 100 Mbps Tcyc scales to 400 ns +/– 40 ns and 40 ns +/– 4 ns respectively.
3. Duty cycle may be stretched/shrunk during speed changes or while transitioning to a received packet's clock domain, as long as minimum duty cycle is not violated and stretching occurs for no more than three Tcyc of the lowest speed transitioned between.

## 7.4 GMII AC Timing Specification

### 7.4.1 1000 Base-T Operation

Figure 37 and Figure 38 and Table 49 and Table 50 provide GMII AC timing specifications.

### 7.4.1.1   1000 BASE-T Transmit Interface

**Figure 37   1000BASE-T Transmit Interface Timing**
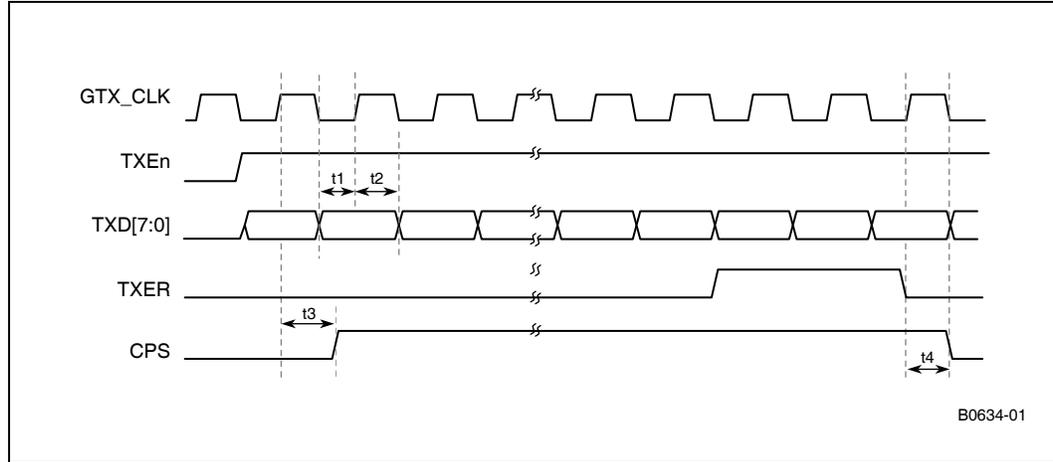


B0634-01

**Table 49   GMII 1000BASE-T Transmit Signal Parameters**

| Symbol | Parameter | Min | Typ[1] | Max | Unit[2] |
|--------|-----------|-----|--------|-----|---------|
| t1 | TXD[7:0], TXEN, TXER Set-up to TXC High | 2.5 | – | – | ns |
| t2 | TXD[7:0], TXEN, TXER Hold from TXC High | 0.5 | – | – | ns |
| t3 | TXEN sampled to CRS asserted | – | – | 16 | BT |
| t4 | TXEN sampled to CRS de-asserted | – | – | 16 | BT |

1. Typical values are at 25$^o$C and are for design aid only; not guaranteed and not subject to production testing.
2. Bit Time (BT) is the duration of one bit as transferred to/from the PHY and is the reciprocal of bit rate. BT for 1000BASE-T = $10^{-9}$ or 1 ns.

### 7.4.1.2   1000BASE-T Receive Interface

**Figure 38   1000BASE-T Receive Interface Timing**
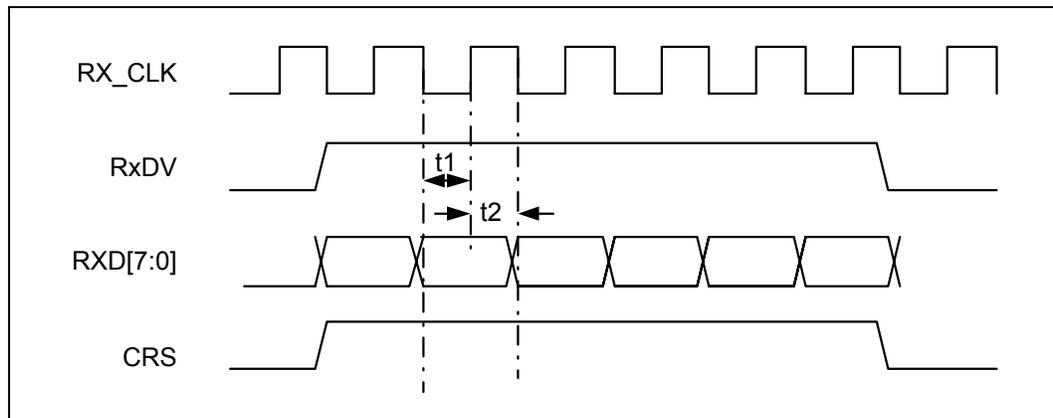
**Table 50     GMII 1000BASE-T Receive Signal Parameters**

| Symbol | Parameter | Min | Typ[1] | Max | Unit[2] |
|--------|-----------|-----|--------|-----|---------|
| t1 | RXD[7:0], RX_DV, RXER Setup to Rx_CLK High | 2.0 | – | – | ns |
| t2 | RXD[7:0], RX_DV, RXER Hold after Rx_CLK High | 0.0 | – | – | ns |

1. Typical values are at 25°C and are for design aid only; not guaranteed and not subject to production testing.
2. Bit Time (BT) is the duration of one bit as transferred to/from the PHY and is the reciprocal of bit rate. BT for 1000BASE-T = $10^{-9}$ or 1 ns.

## 7.5     SerDes AC Timing Specification

**Figure 39     SerDes Timing Diagram**
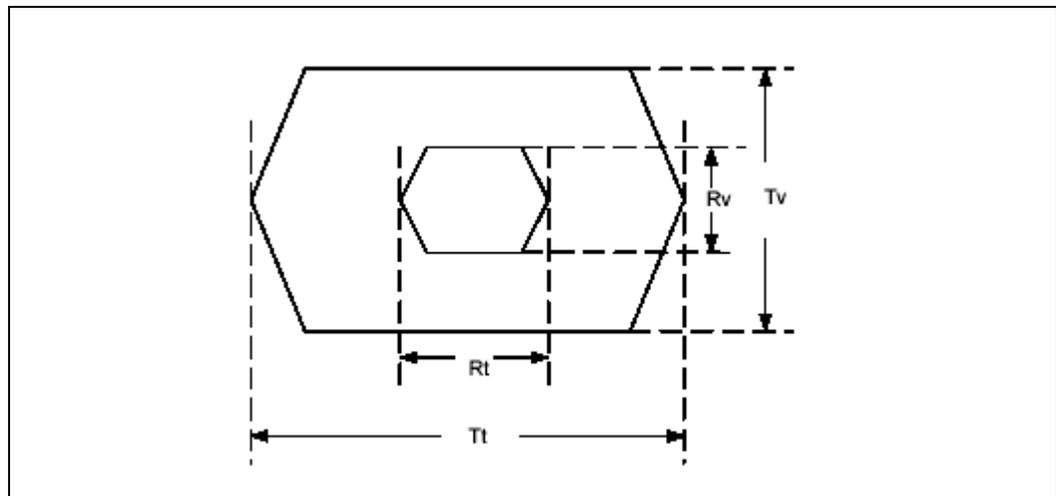


**Table 51     SerDes Timing Parameters**

| Symbol | Parameter | Min | Max | Units |
|--------|-----------|-----|-----|-------|
| Tt | Transmit eye width | 800 | – | pS |
| Rt | Receiver eye width | 280 | – | pS |
| Tv | Transmit amplitude | 1000 | – | mV |
| Rv | Receiver amplitude | 200 | – | mV |

## 7.6     MDIO AC Timing Specification

The MDIO Interface on the IXF1104 MAC can operate in two modes – low-speed and high-speed. In low-speed mode, the MDC clock signal operates at a frequency of 2.5 MHz. In high-speed mode, the MDC clock signal operates at a frequency of 18 MHz. (See Figure 40 through Figure 43 and Table 52.)

## 7.6.1    MDC High-Speed Operation Timing

**Figure 40    MDC High-Speed Operation Timing**
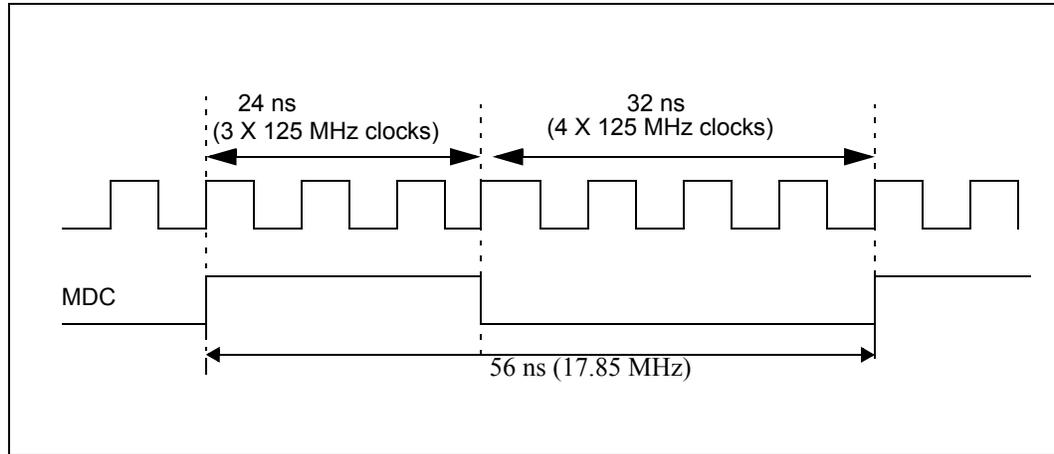


## 7.6.2    MDC Low-Speed Operation Timing

**Figure 41    MDC Low-Speed Operation Timing**
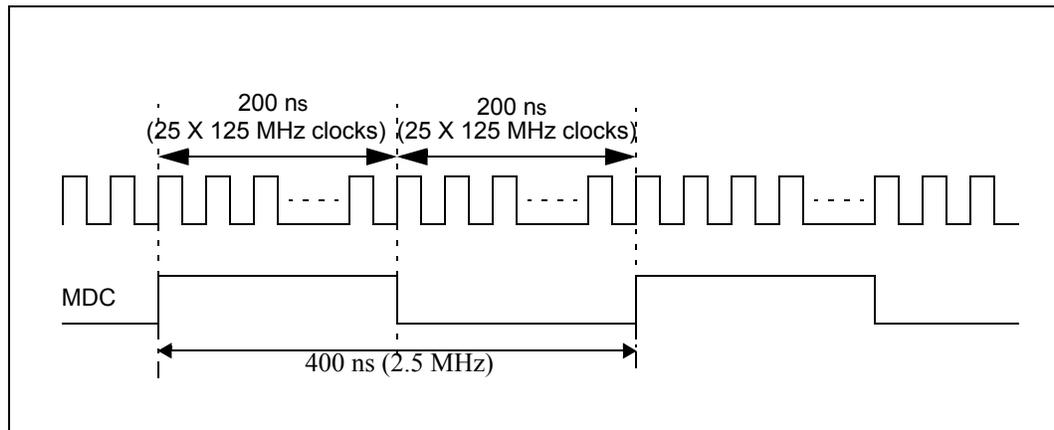
## 7.6.3 MDIO AC Timing

**Figure 42    MDIO Write Timing Diagram**
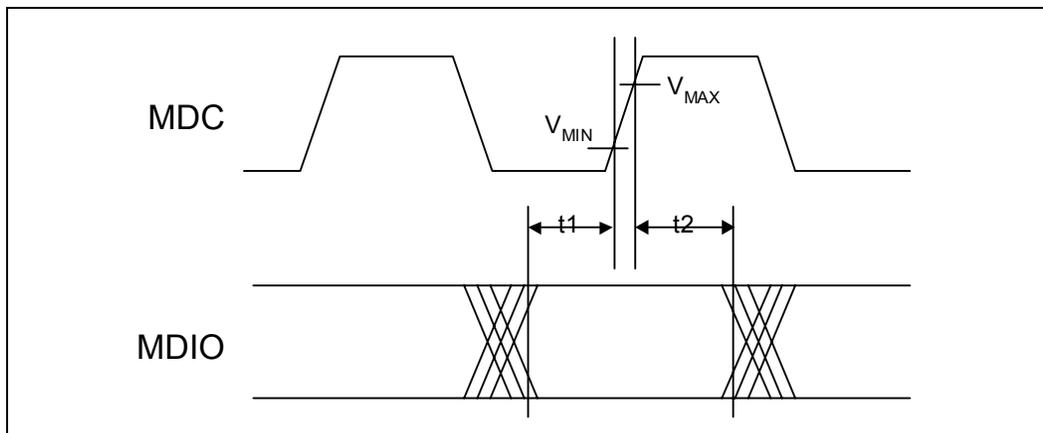


**Figure 43    MDIO Read Timing Diagram**
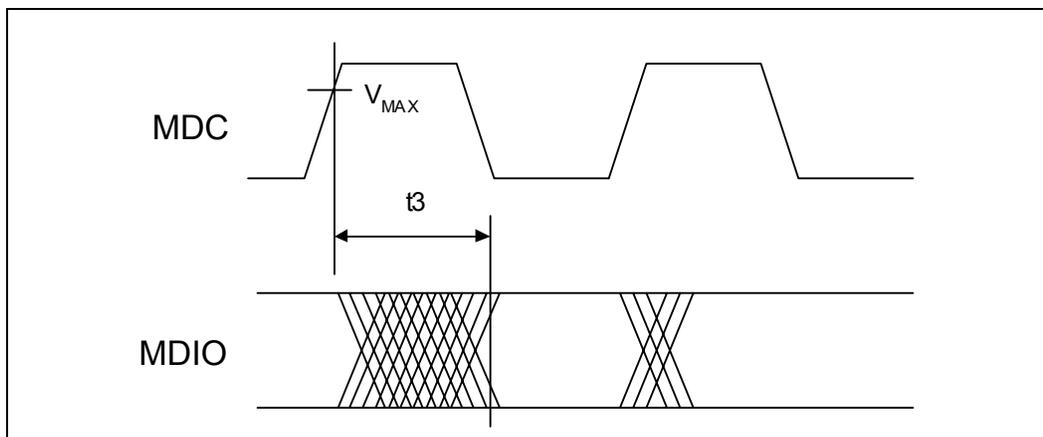


**Table 52    MDIO Timing Parameters**

| Parameter | Symbol | Min | Typ[1] | Max | Units | Test Conditions |
|---|---|---|---|---|---|---|
| MDIO Setup before MDC. | t1 | 10 | – | – | ns | MDC = 17.8 MHz |
| | | 10 | – | – | ns | MDC = 2.5 MHz |
| MDIO Hold after MDC. | t2 | 10 | – | – | ns | MDC = 17.8 MHz |
| | | 10 | – | – | ns | MDC = 2.5 MHz |
| MDC to MDIO Output delay | t3 | 0 | – | 42 | ns | MDC = 17.8 MHz |
| | | 0 | – | 300 | ns | MDC = 2.5 MHz |
| 1.  Typical values are at 25 $^{o}$C and are for design aid only; not guaranteed and not subject to production testing. | | | | | | |

## 7.7 Optical Module and I$^2$C AC Timing Specification

### 7.7.1 I$^2$C Interface Timing

Figure 44 and Figure 45 illustrate bus timing and write cycle, and Table 53 shows the I$^2$C Interface AC timing characteristics.

**Figure 44    Bus Timing Diagram**
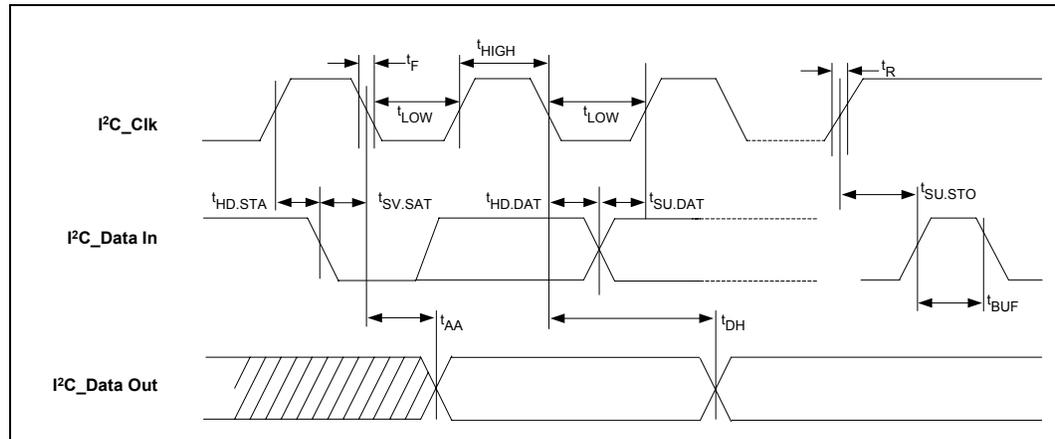


**Figure 45    Write Cycle Diagram**
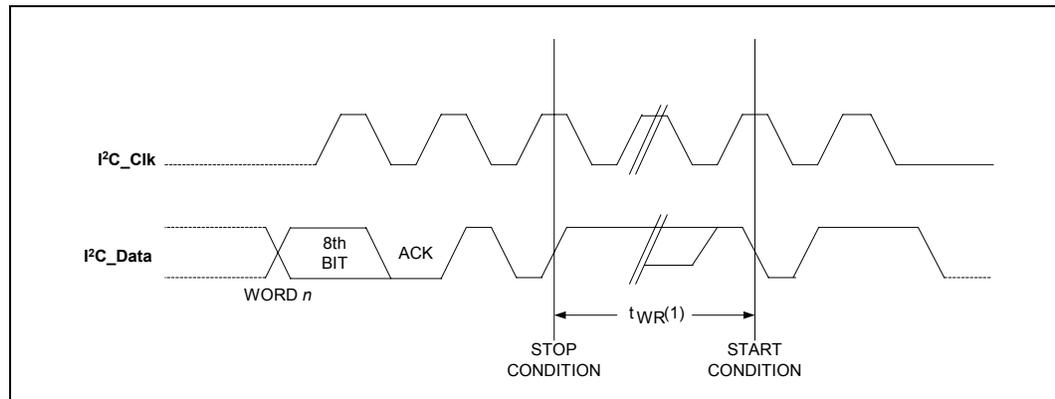


**Table 53    I$^2$C AC Timing Characteristics (Sheet 1 of 2)**

| Symbol | Parameter | Min | Max | Units |
|--------|-----------|-----|-----|-------|
| $f_{SCL}$ | Clock frequency, SCL | - | 100 | kHz |
| $t_{LOW}$ | Clock pulse width low | 4.7 | | µs |
| $t_{HIGH}$ | Clock pulse width High | 4.0 | | µs |
| $t_I$ | Noise suppression | | 100 | µs |
| $t_{AA}$ | Clock low to data valid out | 0.1 | 4.5 | µs |
| $t_{BUF}$ | Time the bus must be free before a new transmission starts | 4.7 | - | µs |
| $t_{HD.STA}$ | Start hold time | 4.0 | - | µs |

**Table 53    I$^2$C AC Timing Characteristics (Sheet 2 of 2)**

| Symbol | Parameter | Min | Max | Units |
|--------|-----------|-----|-----|-------|
| $t_{SU.STA}$ | Start setup time | 4.7 | – | µs |
| $t_{HD.DAT}$ | Data in hold time | 0 | – | µs |
| $t_{SU.DAT}$ | Data in setup time | 200 | – | ns |
| $t_R$ | Inputs rise time | – | 1.0 | µs |
| $t_F$ | Inputs fall time | – | 300 | ns |
| $t_{SU.STO}$ | Stop setup time | 4.7 | – | µs |
| $t_{DH}$ | Data out hold time | 100 | – | ns |
| $t_{WR}$ | Write cycle time | – | 10 | ms |

# 7.8    CPU AC Timing Specification

## 7.8.1    CPU Interface Read Cycle AC Timing

Figure 46, Figure 47, and Table 54 illustrate the CPU interface read and write cycle AC timing.

**Figure 46    CPU Interface Read Cycle AC Timing**
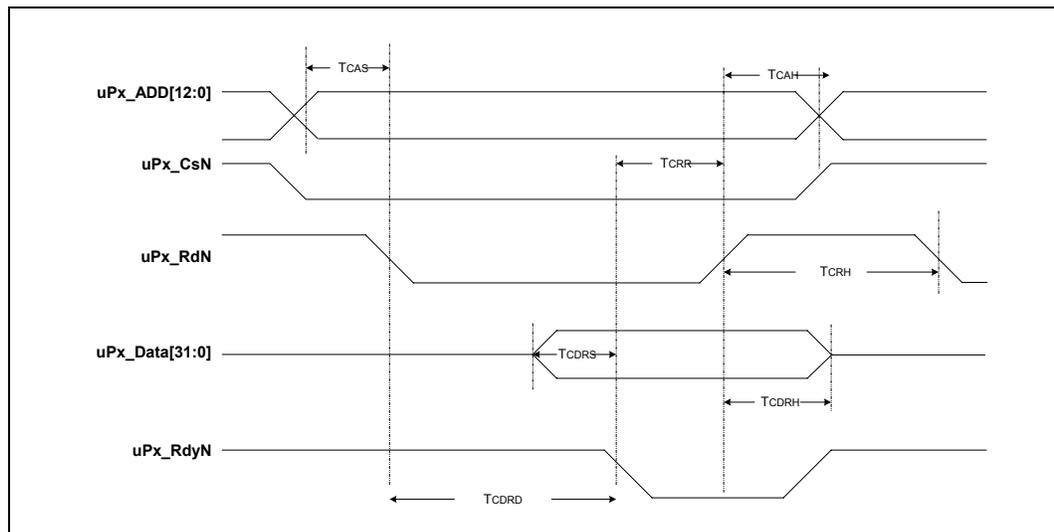
## 7.8.2 CPU Interface Write Cycle AC Timing

**Figure 47 CPU Interface Write Cycle AC Timing**
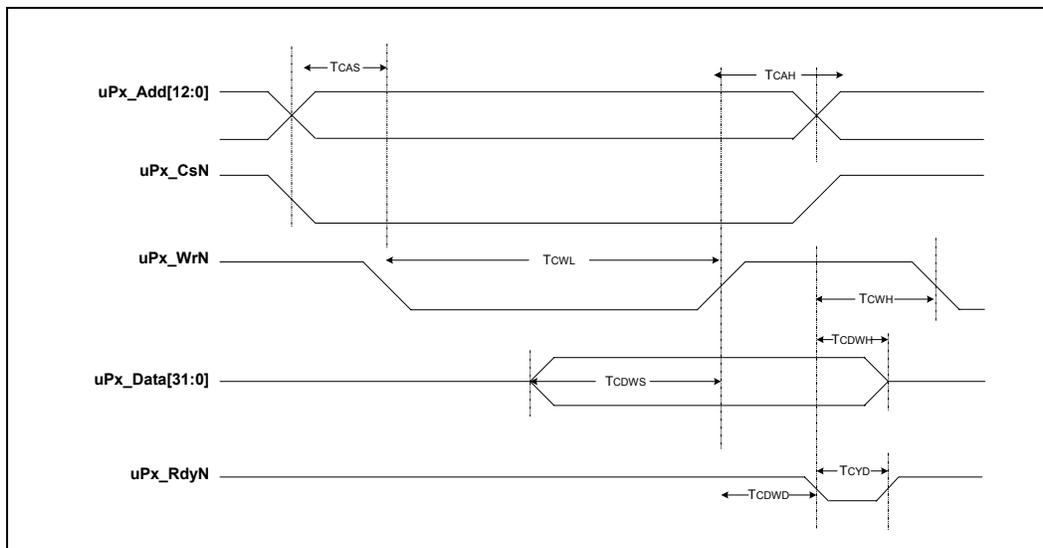


**Table 54 CPU Interface Write Cycle AC Signal Parameters**

| Symbol | Parameter | Min | Max |
|--------|-----------|-----|-----|
| Tcas | Address, chip select setup time | 5 ns | – |
| Tcah | Address, chip select hold time | 10 ns | – |
| Tcrr | Ready assertion to read de-assertion | 10 ns | – |
| Tcrh | Read High width | 24 ns | – |
| Tcdrs | Read data setup time to ready assertion | 10 ns | – |
| Tcdrh | Read data hold time after read de-assertion | 8 ns | 32 ns |
| Tcdrd | Read data driving delay | 24 ns | 355 ns |
| Tcwl | Write assertion width | 40 ns | – |
| Tcwh | Ready assertion to write assertion | 16 ns | – |
| Tcdws | Write data setup to write de-assertion | 10 ns | – |
| Tcdwh | Write data hold time after ready assertion | 5 ns | – |
| Tcdwd | Write data sampling delay | 8 ns | 32 ns |
| Tcyd | Ready width in write cycle | 24 ns | 40 ns |

## 7.9 Transmit Pause Control AC Timing Specification

Figure 48 and Table 55 show the pause control AC timing specifications. The Pause Control interface operates as an asynchronous interface relative to the main system clock (CLK125). There is, however, a relationship between the TXPAUSEADD bus and the strobe signal (TXPAUSEFR).

**Figure 48      Pause Control Interface Timing**
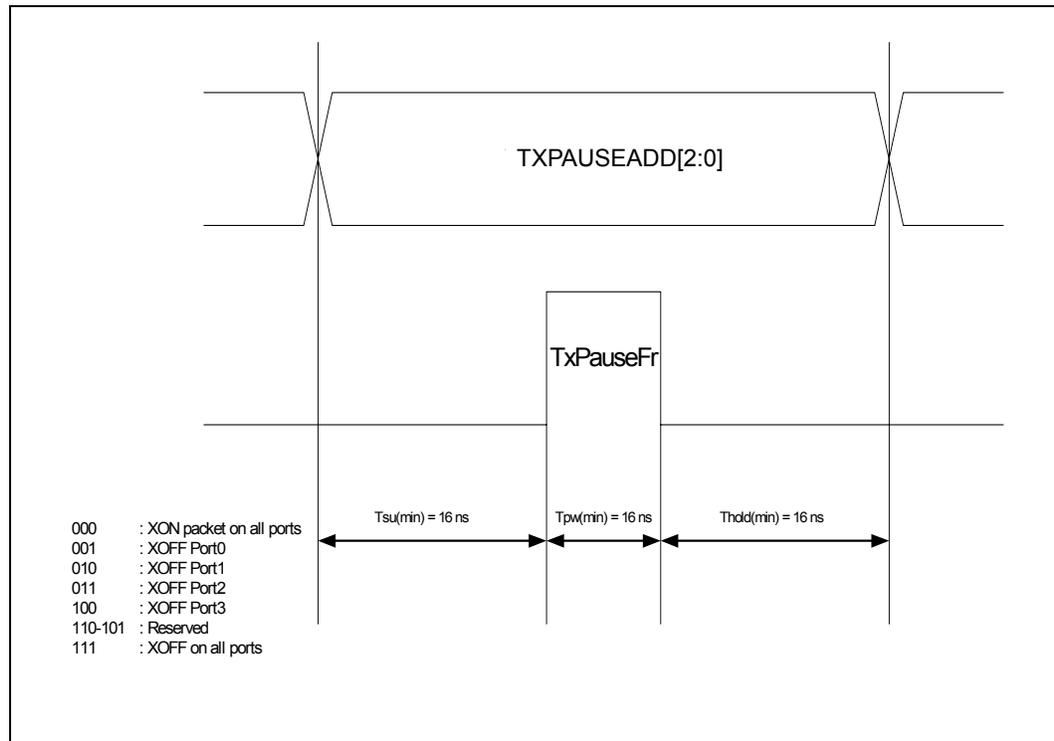


**Table 55      Transmit Pause Control Interface Timing Parameters**

| Symbol | Parameter | Min | Max | Units |
|--------|-----------|-----|-----|-------|
| Tsu | TXPAUSEADD stable prior to TXPAUSEFR High | 16 | – | ns |
| Tpw | TXPAUSEFR pulse width | 16 | – | ns |
| Thold | TXPAUSEADD stable after TXPAUSEFR High | 16 | – | ns |

## 7.10 JTAG AC Timing Specification

Figure 49 and Table 56 provide the JTAG AC timing specifications.

**Figure 49    JTAG AC Timing**
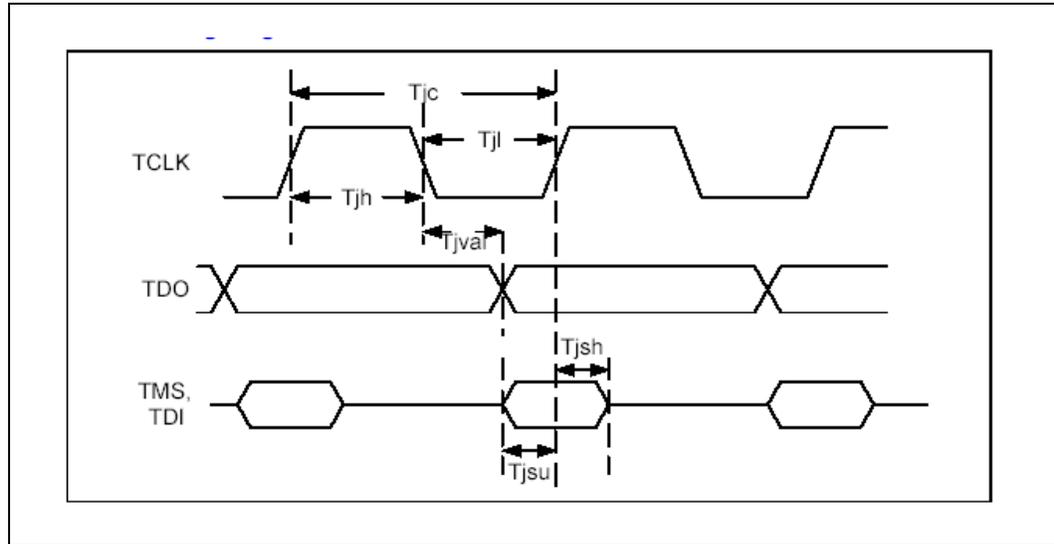


**Table 56    JTAG AC Timing Parameters**

| Symbol | Parameter | Min | Max | Units |
|--------|-----------|-----|-----|-------|
| Tjc | TCLK cycle time | 90 | - | ns |
| Tjh | TCLK High time | 0.4 x Tjc | 0.6 x Tjc | ns |
| Tjl | TCLK low time | 0.4 x Tjc | 0.6 x Tjc | ns |
| Tjval | TCLK falling edge to TDO valid | - | 25 | ns |
| Tjsu | TMS/TDI setup to TCLK | 20 | - | ns |
| Tjsh | TMS/TDI hold from TCLK | 5 | - | ns |

## 7.11 System AC Timing Specification

Figure 50 and Table 57 illustrate the system reset AC timing specifications.

**Figure 50    System Reset AC Timing**
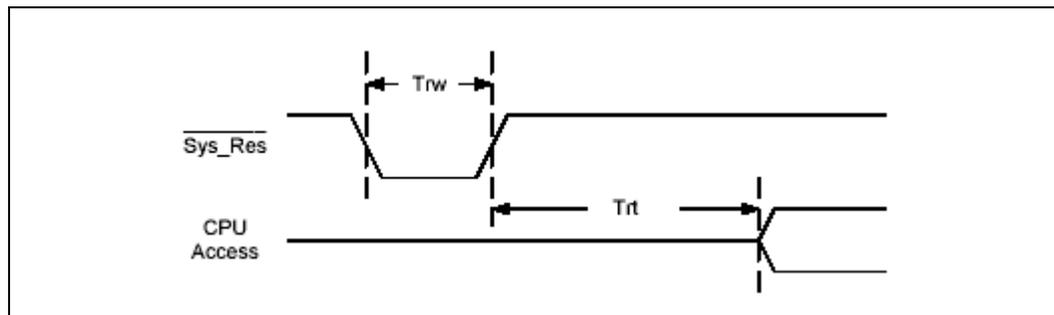
**Table 57     System Reset AC Timing Parameters**

| Symbol | Parameter | Min | Max | Units |
|--------|-----------|-----|-----|-------|
| Trw | Reset pulse width | 1.0 | - | µs |
| Trt | Reset recovery time | 200 | - | µs |

## 7.12     LED AC Timing Specification

Figure 51 and Table 58 provide the LED AC timing specifications.

**Figure 51     LED AC Interface Timing**
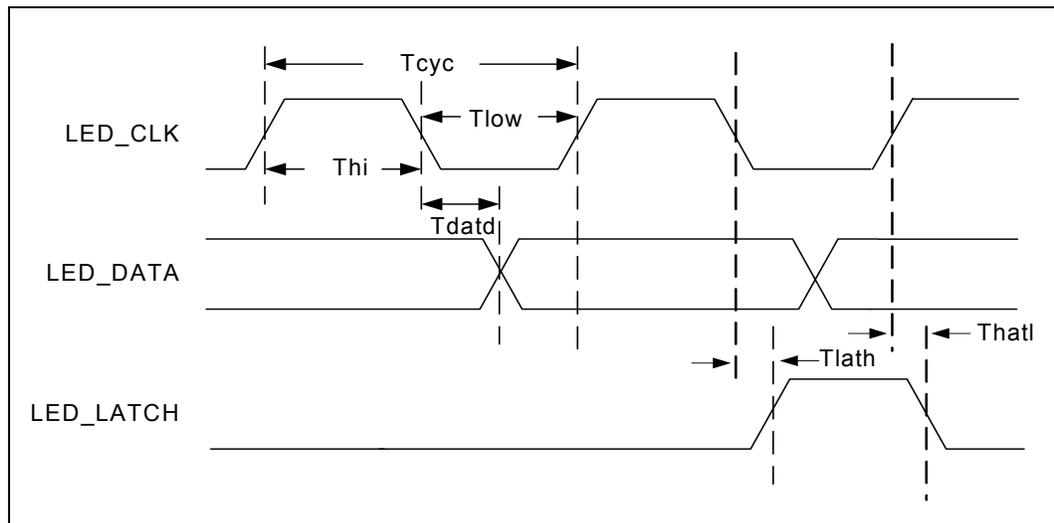


**Table 58     LED Interface AC Timing Parameters**

| Symbol | Parameter | Min | Max | Units |
|--------|-----------|-----|-----|-------|
| Tcyc | LED_CLK cycle time | 1.36 | 1.40 | ms |
| Thi | LED_CLK High time | 680 | 700 | µs |
| Tlow | LED_CLK low time | 680 | 700 | µs |
| Tdatd | LED_CLK falling edge to LED_DATA valid | 2 | 5 | ns |
| Tlath | LED_CLK rising edge to LED_LATCH rising edge | 690 | 700 | µs |
| Tlatl | LED_CLK falling edge to LED_LATCH falling edge | 690 | 700 | µs |

# 8.0    Register Set

The registers shown in this section provide access for configuration, alarm monitoring, and control of the chip. Table 59, *MAC Control Registers ($ Port Index + Offset)*, on page 149 through Table 69, *Optical Module Registers ($ 0x799 - 0x79F)*, on page 155 provide register map details. The registers are listed by ascending address in the table.

## 8.1    Document Structure

The following sections are structured to provide a general overview of the register map. Later sections provide detailed descriptions of each register segment or bit.

All registers are accessed and addressed as 32-bit doublewords. When accessed using 8- or 16-bit accesses, the CPU interface packs or unpacks the partial accesses into a 32-bit register value.

## 8.2    Graphical Representation

Figure 52 represents an overview of the IXF1104 MAC global control status registers that are used to configure or report on all ports. All register locations shown in Figure 52 represent a 32-bit double word.

**Figure 52    Memory Overview Diagram**
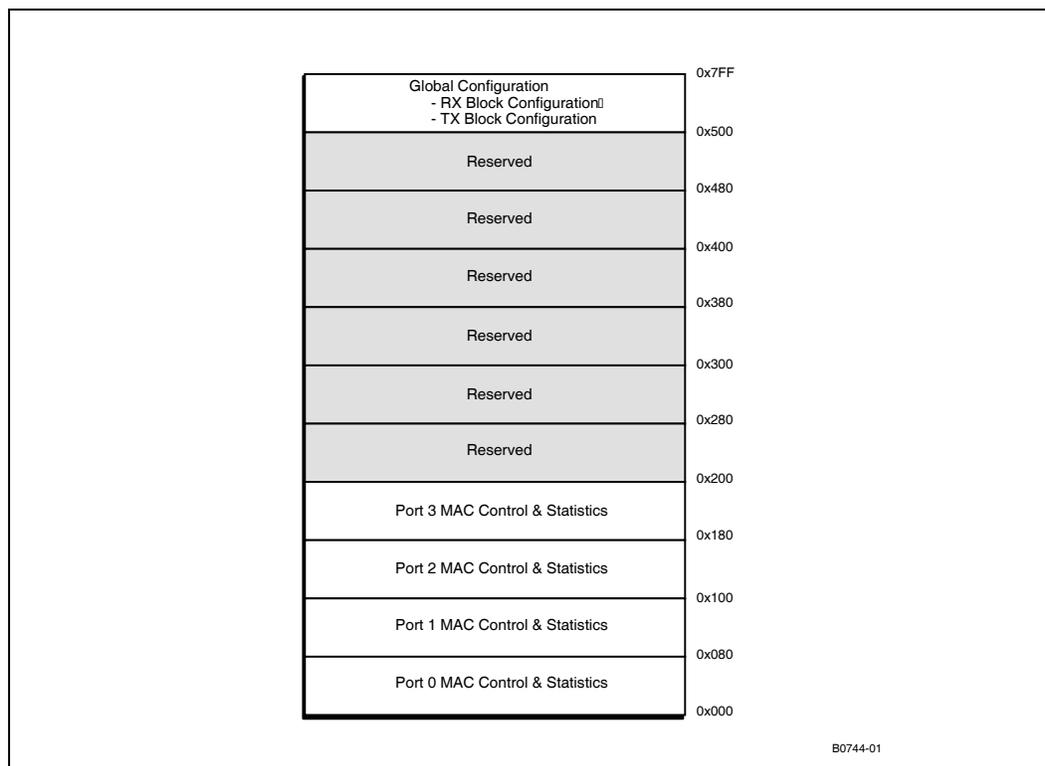
## 8.3 Per Port Registers

Section 8.4 covers all of the registers that are replicated in each port of the IXF1104 MAC. These registers perform an identical function in each port.

The address vector for the IXF1104 MAC is 11 bits wide. This allows for 7 bits of port-specific access and a 4-bit vector to address each port and all global registers. The address format is shown in Figure 53.

**Figure 53    Register Overview Diagram**

| 10 | 6 | 0 |
|----|---|---|
| Port Select & Global Registers | Per-Port Registers | |
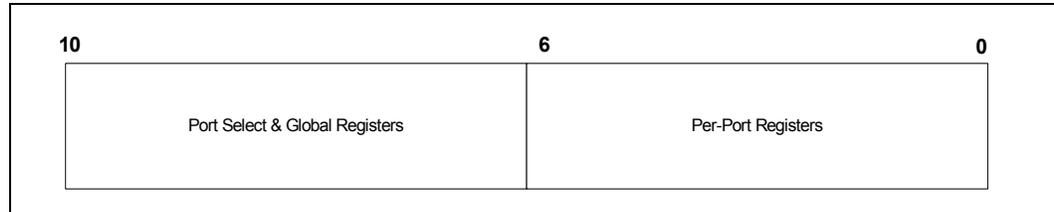
## 8.4 Register Map

Table 59 through Table 69, *Optical Module Registers ($ 0x799 - 0x79F)*, on page 155 present the IXF1104 MAC memory map details. Global control and status registers are used to configure or report on all ports, and some registers are replicated on a per-port basis.

**Note:**    All IXF1104 MAC registers are 32 bits.

**Table 59    MAC Control Registers ($ Port Index + Offset) (Sheet 1 of 2)**

| Register | Bit Size | Mode[1] | Ref Page | Offset |
|----------|----------|---------|----------|--------|
| Section 70, *Station Address ($ Port_Index +0x00 – +0x01)* Low | 32 | R/W | page 156 | 0x00 |
| Section 70, *Station Address ($ Port_Index +0x00 – +0x01)* High | 32 | R/W | page 156 | 0x01 |
| Section 71, *Desired Duplex ($ Port_Index + 0x02)* | 32 | R/W | page 156 | 0x02 |
| Section 72, *FD FC Type ($ Port_Index + 0x03)* | 32 | R/W | page 156 | 0x03 |
| Reserved | 32 | R | – | 0x04 |
| Section 73, *Collision Distance ($ Port_Index + 0x05)* | 32 | R/W | page 157 | 0x05 |
| Section 74, *Collision Threshold ($ Port_Index + 0x06)* | 32 | R/W | page 157 | 0x06 |
| Section 75, *FC TX Timer Value ($ Port_Index + 0x07)* | 32 | R/W | page 157 | 0x07 |
| Section 76, *FD FC Address ($ Port_Index + 0x08 – + 0x09)* FDFCAddressLow | 32 | R/W | page 157 | 0x08 |
| Section 76, *FD FC Address ($ Port_Index + 0x08 – + 0x09)* FDFCAddressHigh | 32 | R/W | page 157 | 0x09 |
| Section 77, *IPG Receive Time 1 ($ Port_Index + 0x0A)* | 32 | R/W | page 158 | 0x0A |
| Section 78, *IPG Receive Time 2 ($ Port_Index + 0x0B)* | 32 | R/W | page 158 | 0x0B |
| Section 79, *IPG Transmit Time ($ Port_Index + 0x0C)* | 32 | R/W | page 158 | 0x0C |
| Reserved | – | RO | – | 0x0D |
| Section 80, *Pause Threshold ($ Port_Index + 0x0E)* | 32 | R/W | page 159 | 0x0E |

**Table 59      MAC Control Registers ($ Port Index + Offset) (Sheet 2 of 2)**

| Register | Bit Size | Mode[1] | Ref Page | Offset |
|---|---|---|---|---|
| Section 81, *Max Frame Size (Addr: Port_Index + 0x0F)* | 32 | R/W | page 159 | 0x0F |
| Section 82, *MAC IF Mode and RGMII Speed ($ Port_Index + 0x10)* | 32 | R/W | page 160 | 0x10 |
| Section 83, *Flush TX ($ Port_Index + 0x11)* | 32 | R/W | page 160 | 0x11 |
| Section 84, *FC Enable ($ Port_Index + 0x12)* | 32 | R/W | page 160 | 0x12 |
| Section 85, *FC Back Pressure Length ($ Port_Index + 0x13)* | 32 | R/W | page 161 | 0x13 |
| Section 86, *Short Runts Threshold ($ Port_Index + 0x14)* | 32 | R/W | page 162 | 0x14 |
| Section 87, *Discard Unknown Control Frame ($ Port_Index + 0x15)* | 32 | R/W | page 162 | 0x15 |
| Section 88, *RX Config Word ($ Port_Index + 0x16)* | 32 | RO | page 162 | 0x16 |
| Section 89, *TX Config Word ($ Port_Index + 0x17)* | 32 | R/W | page 163 | 0x17 |
| Section 90, *Diverse Config Write ($ Port_Index + 0x18)* | 32 | R/W | page 164 | 0x18 |
| Section 91, *RX Packet Filter Control ($ Port_Index + 0x19)* | 32 | R/W | page 165 | 0x19 |
| Section 92, *Port Multicast Address ($ Port_Index +0x1A – +0x1B)* PortMulticastAddressLow | 32 | R/W | page 166 | 0x1A |
| Section 92, *Port Multicast Address ($ Port_Index +0x1A – +0x1B)* PortMulticastAddressHigh | 32 | R/W | page 166 | 0x1B |

**Table 60      MAC RX Statistics Registers ($ Port Index + Offset) (Sheet 1 of 2)**

| Register | Bit Size | Mode[1] | Ref Page | Offset |
|---|---|---|---|---|
| RxOctetsTotalOK | 32 | R | page 167 | 0x20 |
| RxOctetsBAD | 32 | R | page 167 | 0x21 |
| RxUCPckts | 32 | R | page 167 | 0x22 |
| RxMCPkts | 32 | R | page 167 | 0x23 |
| RxBCPkts | 32 | R | page 167 | 0x24 |
| RxPkts64Octets | 32 | R | page 167 | 0x25 |
| RxPkts65to127Octets | 32 | R | page 167 | 0x26 |
| RxPkts128to255Octets | 32 | R | page 167 | 0x27 |
| RxPkts256to511Octets | 32 | R | page 167 | 0x28 |
| RxPkts512to1023Octets | 32 | R | page 167 | 0x29 |
| RxPkts1024to1518Octets | 32 | R | page 167 | 0x2A |
| RxPkts1519toMaxOctets | 32 | R | page 167 | 0x2B |
| RxFCSErrors | 32 | R | page 167 | 0x2C |
| RxTagged | 32 | R | page 167 | 0x2D |
| RxDataError | 32 | R | page 167 | 0x2E |
| RxAlign Errors | 32 | R | page 167 | 0x2F |
| RxLongErrors | 32 | R | page 167 | 0x30 |

**Table 60    MAC RX Statistics Registers ($ Port Index + Offset) (Sheet 2 of 2)**

| Register | Bit Size | Mode[1] | Ref Page | Offset |
|---|---|---|---|---|
| RxJabberErrors | 32 | R | page 167 | 0x31 |
| PauseMacControlReceivedCounter | 32 | R | page 167 | 0x32 |
| RxUnknownMacControlFrameCounter | 32 | R | page 167 | 0x33 |
| RxVeryLongErrors | 32 | R | page 167 | 0x34 |
| RxRuntErrors | 32 | R | page 167 | 0x35 |
| RxShortErrors | 32 | R | page 167 | 0x36 |
| RxCarrierExtendError | 32 | R | page 167 | 0x37 |
| RxSequenceErrors | 32 | R | page 167 | 0x38 |
| RxSymbolErrors | 32 | R | page 167 | 0x39 |

**Table 61    MAC TX Statistics Registers ($ Port Index + Offset)**

| Register | Bit Size | Mode[1] | Ref Page | Offset |
|---|---|---|---|---|
| OctetsTransmittedOK | 32 | R | page 171 | 0x40 |
| OctetsTransmittedBad | 32 | R | page 171 | 0x41 |
| TxUCPkts | 32 | R | page 171 | 0x42 |
| TxMCPkts | 32 | R | page 171 | 0x43 |
| TxBCPkts | 32 | R | page 171 | 0x44 |
| TxPkts64Octets | 32 | R | page 171 | 0x45 |
| TxPkts65to127Octets | 32 | R | page 171 | 0x46 |
| TxPkts128to255Octets | 32 | R | page 171 | 0x47 |
| TxPkts256to511Octets | 32 | R | page 171 | 0x48 |
| TxPkts512to1023Octets | 32 | R | page 171 | 0x49 |
| TxPkts1024to1518Octets | 32 | R | page 171 | 0x4A |
| TxPkts1519toMaxOctets | 32 | R | page 171 | 0x4B |
| TxDeferred | 32 | R | page 171 | 0x4C |
| TxTotalCollisions | 32 | R | page 171 | 0x4D |
| TxSingleCollisions | 32 | R | page 171 | 0x4E |
| TxMultipleCollisions | 32 | R | page 171 | 0x4F |
| TxLateCollisions | 32 | R | page 171 | 0x50 |
| TxExcessiveCollisionErrors | 32 | R | page 171 | 0x51 |
| TxExcessiveDeferralErrors | 32 | R | page 171 | 0x52 |
| TxExcessiveLengthDrop | 32 | R | page 171 | 0x53 |
| TxUnderrun | 32 | R | page 171 | 0x54 |
| TxTagged | 32 | R | page 171 | 0x55 |
| TxCRCError | 32 | R | page 171 | 0x56 |
| TxPauseFrames | 32 | R | page 171 | 0x57 |
| TxFlowControlCollisionsSend | 32 | R | page 171 | 0x58 |

**Table 62** **PHY Autoscan Registers ($ Port Index + Offset)**

| Register | Bit Size | Mode[1] | Ref Page | Offset |
|---|---|---|---|---|
| Section 95, *PHY Control ($ Port Index + 0x60)* | 32 | RO | page 174 | 0x60 |
| Section 96, *PHY Status ($ Port Index + 0x61)* | 32 | RO | page 175 | 0x61 |
| Section 97, *PHY Identification 1 ($ Port Index + 0x62)* | 32 | RO | page 176 | 0x62 |
| Section 98, *PHY Identification 2 ($ Port Index + 0x63)* | 32 | RO | page 177 | 0x63 |
| Section 99, *Auto-Negotiation Advertisement ($ Port Index + 0x64)* | 32 | RO | page 177 | 0x64 |
| Section 100, *Auto-Negotiation Link Partner Base Page Ability ($ Port Index + 0x65)* | 32 | RO | page 178 | 0x65 |
| Section 101, *Auto-Negotiation Expansion ($ Port Index + 0x66)* | 32 | RO | page 179 | 0x66 |
| Section 102, *Auto-Negotiation Next Page Transmit ($ Port Index + 0x67)* | 32 | RO | page 180 | 0x67 |
| Reserved | 32 | RO | – | 0x68 - 0x6F |

**Table 63** **Global Status and Configuration Registers ($ 0x500 - 0X50C)**

| Register | Bit Size | Mode[1] | Ref Page | Address |
|---|---|---|---|---|
| Section 103, *Port Enable ($0x500)* | 32 | R/W | page 181 | 0x500 |
| Section 104, *Interface Mode ($0x501)* | 32 | R/W | page 181 | 0x501 |
| Section 105, *Link LED Enable ($0x502)* | 32 | R/W | page 182 | 0x502 |
| Reserved | 32 | RO | – | 0x503 - 0x504 |
| Section 106, *MAC Soft Reset ($0x505)* | 32 | R/W | page 182 | 0x505 |
| Section 107, *MDIO Soft Reset ($0x506)* | 32 | R/W | page 183 | 0x506 |
| Reserved | 32 | RO | – | 0x507 |
| Section 108, *CPU Interface ($0x508)* | 32 | R/W | page 183 | 0x508 |
| Section 109, *LED Control ($0x509)* | 32 | R/W | page 183 | 0x509 |
| Section 110, *LED Flash Rate ($0x50A)* | 32 | R/W | page 184 | 0x50A |
| Section 111, *LED Fault Disable ($0x50B)* | 32 | R/W | page 184 | 0x50B |
| Section 112, *JTAG ID ($0x50C)* | 32 | R | page 185 | 0x50C |

**Table 64** **RX FIFO Registers ($ 0x580 - 0x5BF) (Sheet 1 of 2)**

| Register | Bit Size | Mode[1] | Ref Page | Address |
|---|---|---|---|---|
| Section 113, *RX FIFO High Watermark Port 0 ($0x580)* | 32 | R/W | page 185 | 0x580 |
| Section 114, *RX FIFO High Watermark Port 1 ($0x581)* | 32 | R/W | page 186 | 0x581 |
| Section 115, *RX FIFO High Watermark Port 2 ($0x582)* | 32 | R/W | page 186 | 0x582 |
| Section 116, *RX FIFO High Watermark Port 3 ($0x583)* | 32 | R/W | page 186 | 0x583 |
| Reserved | 32 | RO | – | 0x584 - 0x589 |
| Section 117, *RX FIFO Low Watermark Port 0 ($0x58A)* | 32 | R/W | page 187 | 0x58A |

**Table 64     RX FIFO Registers ($ 0x580 - 0x5BF) (Sheet 2 of 2)**

| Register | Bit Size | Mode[1] | Ref Page | Address |
|---|---|---|---|---|
| Section 118, *RX FIFO Low Watermark Port 1 ($0x58B)* | 32 | R/W | page 187 | 0x58B |
| Section 119, *RX FIFO Low Watermark Port 2 ($0x58C)* | 32 | R/W | page 187 | 0x58C |
| Section 120, *RX FIFO Low Watermark Port 3 ($0x58D)* | 32 | R/W | page 188 | 0x58D |
| Reserved | 32 | RO | – | 0x58E - 0x593 |
| RX FIFO Overflow Frame Drop Counter Port 0 | 32 | R | page 188 | 0x594 |
| RX FIFO Overflow Frame Drop Counter Port 1 | 32 | R | page 188 | 0x595 |
| RX FIFO Overflow Frame Drop Counter Port 2 | 32 | R | page 188 | 0x596 |
| RX FIFO Overflow Frame Drop Counter Port 3 | 32 | R | page 188 | 0x597 |
| Reserved | 32 | RO | – | 0x598 - 0x59D |
| Section 122, *RX FIFO Port Reset ($0x59E)* | 32 | R/W | page 188 | 0x59E |
| Section 123, *RX FIFO Errored Frame Drop Enable ($0x59F)* | 32 | R/W | page 189 | 0x59F |
| Section 124, *RX FIFO Overflow Event ($0x5A0)* | 32 | R | page 190 | 0x5A0 |
| Reserved | 32 | R | – | 0x5A1 - 0x5A5 |
| RX FIFO Errored Frame Drop Counter Port 0 | 32 | R | page 191 | 0x5A2 |
| RX FIFO Errored Frame Drop Counter Port 1 | 32 | R | page 191 | 0x5A3 |
| RX FIFO Errored Frame Drop Counter Port 2 | 32 | R | page 191 | 0x5A4 |
| RX FIFO Errored Frame Drop Counter Port 3 | 32 | R | page 191 | 0x5A5 |
| Reserved | 32 | RO | – | 0x5A6 - 0x5B1 |
| Section 126, *RX FIFO SPI3 Loopback Enable for Ports 0 - 3 ($0x5B2)* | 32 | R/W | page 192 | 0x5B2 |
| Section 127, *RX FIFO Padding and CRC Strip Enable ($0x5B3)* | 32 | R/W | page 192 | 0x5B3 |
| Reserved | 32 | R | – | 0x5B4 - 0x5B7 |
| Section 128, *RX FIFO Transfer Threshold Port 0 ($0x5B8)* | 32 | R/W | page 193 | 0x5B8 |
| Section 129, *RX FIFO Transfer Threshold Port 1 ($0x5B9)* | 32 | R/W | page 193 | 0x5B9 |
| Section 130, *RX FIFO Transfer Threshold Port 2 ($0x5BA)* | 32 | R/W | page 193 | 0x5BA |
| Section 131, *RX FIFO Transfer Threshold Port 3 ($0x5BB)* | 32 | R/W | page 194 | 0x5BB |
| Reserved | 32 | R | – | 0x5BC - 0x5BF |

**Table 65     TX FIFO Registers ($ 0x600 - 0x63E) (Sheet 1 of 2)**

| Register | Bit Size | Mode[1] | Ref Page | Address |
|---|---|---|---|---|
| TX FIFO High Watermark Port 0 | 32 | R/W | page 195 | 0x600 |
| TX FIFO High Watermark Port 1 | 32 | R/W | page 195 | 0x601 |
| TX FIFO High Watermark Port 2 | 32 | R/W | page 195 | 0x602 |
| TX FIFO High Watermark Port 3 | 32 | R/W | page 195 | 0x603 |

**Table 65      TX FIFO Registers ($ 0x600 - 0x63E) (Sheet 2 of 2)**

| Register | Bit Size | Mode[1] | Ref Page | Address |
|---|---|---|---|---|
| Reserved | 32 | RO | – | 0x604 - 0x609 |
| TX FIFO Low Watermark Port 0 | 32 | R/W | page 196 | 0x60A |
| TX FIFO Low Watermark Port 1 | 32 | R/W | page 196 | 0x60B |
| TX FIFO Low Watermark Port 2 | 32 | R/W | page 196 | 0x60C |
| TX FIFO Low Watermark Port 3 | 32 | R/W | page 196 | 0x60D |
| Reserved | 32 | RO | – | 0x60E - 0x613 |
| TX FIFO MAC Threshold Port 0 | 32 | R/W | page 197 | 0x614 |
| TX FIFO MAC Threshold Port 1 | 32 | R/W | page 197 | 0x615 |
| TX FIFO MAC Threshold Port 2 | 32 | R/W | page 197 | 0x616 |
| TX FIFO MAC Threshold Port 3 | 32 | R/W | page 197 | 0x617 |
| Reserved | – | RO | – | 0x618 - 0x61D |
| TX FIFO Overflow/Underflow Event/Out of Sequence | 32 | R | page 198 | 0x61E |
| Loop RX Data to TX FIFO | 32 | R/W | page 199 | 0x61F |
| TX FIFO Port Reset | 32 | R/W | page 199 | 0x620 |
| TX FIFO Overflow Frame Drop Counter Port 0 | 32 | R | page 200 | 0x621 |
| TX FIFO Overflow Frame Drop Counter Port 1 | 32 | R | page 200 | 0x622 |
| TX FIFO Overflow Frame Drop Counter Port 2 | 32 | R | page 200 | 0x623 |
| TX FIFO Overflow Frame Drop Counter Port 3 | 32 | R | page 200 | 0x624 |
| TX FIFO Errored Frame Drop Counter Port 0 | 32 | R | page 201 | 0x625 |
| TX FIFO Errored Frame Drop Counter Port 1 | 32 | R | page 201 | 0x626 |
| TX FIFO Errored Frame Drop Counter Port 2 | 32 | R | page 201 | 0x627 |
| TX FIFO Errored Frame Drop Counter Port 3 | 32 | R | page 201 | 0x628 |
| Reserved | 32 | R | – | 0x629 - 0x62C |
| TX FIFO Occupancy Counter for Port 0 | 32 | R | page 202 | 0x62D |
| TX FIFO Occupancy Counter for Port 1 | 32 | R | page 202 | 0x62E |
| TX FIFO Occupancy Counter for Port 2 | 32 | R | page 202 | 0x62F |
| TX FIFO Occupancy Counter for Port 3 | 32 | R | page 202 | 0x630 |
| Reserved | 32 | R | – | 0x631 - 0x63E |

**Table 66      MDIO Registers ($ 0x680 - 0x683)**

| Register | Bit Size | Mode[1] | Ref Page | Address |
|---|---|---|---|---|
| Section 142, *MDIO Single Command ($0x680)* | 32 | R/W | page 202 | 0x680 |
| Section 143, *MDIO Single Read and Write Data ($0x681)* | 32 | R/W | page 203 | 0x681 |
| Section 144, *Autoscan PHY Address Enable ($0x682)* | 32 | R/W | page 203 | 0x682 |
| Section 145, *MDIO Control ($0x683)* | 32 | R/W | page 204 | 0x683 |

**Table 67**     **SPI3 Registers ($ 0x700 - 0x716)**

| Register | Bit Size | Mode[1] | Ref Page | Address |
|---|---|---|---|---|
| Section 146, *SPI3 Transmit and Global Configuration ($0x700)* | 32 | R/W | page 204 | 0x700 |
| Section 147, *SPI3 Receive Configuration ($0x701)* | 32 | R/W | page 207 | 0x701 |
| Reserved | 32 | R | – | 0x702 - 0x709 |
| Section 148, *Address Parity Error Packet Drop Counter ($0x70A)* | 32 | R | page 210 | 0x70A |
| Reserved | 32 | R | – | 0x70B - 0x716 |

**Table 68**     **SerDes Registers ($ 0x780 - 0x798)**

| Register | Bit Size | Mode[1] | Ref Page | Address |
|---|---|---|---|---|
| Reserved | 32 | RO | – | 0x780 - 0x783 |
| Section 149, *TX Driver Power Level Ports 0 - 3 ($0x784)* | 32 | R/W | page 210 | 0x784 |
| Reserved | 32 | RO | – | 0x785 - 0x786 |
| *TX and RX Power-Down ($0x787)* | 32 | R/W | page 210 | 0x787 |
| Reserved | 32 | RO | – | 0x788 - 0x792 |
| Section 151, *RX Signal Detect Level Ports 0 - 3 ($0x793)* | 32 | R/W | page 211 | 0x793 |
| Section 152, *Clock and Interface Mode Change Enable Ports 0 - 3 ($0x794)* | 32 | R/W | page 211 | 0x794 |
| Reserved | 32 | RO | – | 0x795 - 0x798 |

**Table 69**     **Optical Module Registers ($ 0x799 - 0x79F)**

| Register | Bit Size | Mode[1] | Ref Page | Address |
|---|---|---|---|---|
| Section 153, *Optical Module Status Ports 0-3 ($0x799)* | 32 | R | page 212 | 0x799 |
| Section 154, *Optical Module Control Ports 0 - 3 ($0x79A)* | 32 | R/W | page 213 | 0x79A |
| Section 155, *$I^2C$ Control Ports 0 - 3 ($0x79B)* | 32 | R/W | page 214 | 0x79B |
| Reserved | 32 | RO | – | 0x79C - 0x79E |
| Section 156, *$I^2C$ Data Ports 0 - 3 ($0x79F)* | 32 | R/W | page 214 | 0x79F |

## 8.4.1     MAC Control Registers

Table 70 through Table 92, *Port Multicast Address ($ Port_Index +0x1A – +0x1B)*, on page 166 provide details on the control and status registers associated with each MAC port. The register address is 'Port_index + 0x**', where the port index is set at any value from 0x0 through 0x5. All registers are 32-bit. The unused bits of the registers are read-only and are set permanently to zero.

**Table 70      Station Address ($ Port_Index +0x00 − +0x01)**

| Name | Description | Address | Type[1] | Default |
|---|---|---|---|---|
| Station Address Low | **Source MAC address bit 31-0.** This address is inserted in the source address field when transmitting pause frames, and is also used to compare against unicast pause frames at the receiving side. | Port_Index + 0x00 | R/W | 0x0000000 |
| Station Address High | **Source MAC address bit 47-32.** This address is inserted in the source address field when transmitting pause frames, and is also used to compare against unicast pause frames at the receiving side. Bits 15:0 of this register are assigned to bits 47:32 of the station address. | Port_Index + 0x01 | R/W | 0x00000000 |
| 1.   RO = Read Only, No clear on Read; R = Read, Clear on Read; W = Write only; R/W = Read/Write, No clear; R/W/C = Read/Write, Clear on Write | | | | |

**Table 71      Desired Duplex ($ Port_Index + 0x02)**

| Bit | Name | Description | Type[1] | Default |
|---|---|---|---|---|
| **Register Description:** Chooses between half-duplex and full-duplex operation in RGMII 100 Mbps or 10 Mbps mode only. This register must be set to the default value of 1 and must not be changed when operating in RGMII 1000 Mbps, GMII, or fiber mode. | | | | 0x00000001 |
| 31:1 | Reserved | Reserved | R | 0x00000000 |
| 0 | Duplex Select | 0 = Half-duplex 1 = Full-duplex **NOTE:** Half-duplex operation applies only to 10/100 Mbps speed on copper media in RGMII mode only. Gigabit speed on either media requires full-duplex. | R/W | 1 |
| 1.   RO = Read Only, No clear on Read; R = Read, Clear on Read; W = Write only; R/W = Read/Write, No clear; R/W/C = Read/Write, Clear on Write | | | | |

**Table 72      FD FC Type ($ Port_Index + 0x03)**

| Name | Description | Address | Type[1] | Default |
|---|---|---|---|---|
| FD FC Type | This value fills the Type field of the Transmitted Pause frames. Only bits 15:0 of this register are used. | Port_Index + 0x03 | R/W | 0x00008808 |
| 1.   RO = Read Only, No clear on Read; R = Read, Clear on Read; W = Write only; R/W = Read/Write, No clear; R/W/C = Read/Write, Clear on Write | | | | |

**Table 73      Collision Distance ($ Port_Index + 0x05)**

| Name | Description | Address | Type[1] | Default |
|------|-------------|---------|---------|---------|
| Collision Distance | This is a 10-bit value that sets the limit for late collision. Collisions happening at byte times beyond the configured value are considered to be late collisions. (Only valid in half-duplex). | Port_Index + 0x05 | R/W | 0x00000043 |
| 1. RO = Read Only, No clear on Read; R = Read, Clear on Read; W = Write only; R/W = Read/Write, No clear; R/W/C = Read/Write, Clear on Write ||||||

**Table 74      Collision Threshold ($ Port_Index + 0x06)**

| Name | Description | Address | Type[1] | Default |
|------|-------------|---------|---------|---------|
| Collision Threshold | This is a 4-bit value that sets the limit for excessive collisions. When the number of transmission attempts performed for a packet exceeds this value, it is considered to be an excessive collision and the frame is dropped. (Only valid in half-duplex). | Port_Index + 0x06 | R/W | 0x0000000F |
| 1. RO = Read Only, No clear on Read; R = Read, Clear on Read; W = Write only; R/W = Read/Write, No clear; R/W/C = Read/Write, Clear on Write ||||||

**Table 75      FC TX Timer Value ($ Port_Index + 0x07)**

| Name | Description | Address | Type[1] | Default |
|------|-------------|---------|---------|---------|
| FC TX Timer Value | The 16-bit pause length inserted in the flow control pause frame sent to the receiving station. The value is in 512-bit times. | Port_Index + 0x07 | R/W | 0x0000005E |
| 1. RO = Read Only, No clear on Read; R = Read, Clear on Read; W = Write only; R/W = Read/Write, No clear; R/W/C = Read/Write, Clear on Write ||||||

**Table 76      FD FC Address ($ Port_Index + 0x08 − + 0x09)**

| Name | Description | Address | Type[1] | Default |
|------|-------------|---------|---------|---------|
| FD FC Address Low | The lowest 32 bits of the 48-bit globally assigned multicast pause frame destination address. | Port_Index + 0x08 | R/W | 0xC2000001 |
| FD FC Address High | The highest 16 bits (47:32) of the globally assigned multicast pause frame destination address. The higher 16-bit address is derived from bits 15:0 of this register. | Port_Index + 0x09 | R/W | 0x00000180 |
| 1. RO = Read Only, No clear on Read; R = Read, Clear on Read; W = Write only; R/W = Read/Write, No clear; R/W/C = Read/Write, Clear on Write ||||||

**Table 77** **IPG Receive Time 1 ($ Port_Index + 0x0A)**

| Name | Description | Address | Type[1] | Default |
|------|-------------|---------|---------|---------|
| IPG Receive Time 1 | This timer is used during half-duplex operation when there is a packet waiting for transmission from the MAC. This timer starts after CRS is de-asserted. If CRS is asserted during this time, no transmission is initiated and the counter restarts once CRS is de-asserted again.<br><br>The value specified in this register is calculated as follows: (register_value * 8) = RXIPG1 in terms of bit times. Therefore, a default value of 8 gives the following: (8 * 8 = 64 bit times for the default). | Port_Index + 0x0A | R/W | 0x00000008 |
| 1. RO = Read Only, No clear on Read; R = Read, Clear on Read; W = Write only; R/W = Read/Write, No clear; R/W/C = Read/Write, Clear on Write | | | | |

**Table 78** **IPG Receive Time 2 ($ Port_Index + 0x0B)**

| Name | Description | Address | Type[1] | Default |
|------|-------------|---------|---------|---------|
| IPG Receive Time 2 | This is only used in half-duplex operation. It starts counting at the same time as RXIPG1. Once RXIPG1 expires, a frame is transmitted when RXIPG2 expires regardless of the CRS value. If CRS is asserted before RXIPG1 expires, no transmission occurs and both RXIPG1 an RXIPG2 are reset once CRS is de-asserted again.<br><br>The value specified in this register is calculated as follows: (register_value +5) * 8 = RXIPG2 in terms of bit times. Therefore, a default of 7 gives the following:<br>(7+5) * 8 = 96 bit times for default. | Port_Index + 0x0B | R/W | 0x00000007 |
| 1. RO = Read Only, No clear on Read; R = Read, Clear on Read; W = Write only; R/W = Read/Write, No clear; R/W/C = Read/Write, Clear on Write | | | | |

**Table 79** **IPG Transmit Time ($ Port_Index + 0x0C)**

| Name | Description | Address | Type[1] | Default |
|------|-------------|---------|---------|---------|
| IPG Transmit Time | This is a 10-bit value configuring IPG time for back-to-back transmissions.<br><br>The value specified in this register is calculated as follows: (register_value +4) * 8 = TXIPG in terms of bit times. Therefore, a default value of 8 gives the following:<br>(8+4) * 8 = 96 bit times for the default. | Port_Index + 0x0C | R/W | 0x00000008 |
| 1. RO = Read Only, No clear on Read; R = Read, Clear on Read; W = Write only; R/W = Read/Write, No clear; R/W/C = Read/Write, Clear on Write | | | | |

**Table 80        Pause Threshold ($ Port_Index + 0x0E)**

| Name | Description | Address | Type[1] | Default |
|------|-------------|---------|---------|---------|
| Pause Threshold | When a pause frame has been sent, an internal timer checks when the next pause frame must be scheduled for transmission to keep the link partner in pause mode (this is required only if the flow control has to be extended for one more session). The pause threshold value is a 16-bit value that sets the time in terms of 512-bit quantum after the previous pause frame when the next pause frame has to be sent. This ensures that the link partner is kept in pause mode continuously. | Port_Index + 0x0E | R/W | 0x0000002F |
| 1. RO = Read Only, No clear on Read; R = Read, Clear on Read; W = Write only; R/W = Read/Write, No clear; R/W/C = Read/Write, Clear on Write | | | | |

**Table 81        Max Frame Size (Addr: Port_Index + 0x0F)**

| Name | Description | Address | Type[1] | Default |
|------|-------------|---------|---------|---------|
| Max Frame Size | This is a 14-bit value configuring the maximum frame size the MAC can receive or transmit without activating any error counters, and without truncation. This value is excluding the 4-byte CRC in the transmit direction when CRC append is enabled in the MAC. Hence, this value has to be set four bytes less when CRC append is enabled in the MAC. The maximum frame size is internally adjusted by +4 if the frame is VLAN tagged. | Port_Index + 0x0F | R/W | 0x000005EE |
| 1. RO = Read Only, No clear on Read; R = Read, Clear on Read; W = Write only; R/W = Read/Write, No clear; R/W/C = Read/Write, Clear on Write | | | | |

**Table 82        MAC IF Mode and RGMII Speed ($ Port_Index + 0x10)**

| Bit | Name | Description | Type[1] | Default |
|---|---|---|---|---|
| | | | | |
| **Register Description – MAC IF Mode:** Determines the MAC operation frequency and mode per port.<br><br>Changes to the data setting of this register must be made in conjunction with the *Clock and Interface Mode Change Enable Ports 0 - 3 ($0x794)* to ensure a safe transition to a new operational mode. Changes to this register must follow a proper sequence. Refer to Section 6.1, *Change Port Mode Initialization Sequence*, on page 127 for the proper sequence for changing the port mode and speed. | | | | 0x00000003 |
| 31:3 | Reserved | Reserved | R | 0x00000000 |
| 2:0 | Port Mode | These bits are used to define the clock mode and the RGMII/GMII mode of operation.<br>000 = Reserved<br>001 = Reserved<br>010 = GMII 1000 Mbps operation<br>011 = Reserved<br>100 = RGMII 10 Mbps operation<br>101 = RGMII 100 Mbps operation<br>11x = RGMII 1000 Mbps operation | R/W | 011 |
| 1.  RO = Read Only, No clear on Read; R = Read, Clear on Read; W = Write only; R/W = Read/Write, No clear; R/W/C = Read/Write, Clear on Write | | | | |

**Table 83        Flush TX ($ Port_Index + 0x11)**

| Bit | Name | Description | Type[1] | Default |
|---|---|---|---|---|
| **Register Description:** Used to flush all TX data. It is used if all traffic sent to a port should be stopped. | | | | 0x00000000 |
| 31:1 | Reserved | Reserved | R | 0x00000000 |
| 0 | Flush TX | This bit flushes all TX data and is used if all the traffic sent to a port should be stopped. | R/W | 0 |
| 1.  RO = Read Only, No clear on Read; R = Read, Clear on Read; W = Write only; R/W = Read/Write, No clear; R/W/C = Read/Write, Clear on Write | | | | |

**Table 84        FC Enable ($ Port_Index + 0x12) (Sheet 1 of 2)**

| Bit | Name | Description | Type[1] | Default |
|---|---|---|---|---|
| **Register Description:** Indicates which flow control mode is used for the RX and TX MAC. | | | | 0x00000007 |
| 31:3 | Reserved | Reserved | R | 0x00000000 |
| 1.  RO = Read Only, No clear on Read; R = Read, Clear on Read; W = Write only; R/W = Read/Write, No clear; R/W/C = Read/Write, Clear on Write | | | | |

**Table 84      FC Enable ($ Port_Index + 0x12) (Sheet 2 of 2)**

| Bit | Name | Description | Type[1] | Default |
|---|---|---|---|---|
| 2 | TX HDFC | When TX HDFC is enabled (half-duplex mode only), the MAC generates deliberate collisions on incoming packets when the RX FIFO occupancy crosses the High Watermark (flow control).<br>0 = Disable TX half-duplex flow control<br>1 = Enable TX half-duplex flow control | R/W | 1 |
| 1 | TX FDFC | 0 = Disable TX full-duplex flow control [the MAC will not generate internally any flow control frames based on the RX FIFO watermarks or the Transmit Pause Control interface<br>1 = Enable TX full-duplex flow control [enables the MAC to send flow control frames to the link partner based on the RX FIFO programmable watermarks or the Transmit Pause Control interface] | R/W | 1 |
| 0 | RX FDFC | 0 = Disable RX full-duplex flow control [the MAC will not respond to flow control frames sent to it by the link partner]<br>1 = Enable RX full-duplex flow control [MAC will respond to flow control frames sent by the link partner and will stop packet transmission for the time specified in the flow control frame] | R/W | 1 |
| 1. RO = Read Only, No clear on Read; R = Read, Clear on Read; W = Write only; R/W = Read/Write, No clear; R/W/C = Read/Write, Clear on Write |||||

**Table 85      FC Back Pressure Length ($ Port_Index + 0x13)**

| Name | Description | Address | Type[1] | Default |
|---|---|---|---|---|
| FC Back Pressure Length | This register sets number the byte cycles for which the collision has to be applied.<br>The 6-bit configuration holds the value in bytes, which applies to the minimum length/duration of back pressure in half-duplex mode. Flow control in the receive path is executed by deliberately colliding the incoming packets in half-duplex mode.<br>Register bits 5:0 are used alone. | Port Add + 0x13 | R/W | 0x0000000C |
| 1. RO = Read Only, No clear on Read; R = Read, Clear on Read; W = Write only; R/W = Read/Write, No clear; R/W/C = Read/Write, Clear on Write |||||

**Table 86      Short Runs Threshold ($ Port_Index + 0x14)**

| Name | Description | Address | Type[1] | Default |
|------|-------------|---------|---------|---------|
| Short Runs Threshold | The 5-bit configuration holds the value in bytes, which applies to the threshold in determining between runts and short. The bits 4:0 of this register are alone used.<br><br>A received packet is reported as a short packet when the length (excluding Preamble and SFD) is less than this value.<br><br>A received packet is reported as a runt packet when the length (excluding Preamble and SFD) is equal to or greater than this value and less than 64-bytes.<br><br>*Note:*   This register is only relevant when the IXF1104 MAC port is configured for copper operation (the line side interface is configured for either RGMII or GMII). | Port_Index + 0x14 | R/W | 0x00000008 |
| 1.   RO = Read Only, No clear on Read; R = Read, Clear on Read; W = Write only; R/W = Read/Write, No clear; R/W/C = Read/Write, Clear on Write | | | | |

**Table 87      Discard Unknown Control Frame ($ Port_Index + 0x15)**

| Bit | Name | Description | Type[1] | Default |
|-----|------|-------------|---------|---------|
| **Register Description:** Discards or forwards unknown control frames. Known control frames are pause frames. | | | | 0x00000000 |
| 31:1 | Reserved | Reserved | R | 0x00000000 |
| 0 | Discard Unknown Control Frame | 0 = Forward unknown control frames<br>1 = Discard unknown control frames | R/W | 0 |
| 1.   RO = Read Only, No clear on Read; R = Read, Clear on Read; W = Write only; R/W = Read/Write, No clear; R/W/C = Read/Write, Clear on Write | | | | |

**Table 88      RX Config Word ($ Port_Index + 0x16) (Sheet 1 of 2)**

| Bit | Name | Description | Type[1] | Default |
|-----|------|-------------|---------|---------|
| **Register Description:** This register is used in fiber MAC only for auto-negotiation and to report the receive status. The lower 16 bits of this register are the "config_reg" received from the link partner, as described in IEEE 802.3 2000 Edition, Section 37.2.1. | | | | 0x00000000 |
| 31:22 | Reserved | Reserved | RO | 0x000 |
| 21 | An_complete | Auto-negotiation complete. This bit remains cleared from the time auto-negotiation is reset until auto-negotiation reaches the "LINK_OK" state. It remains set until auto-negotiation is disabled or restarted.<br><br>This bit is only valid if auto-negotiation is enabled. | RO | 0 |
| 20 | Rx Sync | 0 = Loss of synchronization<br>1 = Bit synchronization. The bit remains Low until the register is read. | RO | 0 |
| 1.   RO = Read Only, No clear on Read; R = Read, Clear on Read; W = Write only; R/W = Read/Write, No clear; R/W/C = Read/Write, Clear on Write | | | | |

**Table 88      RX Config Word ($ Port_Index + 0x16) (Sheet 2 of 2)**

| Bit | Name | Description | Type[1] | Default |
|---|---|---|---|---|
| 19 | RX Config | 0 = Receiving idle/data stream<br>1 = Receiving /C/ ordered sets | RO | 0 |
| 18 | Config Changed | 0 = RxConfigWord has changed since last read<br>1 = RxConfigWord has not changed since last read.<br>This bit remains High until the register is read. | R | 0 |
| 17 | Invalid Word | 0 = Have not received an invalid symbol<br>1 = Have received an invalid symbol<br>This bit remains High until the register is read. | RO | 0 |
| 16 | Carrier Sense | 0 = Device is not receiving idle characters; carrier sense is true.<br>1 = Device is receiving idle characters; carrier sense is false. | RO | 0 |
| 15 | Next Page | Next Page request | RO | 0 |
| 14 | Reserved | Reserved | RO | 0 |
| 13:12[2] | Remote Fault [1:0] | Remote fault definitions:<br>00 = No error, link okay<br>01 = Offline<br>10 = Link failure<br>11 = Auto-negotiation_Error | R/W | 00 |
| 11:9 | Reserved | Reserved | RO | 000 |
| 8 | Asym Pause | Asym Pause. The ability to send pause frames. | RO | 0 |
| 7 | Sym Pause | Sym Pause. The ability to send and receive pause frames. | RO | 0 |
| 6 | Half Duplex | Half-duplex | RO | 0 |
| 5 | Full Duplex | Full-duplex | RO | 0 |
| 4:0 | Reserved | Reserved | RO | 0x0 |

1.  RO = Read Only, No clear on Read; R = Read, Clear on Read; W = Write only; R/W = Read/Write, No clear; R/W/C = Read/Write, Clear on Write

**Table 89      TX Config Word ($ Port_Index + 0x17) (Sheet 1 of 2)**

| Bit | Name | Description | Type[1] | Default |
|---|---|---|---|---|
| **Register Description:** This register is used in fiber MAC for auto-negotiation only. The contents of this register are sent as the config_word. The contents of this register are the "config_reg" sent to the link partner, as described in IEEE 802.3 2000 Edition, subclause 37.2.1. | | | | 0x000001A0 |
| 31:16 | Reserved | Reserved | RO | 0x0000 |
| 15 | Next Page | Next Page request | R/W | 0 |
| 14 | Reserved | Write as 0, ignore on read | R/W | 0 |

1.  RO = Read Only, No clear on Read; R = Read, Clear on Read; W = Write only; R/W = Read/Write, No clear; R/W/C = Read/Write, Clear on Write
*Note:*    A value of 0x0 must be written to all reserved bits of the *Section 89, TX Config Word ($ Port_Index + 0x17)* Register.

**Table 89    TX Config Word ($ Port_Index + 0x17) (Sheet 2 of 2)**

| Bit | Name | Description | Type[1] | Default |
|---|---|---|---|---|
| 13:12[2] | Remote Fault [1:0] | Remote fault definitions:<br>00 = No error, link okay<br>01 = Offline<br>10 = Link failure<br>11 = Auto-negotiation_Error | R/W | 00 |
| 11:9 | Reserved | Write as 0, ignore on Read | R/W | 000 |
| 8 | Asym Pause | Asym Pause. The ability to send pause frames. | R/W | 1 |
| 7 | Sym Pause | Sym Pause. The ability to send and receive pause frames. | R/W | 1 |
| 6 | Half Duplex | Half-duplex | R/W | 0 |
| 5 | Full Duplex | Full-duplex | R/W | 1 |
| 4:0 | Reserved | Write as 0, ignore on read | R/W | 0x00 |

1. RO = Read Only, No clear on Read; R = Read, Clear on Read; W = Write only; R/W = Read/Write, No clear; R/W/C = Read/Write, Clear on Write
*Note:* A value of 0x0 must be written to all reserved bits of the Section 89, *TX Config Word ($ Port_Index + 0x17)* Register.

**Table 90    Diverse Config Write ($ Port_Index + 0x18) (Sheet 1 of 2)**

| Bit | Name | Description | Type[1] | Default |
|---|---|---|---|---|
| **Register Description:** This register contains various configuration bits for general use. | | | | 0x00110D |
| 31:19 | Reserved | Reserved | RO | 0x0000 |
| 18:13 | Reserved | Write as 0, ignore on Read. | R/W | 0x0000 |
| 12 | Reserved[2] | Write as 1, ignore on Read. | R/W | 1 |
| 11-9 | Reserved[2] | Write as 0, ignore on Read. | R/W | 0x0 |
| 8 | Reserved[2] | Write as 1, ignore on Read. | R/W | 1 |
| 7 | pad_enable | 0 = Normal operation<br>1 = Enable padding of undersized packets<br><br>*Note:* Assertion of this bit results in the automatic addition of a CRC to the padded packet. | R/W | 0 |
| 6 | crc_add | 0 = Normal operation<br>1 = Enable automatic CRC appending | R/W | 0 |
| 5 | AN_enable | Enable auto-negotiation (used for fiber mode only) to be performed by the hardware state machines in the MAC.<br>The hardware auto-negotiation (AN) state machine controls the config words transmitted when this bit is set.<br>*Note:* In copper mode, this bit must be set to 0 (reserved). | R/W | 0 |
| 4[2] | Reserved | Write as 0, ignore on Read. | R/W | 0 |

1. RO = Read Only, No clear on Read; R = Read, Clear on Read; W = Write only; R/W = Read/Write, No clear; R/W/C = Read/Write, Clear on Write
2. Reserved bits must be written to the default value for proper operation.

**Table 90** **Diverse Config Write ($ Port_Index + 0x18) (Sheet 2 of 2)**

| Bit | Name | Description | Type[1] | Default |
|---|---|---|---|---|
| 3:2[2] | Reserved | Write as 1, ignore on Read. | R/W | 11 |
| 1[2] | Reserved | Write as 0, ignore on Read. | R/W | 0 |
| 0[2] | Reserved | Write as 1, ignore on Read. | R/W | 1 |
| 1. RO = Read Only, No clear on Read; R = Read, Clear on Read; W = Write only; R/W = Read/Write, No clear; R/W/C = Read/Write, Clear on Write<br>2. Reserved bits must be written to the default value for proper operation. | | | | |

**Table 91** **RX Packet Filter Control ($ Port_Index + 0x19) (Sheet 1 of 2)**

| Bit | Name | Description | Type[1] | Default |
|---|---|---|---|---|
| **Register Description:** This register allows for specific packet types to be marked for filtering and is used in conjunction with the Section 125, *RX FIFO Errored Frame Drop Counter Ports 0 - 3 ($0x5A2 - 0x5A5)*. | | | | 0x00000000 |
| 31:6 | Reserved | Reserved | | 0 |
| 5 | CRC Error Pass | This bit enables a Global filter on frames with a CRC Error.<br>0 = When CRC Error Pass = 0, all frames with a CRC Error are marked as bad.[2]<br>1 = Frames with a CRC Error are not marked as bad and are passed to the SPI3 interface for transfer as good frames, regardless of the state of the bits in the Section 123, *RX FIFO Errored Frame Drop Enable ($0x59F)*.<br><br>*Note:* When the CRC Error Pass Filter bit = 0, it takes precedence over the other filter bits. Any packet, whether is a Pause, Unicast, Multicast or Broadcast packet with a CRC error, is marked as a bad frame when CRC Error Pass = 0 | R/W | 0 |
| 4 | Pause Frame Pass | This bit enables a Global filter on Pause frames.<br>0 = All pause frames are dropped.[2]<br>1 = All pause frames are passed to the SPI3 Interface.<br><br>*Note:* Pause Frames can only be filtered if RXFD flow control is enabled in the Section 84, *FC Enable ($ Port_Index + 0x12)*. | R/W | 0 |
| 3 | VLAN Drop En | This bit enables a global filter on VLAN frames.<br>0 = All VLAN frames are passed to the SPI3 Interface.<br>1 = All VLAN frames are dropped.[2] | R/W | 0 |
| 1. RO = Read Only, No clear on Read; R = Read, Clear on Read; W = Write only; R/W = Read/Write, No clear; R/W/C = Read/Write, Clear on Write<br>2. Used in conjunction with the Section Table 123, *RX FIFO Errored Frame Drop Enable ($0x59F), on page 189*. This allows the frame to be dropped in the RX FIFO. Otherwise, the frame is sent out the SP3 interface and may be optionally signaled with an RERR (see bit 0 of Section 147, *SPI3 Receive Configuration ($0x701)*. | | | | |

**Table 91     RX Packet Filter Control ($ Port_Index + 0x19) (Sheet 2 of 2)**

| Bit | Name | Description | Type[1] | Default |
|---|---|---|---|---|
| 2 | B/Cast Drop En | This bit enables a Global filter on broadcast frames.<br>0 = All broadcast frames are passed to the SPI3 Interface.<br>1 = All broadcast frames are dropped.[2] | R/W | 0 |
| 1 | M/Cast Match En | This bit enables a filter on multicast frames.<br>0 = All muticast frames are good and passed to the SPI3 Interface.<br>1 = Only multicast frames with a destination address that matches the PortMulticastAddress are forwarded. All other muticast frames are dropped.[2] | R/W | 0 |
| 0 | U/Cast Match En[2] | This bit enables a filter on unicast frames.<br>0 = All unicast frames are good and are passed to the SPI3 Interface.<br>1 = Only unicast frames with a Destination Address that matches the Station Address are forwarded. All other unicast frames are dropped.[2]<br><br>*Note:* The VLAN filter overrides the unicast filter. Therefore, a VLAN frame cannot be filtered based on the unicast address. | R/W | 0 |

1. RO = Read Only, No clear on Read; R = Read, Clear on Read; W = Write only; R/W = Read/Write, No clear; R/W/C = Read/Write, Clear on Write
2. Used in conjunction with the Section Table 123, *RX FIFO Errored Frame Drop Enable ($0x59F)*, on page 189. This allows the frame to be dropped in the RX FIFO. Otherwise, the frame is sent out the SP3 interface and may be optionally signaled with an RERR (see bit 0 of Section 147, *SPI3 Receive Configuration ($0x701)*.

**Table 92     Port Multicast Address ($ Port_Index +0x1A − +0x1B)**

| Name | Description | Address | Type[*] | Default |
|---|---|---|---|---|
| Port Multicast Address Low | This address compares against multicast frames at the receiving side if multicast filtering is enabled.<br>This register contains bits 31:0 of the address. | Port_Index + 0x1A | R/W | 0x0000000 |
| Port Multicast Address High | This address compares against multicast frames at the receiving side if Multicast filtering is enabled.<br>This register contains bits 47:32 of the address. | Port_Index + 0x1B | R/W | 0x00000000 |

1. RO = Read Only, No clear on Read; R = Read, Clear on Read; W = Write only; R/W = Read/Write, No clear; R/W/C = Read/Write, Clear on Write

## 8.4.2     MAC RX Statistics Register Overview

The MAC RX Statistics registers contain the MAC receiver statistic counters and are cleared when read. The software polls these registers and accumulates values to ensure that the counters do not wrap. The 32-bit counters wrap after approximately 30 seconds.

Table 93 covers the RX statistics for the four MAC ports. Port_Index is the port number (0, 1, 2, or 3).

**Table 93    MAC RX Statistics ($ Port_Index + 0x20 − + 0x39) (Sheet 1 of 4)**

| Name | Description | Address | Type[1] | Default |
|---|---|---|---|---|
| RxOctetsTotalOK | Counts the bytes received in all legal frames, including all bytes from the destination MAC address to and including the cyclic redundancy check (CRC). The initial preamble and Start of Frame Delimiter (SFD) bytes are not counted. | Port_Index + 0x20 | R | 0x00000000 |
| RxOctetsBAD[2] | Counts the bytes received in all bad frames with legal size (frames with CRC error, alignment errors, or code violations), including all bytes from the destination MAC address to (and including) the CRC. The initial preamble and SFD bytes are not counted. Frames with illegal size do not add to this counter (shorts, runts, longs, jabbers, and very longs). *Note:* This register does not count octets on undersized received packets. | Port_Index + 0x21 | R | 0x00000000 |
| RxUCPkts | The total number of unicast packets received (excluding bad packets). *Note:* This count includes non-pause control and VLAN packets, which are also counted in other counters. These packet types are counted twice. Take care when summing register counts for reporting Management Information Base (MIB) information. | Port_Index + 0x22 | R | 0x00000000 |
| RxMCPkts | The total number of multicast packets received (excluding bad packets) *Note:* This count includes pause control packets, which are also counted in the PauseMacControl-ReceivedCounter. These packet types are counted twice. Take care when summing register counts for reporting MIB information. | Port_Index + 0x23 | R | 0x00000000 |
| RxBCPkts | The total number of Broadcast packets received (excluding bad packets). | Port_Index + 0x24 | R | 0x00000000 |
| RxPkts64Octets | The total number of packets received (including bad packets) that were 64 octets in length. Incremented for tagged packets with a length of 64 bytes, including tag field. | Port_Index + 0x25 | R | 0x00000000 |
| RxPkts65to127 Octets | The total number of packets received (including bad packets) that were 65-127 octets in length. Incremented for tagged packets with a length of 65-127 bytes, including tag field. | Port_Index + 0x26 | R | 0x00000000 |
| RxPkts128t0255 Octets | The total number of packets received (including bad packets) that were 128-255 octets in length. Incremented for tagged packets with a length of 128-255 bytes, including tag field. | Port_Index + 0x27 | R | 0x00000000 |

1. RO = Read Only, No clear on Read; R = Read, Clear on Read; W = Write only; R/W = Read/Write, No clear; R/W/C = Read/Write, Clear on Write
2. When sending in large frames, the counters can only handle certain limits. The behavior of the LongErrors and VeryLongErrors counters is as follows: VeryLongErrors counts frames that are 2*maxframesize, dependent upon where maxframesize is set. If maxframesize sets greater than half of the available count in RxOctetsBad (2^14-1), VeryLongErrors is never incremented, but LongErrors is incremented. This is due to a limitation in the counter size, which means that an accurate count will not occur in the RxOctetsBAD counter if the frame is larger than 2^14-1.
3. This register is relevant only when configured for copper operation.
4. This register is relevant only when configured for fiber operation (line side interface is SerDes).

**Table 93     MAC RX Statistics ($ Port_Index + 0x20 − + 0x39) (Sheet 2 of 4)**

| Name | Description | Address | Type[1] | Default |
|---|---|---|---|---|
| RxPkts256to511 Octets | The total number of packets received (including bad packets) that were 256-511 octets in length. Incremented for tagged packets with a length of 256-511 bytes, including tag field. | Port_Index + 0x28 | R | 0x00000000 |
| RxPkts512to1023O ctets | The total number of packets received (including bad packets) that were 512-1023 octets in length. Incremented for tagged packets with a length of 512-1023 bytes, including tag field. | Port_Index + 0x29 | R | 0x00000000 |
| RxPkts1024to1518 Octets | The total number of packets received (including bad packets) that were 1024-1518 octets in length. Incremented for tagged packet with a length between 1024-1522, including the tag. | Port_Index + 0x2A | R | 0x00000000 |
| RxPkts1519toMaxO ctets | The total number of packets received (including bad packets) that were greater than 1518 octets in length. Incremented for tagged packet with a length between 1523-max frame size, including the tag. | Port_Index + 0x2B | R | 0x00000000 |
| RxFCSErrors | Number of frames received with legal size, but with wrong CRC field (also called Frame Check Sequence (FCS) field). *Note:* Legal size is 64 bytes through the value programmed in the Section Table 81, Max Frame Size (Addr: Port_Index + 0x0F), on page 159. | Port_Index + 0x2C | R | 0x00000000 |
| RxTagged | Number of OK frames with VLAN tag. (Type field = 0x8100) | Port_Index + 0x2D | R | 0x00000000 |
| RxDataError[3] | Number of frames received with legal length, containing a code violation (signaled with RX_ERR on RGMII). | Port_Index + 0x2E | R | 0x00000000 |
| RxAlignErrors[3] | Frames with a legal frame size, but containing less than eight additional bits. This occurs when the frame is not byte aligned. The CRC of the frame is wrong when the additional bits are stripped. If the CRC is OK, then the frame is not counted but treated as an OK frame. This counter increments in 10 Mbps or 100 Mbps RGMII mode only. *Note:* This counter increments in 10 or 100 Mbps RGMII mode only. | Port_Index + 0x2F | R | 0x00000000 |

1. RO = Read Only, No clear on Read; R = Read, Clear on Read; W = Write only; R/W = Read/Write, No clear; R/W/C = Read/Write, Clear on Write
2. When sending in large frames, the counters can only handle certain limits. The behavior of the LongErrors and VeryLongErrors counters is as follows: VeryLongErrors counts frames that are 2*maxframesize, dependent upon where maxframesize is set. If maxframesize sets greater than half of the available count in RxOctetsBad (2^14-1), VeryLongErrors is never incremented, but LongErrors is incremented. This is due to a limitation in the counter size, which means that an accurate count will not occur in the RxOctetsBAD counter if the frame is larger than 2^14-1.
3. This register is relevant only when configured for copper operation.
4. This register is relevant only when configured for fiber operation (line side interface is SerDes).

**Table 93     MAC RX Statistics ($ Port_Index + 0x20 − + 0x39) (Sheet 3 of 4)**

| Name | Description | Address | Type[1] | Default |
|---|---|---|---|---|
| RxLongErrors[2] | Frames bigger than the maximum allowed, with both OK CRC and the integral number of octets. Default maximum allowed is 1518 bytes untagged and 1522 bytes tagged, but the value can be changed by a register. Frames bigger than the larger of 2*maxframesize and 50,000 bits are not counted here, but they are counted in the VeryLongError counter. | Port_Index + 0x30 | R | 0x00000000 |
| RxJabberErrors | Frames bigger than the maximum allowed, with either a bad CRC or a non-integral number of octets. The default maximum allowed is 1518 bytes untagged and 1522 bytes tagged, but the value can be changed by a register. Frames bigger than the larger of 2*maxframesize and 50,000 bits are not counted here, but they are counted in the VeryLongError counter. | Port_Index + 0x31 | R | 0x00000000 |
| RxPauseMacControlReceivedCounter | Number of Pause MAC control frames received. This statistic register increments on any valid 64-byte pause frame with a valid CRC and also increments on a 64-byte pause frame with an invalid CRC if bit 5 of the Section 91, RX Packet Filter Control ($ Port_Index + 0x19) is set to 1. | Port_Index + 0x32 | R | 0x00000000 |
| RxUnknownMac ControlFrame Counter | Number of MAC control frames received with an op code different from 0001 (Pause). | Port_Index + 0x33 | R | 0x00000000 |
| RxVeryLongErrors[2] | Frames bigger than the larger of 2*maxframesize and 50,000 bits | Port_Index + 0x34 | R | 0x00000000 |

1.  RO = Read Only, No clear on Read; R = Read, Clear on Read; W = Write only; R/W = Read/Write, No clear; R/W/C = Read/Write, Clear on Write
2.  When sending in large frames, the counters can only handle certain limits. The behavior of the LongErrors and VeryLongErrors counters is as follows: VeryLongErrors counts frames that are 2*maxframesize, dependent upon where maxframesize is set. If maxframesize sets greater than half of the available count in RxOctetsBad (2^14-1), VeryLongErrors is never incremented, but LongErrors is incremented. This is due to a limitation in the counter size, which means that an accurate count will not occur in the RxOctetsBAD counter if the frame is larger than 2^14-1.
3.  This register is relevant only when configured for copper operation.
4.  This register is relevant only when configured for fiber operation (line side interface is SerDes).

**Table 93     MAC RX Statistics ($ Port_Index + 0x20 − + 0x39) (Sheet 4 of 4)**

| Name | Description | Address | Type[1] | Default |
|---|---|---|---|---|
| RxRuntErrors[3] | The total number of packets received that are less than 64 octets in length, but longer than or equal to 96 bit times, which corresponds to a 4-byte frame with a well-formed preamble and SFD. This is the shortest fragment and can be transmitted in case of a collision event on a half-duplex segment. This counter indicates fragment sizes, which is expected on half-duplex segments but not on full-duplex links, and the counter is only fully updated after receipt of a good frame following a fragment.<br>*Note:* The ShortRuntsThreshold register controls the byte count used to determine the difference between Runts and Shorts and therefore controls which counter is incremented for a given frame size. This counter is only updated after receipt of two good frames.<br>*Note:* This counter is only valid when the selected port within the IXF1104 MAC is operating in copper (RGMII or GMII) mode. The RuntError counter is not updated when the selected port within the IXF1104 MAC is configured to operated in fiber (SerDes) mode. | Port_Index + 0x35 | R | 0x00000000 |
| RxShort Errors[3] | The total number of packets received that are less than 96 bit times, which corresponds to a 4-byte frame with a well-formed preamble and SFD. This counter indicates fragment sizes illegal in all modes and is only fully updated after reception of a good frame following a fragment.<br>*Note:* This register is only relevant when the IXF1104 MAC port is configured for copper operation (the line side interface is configured for either RGMII or GMII operation). This register will not increment when the IXF1104 MAC port is configured for fiber operation using the SerDes interface. | Port_Index + 0x36 | R | 0x00000000 |
| RxCarrier Extend Error | **Not applicable.** | Port_Index + 0x37 | R | 0x00000000 |
| RxSequenceErrors[4] | Records the number of sequencing errors that occur in fiber mode. | Port_Index + 0x38 | R | 0x00000000 |
| RxSymbolErrors[4] | Records the number of symbol errors encountered by the PHY. | Port_Index + 0x39 | R | 0x00000000 |

1. RO = Read Only, No clear on Read; R = Read, Clear on Read; W = Write only; R/W = Read/Write, No clear; R/W/C = Read/Write, Clear on Write
2. When sending in large frames, the counters can only handle certain limits. The behavior of the LongErrors and VeryLongErrors counters is as follows: VeryLongErrors counts frames that are 2*maxframesize, dependent upon where maxframesize is set. If maxframesize sets greater than half of the available count in RxOctetsBad (2^14-1), VeryLongErrors is never incremented, but LongErrors is incremented. This is due to a limitation in the counter size, which means that an accurate count will not occur in the RxOctetsBAD counter if the frame is larger than 2^14-1.
3. This register is relevant only when configured for copper operation.
4. This register is relevant only when configured for fiber operation (line side interface is SerDes).

### 8.4.3 MAC TX Statistics Register Overview

The MAC TX Statistics registers contain all the MAC transmit statistic counters and are cleared when read. The software must poll these registers to accumulate values and to ensure that the counters do not wrap. The 32-bit counters wrap after approximately 30 seconds.

Table 94 covers all four MAC ports TX statistics. Port_Index is the port number (0, 1, 2, or 3).

**Table 94    MAC TX Statistics ($ Port_Index +0x40 − +0x58) (Sheet 1 of 4)**

| Name | Description | Address | Type[1] | Default |
|------|-------------|---------|---------|---------|
| OctetsTransmittedOK | Counts the bytes transmitted in all legal frames. The count includes all bytes from the destination MAC address to and including the CRC. The initial preamble and SFD bytes are not counted. Any initial collided transmission attempts before a successful frame transmission do not add to this counter. | Port_Index + 0x40 | R | 0x00000000 |
| OctetsTransmittedBad | Counts the bytes transmitted in all bad frames. The count includes all bytes from the destination MAC address to and including the CRC. The initial preamble and SFD bytes are not counted. Late collision counted: The count is close to the actual number of bytes transmitted before the frame is discarded. Excessive collision counted: The count is close to the actual number of bytes transmitted before the frame is discarded. TX under-run counted: The count is expected to match the number of bytes actually transmitted before the frame is discarded. TX CRC error counted: All bytes not sent with success are counted by this counter. Any initial collided transmission attempts before a successful frame transmission do not add to this counter. | Port_Index + 0x41 | R | 0x00000000 |
| TxUCPkts | The total number of unicast packets transmitted (excluding bad packets). | Port_Index + 0x42 | R | 0x00000000 |

1. RO = Read Only, No clear on Read; R = Read, Clear on Read; W = Write only; R/W = Read/Write, No clear; R/W/C = Read/Write, Clear on Write

**Table 94**     **MAC TX Statistics ($ Port_Index +0x40 − +0x58) (Sheet 2 of 4)**

| Name | Description | Address | Type[1] | Default |
|---|---|---|---|---|
| TxMCPkts | The total number of multicast packets transmitted (excluding bad packets). *Note:* This count includes pause control packets, which are also counted in the TxPauseFrames Counter. Thus, these types of packets are counted twice. Take care when summing register counts for reporting MIB information. | Port_Index + 0x43 | R | 0x00000000 |
| TxBCPkts | The total number of broadcast packets transmitted (excluding bad packets). | Port_Index + 0x44 | R | 0x00000000 |
| TxPkts64Octets | The total number of packets transmitted (including bad packets) that were 64 octets in length. Incremented for tagged packets with a length of 64 bytes, including tag field. | Port_Index + 0x45 | R | 0x00000000 |
| Txpkts65to127Octets | The total number of packets transmitted (including bad packets) that were 65-127 octets in length. Incremented for tagged packets with a length of 65-127 bytes, including tag field. | Port_Index + 0x46 | R | 0x00000000 |
| Txpkts128to255Octets | The total number of packets transmitted (including bad packets) that were 128-255 octets in length. Incremented for tagged packets with a length of 128-255 bytes, including tag field. | Port_Index + 0x47 | R | 0x00000000 |
| Txpkts256to511Octets | The total number of packets transmitted (including bad packets) that were 256-511 octets in length. Incremented for tagged packets with a length of 256-511 bytes, including tag field. | Port_Index + 0x48 | R | 0x00000000 |
| Txpkts512to1023Octets | The total number of packets transmitted (including bad packets) that were 512-1023 octets in length. Incremented for tagged packets with a length of 512-1023 bytes, including tag field. | Port_Index + 0x49 | R | 0x00000000 |
| Txpkts1024to1518Octets | The total number of packets transmitted (including bad packets) that were 1024-1518 octets in length. Incremented for tagged packet with a length between 1024-1522, including the tag. | Port_Index + 0x4A | R | 0x00000000 |
| Txpkts1519toMaxOctets | The total number of packets transmitted (including bad packets) that were greater than 1518 octets in length. Incremented for tagged packet with a length between 1523 - max fame size, including the tag. | Port_Index + 0x4B | R | 0x00000000 |
| 1. RO = Read Only, No clear on Read; R = Read, Clear on Read; W = Write only; R/W = Read/Write, No clear; R/W/C = Read/Write, Clear on Write | | | | |

**Table 94     MAC TX Statistics ($ Port_Index +0x40 − +0x58) (Sheet 3 of 4)**

| Name | Description | Address | Type[1] | Default |
|------|-------------|---------|---------|---------|
| TxDeferred | Number of times the initial transmission attempt of a frame is postponed due to another frame already being transmitted on the Ethernet network. TxTotalCollisions.<br>***Note:***   NA - half-duplex only | Port_Index + 0x4C | R | 0x00000000 |
| TxTotalCollisions | Sum of all collision events.<br>***Note:***   NA - half-duplex only | Port_Index + 0x4D | R | 0x00000000 |
| TxSingleCollisions | A count of successfully transmitted frames on a particular interface where the transmission is inhibited by exactly one collision. A frame that is counted by an instance of this object is also counted by the corresponding instance of either the UnicastPkts, MulticastPkts, or BroadcastPkts, and is not counted by the corresponding instance of the MultipleCollisionFrames object.<br>***Note:***   NA - half-duplex only | Port_Index + 0x4E | R | 0x00000000 |
| TxMultipleCollisions | A count of successfully transmitted frames on a particular interface for which transmission is inhibited by more than one collision. A frame that is counted by an instance of this object is also counted by the corresponding instance of either the UnicastPkts, MulticastPkts, or BroadcastPkts, and is not counted by the corresponding instance of the SingleCollisionFrames object.<br>***Note:***   NA - half-duplex only | Port_Index + 0x4F | R | 0x00000000 |
| TxLateCollisions | The number of times a collision is detected on a particular interface later than 512 bit-times into the transmission of a packet. Such frame are terminated and discarded.<br>***Note:***   NA - half-duplex only | Port_Index + 0x50 | R | 0x00000000 |
| TxExcessiveCollisionErrors | A count of frames, which collides 16 times and is then discarded by the MAC. Not effecting xMultipleCollisions<br>***Note:***   NA - half-duplex only | Port_Index + 0x51 | R | 0x00000000 |
| TxExcessiveDeferralErrors | Number of times frame transmission is postponed more than 2*MaxFrameSize because of another frame already being transmitted on the Ethernet network. This causes the MAC to discard the frame.<br>***Note:***   NA - half-duplex only | Port_Index + 0x52 | R | 0x00000000 |
| TxExcessiveLengthDrop | Frame transmissions aborted by the MAC because the frame is longer than maximum frame size. These frames are truncated by the MAC when the maximum frame size violation is detected by the MAC. | Port_Index + 0x53 | R | 0x00000000 |

1.   RO = Read Only, No clear on Read; R = Read, Clear on Read; W = Write only; R/W = Read/Write, No clear; R/W/C = Read/Write, Clear on Write

**Table 94** **MAC TX Statistics ($ Port_Index +0x40 − +0x58) (Sheet 4 of 4)**

| Name | Description | Address | Type[1] | Default |
|---|---|---|---|---|
| TxUnderrun | Internal TX error that causes the MAC to end the transmission before the end of the frame because the MAC did not get the needed data in time for transmission. The frames are lost and a fragment or a CRC error is transmitted. | Port_Index + 0x54 | R | 0x00000000 |
| TxTagged | Number of OK frames with VLAN tag. (Type field = 0x8100). | Port_Index + 0x55 | R | 0x00000000 |
| TxCRCError | Number of frames transmitted with a legal size but with the wrong CRC field (also called FCS field). | Port_Index + 0x56 | R | 0x00000000 |
| TxPauseFrames | Number of pause MAC frames transmitted. | Port_Index + 0x57 | R | 0x00000000 |
| TxFlowControlCollisions Send | Intentionally generates collisions to curb reception of incoming traffic due to insufficient memory available for additional frames. The port must be in half-duplex mode with flow control enabled.<br>*Note:* To receive a correct statistic, a last frame may have to be transmitted after the last flow control collisions send.<br>*Note:* NA - half-duplex only | Port_Index + 0x58 | R | 0x00000000 |
| 1. RO = Read Only, No clear on Read; R = Read, Clear on Read; W = Write only; R/W = Read/Write, No clear; R/W/C = Read/Write, Clear on Write | | | | |

## 8.4.4 PHY Autoscan Registers

**Note:** These register hold the current values of the PHY registers only when Autoscan (see Section 5.5.8, *Autoscan Operation*, on page 102) is enabled and the IXF1104 MAC is configured in copper mode. These registers are not applicable in fiber mode.

**Table 95** **PHY Control ($ Port Index + 0x60) (Sheet 1 of 2)**

| Bit | Name | Description | Type[1] | Default |
|---|---|---|---|---|
| | | | | 0x00000010 001000 |
| 31:16 | Reserved | Reserved | RO | 0x0000 |
| 15 | Reset | PHY Soft Reset. Resets the PHY registers to their default value.<br>This register bit self-clears after the reset is complete.<br>0 = Normal Operation<br>1 = PHY reset | RO | 0 |
| 14 | Loopback | 0 = Disable loopback mode<br>1 = Enable loopback mode | RO | 0 |
| 1. RO = Read Only; RR = Clear on Read; W = Write; R/W = Read/Write<br>2. This register is ignored if auto-negotiation is enabled. | | | | |

**Table 95     PHY Control ($ Port Index + 0x60) (Sheet 2 of 2)**

| Bit | Name | Description | Type[1] | Default |
|---|---|---|---|---|
| 13 | Speed Selection | 0.6 (Speed<1> 0.13 (Speed<0>)<br>00 = 10 Mbps<br>01 = 100 Mbps<br>10 = 1000 Mbps (manual mode not allowed)<br>11 = Reserved | RO | $0^2$ |
| 12 | Auto-Negotiation Enable | 0 = Disable auto-negotiation process<br>1 = Enable auto-negotiation process<br>This register bit must be enabled for 1000BASE-T operation. | RO | 1 |
| 11 | Power-Down | 0 = Normal operation<br>1 = Power-down | RO | 0 |
| 10 | Isolate | 0 =<br>1 = Electrically isolate PHY from GMII | RO | 0 |
| 9 | Restart Auto-Negotiation | 0 = Normal operation<br>1 = Restart auto-negotiation process | RO | 0 |
| 8 | Duplex Mode | 0 = Half-duplex mode<br>1 = Full-duplex mode | RO | $1^2$ |
| 7 | Collision Test | 0 = Disable COL signal test<br>1 = Enable COL signal test<br>This register bit is ignored unless loopback is enabled (Register bit 0.14 = 1) | RO | 0 |
| 6 | Speed Selection 1000 Mbps | 0.6 (Speed<1>) 0.13 (Speed<0>)<br>00 = 10 Mbps<br>01 = 100 Mbps<br>10 = 1000 Mbps (manual mode now allowed)<br>11 = Reserved | RO | $0^2$ |
| 5:0 | Reserved | Reserved | RO | 0 |

1.  RO = Read Only; RR = Clear on Read; W = Write; R/W = Read/Write
2.  This register is ignored if auto-negotiation is enabled.

**Table 96     PHY Status ($ Port Index + 0x61) (Sheet 1 of 2)**

| Bit | Name | Description | Type[1] | Default |
|---|---|---|---|---|
|  |  |  |  | 0x00111100100001001 |
| 31:16 | Reserved | Reserved | RO | 0 |
| 15 | 100BASE-T4 | 0 = PHY not able to operate in 100BASE-T4<br>1 = PHY able to operate in 100BASE-T4 | RO | 0 |
| 14 | 100BASE-X Full-Duplex | 0 = PHY not able to operate in 100BASE-X in full-duplex mode<br>1 = PHY able to operate in 100BASE-X in full-duplex mode | RO | 1 |
| 13 | 100BASE-X Half-Duplex | 0 = PHY not able to operate in 100BASE-X in half-duplex mode<br>1 = PHY able to operate in 100BASE-X in half-duplex mode | RO | 1 |

1.  R = Read Only; RR = Clear on Read; W = Write; R/W = Read/Write

**Table 96      PHY Status ($ Port Index + 0x61) (Sheet 2 of 2)**

| Bit | Name | Description | Type[1] | Default |
|-----|------|-------------|---------|---------|
| 12 | 10 Mbps Full-Duplex | 0 = PHY not able to operate in 10 Mbps in full-duplex mode<br>1 = PHY able to operate in 10 Mbps in full-duplex mode | RO | 1 |
| 11 | 10 Mbps Half-Duplex | 0 = PHY not able to operate in 10 Mbps in half-duplex mode<br>1 = PHY able to operate in 10 Mbps in half-duplex mode | RO | 1 |
| 10 | 100BASE-T2 Full-Duplex | 0 = PHY not able to operate in 10BASE-T2 in full-duplex mode (not supported)<br>1 = PHY able to operate in 100BASE-T2 in full-duplex mode | RO | 0 |
| 9 | 100BASE-T2 Half-Duplex | 0 = PHY not able to operate in 100BASE-T2 in half-duplex mode<br>1 = PHY able to operate in 100BASE-T2 in half-duplex mode | RO | 0 |
| 8 | Extended Status | 0 = No extended status information in Register 15<br>1 = Extended status information in Register 15 | RO | 1 |
| 7 | Reserved | Reserved | RO | 0 |
| 6 | MF Preamble Suppression | 0 = PHY will not accept management frames with preamble suppressed<br>1 = PHY will accept management frames with preamble suppressed | RO | 0 |
| 5 | Reserved | Reserved | RO | 0 |
| 4 | Remote Fault | 0 =<br>1 = Remote fault condition detected | RO | 0 |
| 3 | Auto-Negotiation Ability | 0 =<br>1 = PHY is able to perform auto-negotiation | RO | 1 |
| 2 | Link Status | 0 = Link is down<br>1 = Link is up | RO | 0 |
| 1 | Jabber Detect | 0 = Jabber condition not detected<br>1 = Jabber condition detected | RO | 0 |
| 0 | Extended Capability | 0 = No extended register capabilities<br>1 = Extended register capabilities | RO | 1 |
| 1. R = Read Only; RR = Clear on Read; W = Write; R/W = Read/Write | | | | |

**Table 97      PHY Identification 1 ($ Port Index + 0x62)**

| Bit | Name | Description | Type[1] | Default |
|-----|------|-------------|---------|---------|
|  |  |  |  | 0x00013 |
| 31:16 | Reserved | Reserved | RO | 0 |
| 15:0 | PHY ID Number | The PHY identifier is composed of register bits 18.3 of the OUI (Organizationally Unique Identifier) | RO | h0013 |
| 1. RO = Read Only; RR = Clear on Read; W = Write; R/W = Read/Write | | | | |

**Table 98     PHY Identification 2 ($ Port Index + 0x63)**

| Bit | Name | Description | Type[1] | Default |
|---|---|---|---|---|
| | | | | 0x001111001 00000000 |
| 31:16 | Reserved | Reserved | RO | 0 |
| 15:10 | PHY ID Number | The PHY identifier is composed of register bits 24:19 of the OUI (Organizationally Unique Identifier) | RO | 011110 |
| 9:4 | Manufacturer's Model | Six bits containing the manufacturer's part number | RO | 010000 |
| 3:0 | Manufacturer's Revision Number | Four bits containing the manufacturer's revision number | RO | 0000 |
| 1.   RO = Read Only; RR = Clear on Read; W = Write; R/W = Read/Write | | | | |

**Table 99     Auto-Negotiation Advertisement ($ Port Index + 0x64) (Sheet 1 of 2)**

| Bit | Name | Description | Type[1] | Default |
|---|---|---|---|---|
| | | | | 0x00000100 111100001 |
| 31:16 | Reserved | Reserved | RO | 0 |
| 15 | Next Page | 0 = <br>1 = Manual control of Next Page (software) | RO | 0 |
| 14 | Reserved | Reserved | RO | 0 |
| 13 | Remote Fault | 0 = No remote fault<br>1 = Remote fault | RO | 0 |
| 12 | Reserved | Reserved | RO | 0 |
| 11 | ASM_DIR | Advertise Asymmetric Pause Direction register bit. This register bit is used in conjunction with Pause (Register bit 4.10)<br>0 = Link partner is not capable of asymmetric pause<br>1 = Link partner is capable of asymmetric pause | RO | 1 |
| 10 | Pause | Advertise to link partner that Pause operation is desired (IEEE 802.3x Standard) | RO | 0 |
| 9 | 100BASE-T4 | 0 = 100BASE-T4 capability is not available<br>1 = 100BASE-T4 capability is available<br>The IXF1104 MAC does not support 100BASE-T4, but allows this register bit to be set to advertise in auto-negotiation sequence for 100BASE-T4 operation. If this capability is desired, an external 100BASE-T4 transceiver can be switched in. | RO | 0 |
| 8 | 100BASE-TX Full-Duplex | 0 = DTE is not 100BASE-TX, full-duplex mode capable<br>1 = DTE is 100BASE-TX, full-duplex mode capable | RO | 1 |
| 7 | 100BASE-TX Half-Duplex | 0 = DTE is not 100BASE-TX, half-duplex mode capable<br>1 = DTE is 100BASE-TX, half-duplex mode capable | RO | 1 |
| 1.   RO = Read Only; RR = Clear on Read; W = Write; R/W = Read/Write | | | | |

**Table 99**     **Auto-Negotiation Advertisement ($ Port Index + 0x64) (Sheet 2 of 2)**

| Bit | Name | Description | Type[1] | Default |
|---|---|---|---|---|
| 6 | 10BASE-T Full-Duplex | 0 = DTE is not 10BASE-T, full-duplex mode capable<br>1 = DTE is 10BASE-T, full-duplex mode capable | RO | 1 |
| 5 | 10BASE-T Half-Duplex | 0 = DTE is not 10BASE-T, half-duplex mode capable<br>1 = DTE is 10BASE-T, half-duplex mode capable | RO | 1 |
| 4:0 | Selector Field, S[4:0] | 00001 =IEEE 802.3<br>00010 =IEEE 802.9 ISLAN-16T<br><br>Reserved for future auto-negotiation development<br>11111 = Reserved for future auto-negotiation development<br>Unspecified or reserved combinations should not be transmitted<br>Setting this field to a value other than 00001 will most likely cause auto-negotiation to fail | RO | 00001 |
| 1. RO = Read Only; RR = Clear on Read; W = Write; R/W = Read/Write | | | | |

**Table 100**     **Auto-Negotiation Link Partner Base Page Ability ($ Port Index + 0x65) (Sheet 1 of 2)**

| Bit | Name | Description | Type[1] | Default |
|---|---|---|---|---|
|  |  |  |  | 0x0---01001111000 01 |
| 31:16 | Reserved | Reserved | RO | 0 |
| 15 | Next Page | 0 = Link partner has no ability to send multiple pages<br>1 = Link partner has the ability to send multiple pages | RO | NA |
| 14 | Acknowledge | 0 = Link partner has not received Link Code Word from the IXF1104 MAC<br>1 = Link partner has received Link Code Word from the IXF1104 MAC | RO | NA |
| 13 | Remote Fault | 0 = No remote fault<br>1 = Remote fault | RO | NA |
| 12 | Reserved | Reserved | RO | 0 |
| 11 | ASM_DIR | Advertise Asymmetric Pause Direction Register bit. This register bit is used in conjunction with Pause (Register bit 4.10)<br>0 = Link partner is not capable of asymmetric pause<br>1 = Link partner is capable of asymmetric pause | RO | 1 |
| 10 | Link Partner Pause | Link partner wants to utilize Pause Operation as defined in IEEE 802.3x Standard | RO | 0 |
| 9 | 1000BASE-T4 | 0 = Link partner is not 100BASE-T4 capable<br>1 = Link partner is 100BASE-T4 capable | RO | 0 |
| 1. RO = Read Only; RR = Clear on Read; W = Write; R/W = Read/Write | | | | |

**Table 100    Auto-Negotiation Link Partner Base Page Ability ($ Port Index + 0x65) (Sheet 2 of 2)**

| Bit | Name | Description | Type[1] | Default |
|-----|------|-------------|---------|---------|
| 8 | 100BASE-TX Full-Duplex | 0 = Link partner is not 100BASE-TX, full-duplex mode capable<br>1 = Link partner is 100BASE-TX, full-duplex mode capable | RO | 1 |
| 7 | 100BASE-TX Half-Duplex | 0 = Link partner is not 100BASE-TX, half-duplex mode capable<br>1 = Link partner is 100BASE-TX, half-duplex mode capable | RO | 1 |
| 6 | 10BASE-T Full-Duplex | 0 = Link partner is not 10BASE-T, full-duplex mode capable<br>1 = Link partner is 10BASE-T, full-duplex mode capable | RO | 1 |
| 5 | 10BASE-T Half-Duplex | 0 = Link partner is not 10BASE-T, half-duplex mode capable<br>1 = Link partner is 10BASE-T, half-duplex mode capable | RO | 1 |
| 4:0 | Selector Field, S[4:0] | 00001 =IEEE 802.3<br>00010 =IEEE 802.9 ISLAN-16T<br>00000 =Reserved for future auto-negotiation development<br>11111 = Reserved for future auto-negotiation development<br>Unspecified or reserved combinations should not be transmitted<br><br>Setting this field to a value other than 00001 will most likely cause auto-negotiation to fail | RO | 00001 |
| 1. RO = Read Only; RR = Clear on Read; W = Write; R/W = Read/Write | | | | |

**Table 101    Auto-Negotiation Expansion ($ Port Index + 0x66)  (Sheet 1 of 2)**

| Bit | Name | Description | Type[1] | Default |
|-----|------|-------------|---------|---------|
| | | | | 0x0000000 |
| 31:6 | Reserved | Reserved | RO | 0 |
| 5 | Base Page | This register bit indicates the status of the auto-negotiation variable, base page. It flags synchronization with the auto-negotiation state diagram allowing detection of interrupted links.<br>This register bit is only used if Register bit 16.1 (alternate Next Page feature) is set.<br>0 = base_page = false<br>1 = base_page = true | RO | 0 |
| 4 | Parallel Detection Fault | 0 = Parallel detection fault has not occurred<br>1 = Parallel detection fault has occurred | RO | 0 |
| 3 | Link Partner Next Page Able | 0 = Link partner is not Next Page able<br>1 = Link partner is Next Page able | RO | 0 |
| 1. RO = Read Only; RR = Clear on Read; W = Write; R/W = Read/Write | | | | |

**Table 101    Auto-Negotiation Expansion ($ Port Index + 0x66) (Continued) (Sheet 2 of 2)**

| Bit | Name | Description | Type[1] | Default |
|---|---|---|---|---|
| 2 | Next Page Able | 0 = Local device is not Next Page able<br>1 = Local device is Next Page able | RO | 0 |
| 1 | Page Received | Indicates that a new page has been received and the received code word has been loaded into Register 5 (base pages) or Register 8 (next pages) as specified in the EEE 802.3 Standard.<br>This bit clears on Read. | RO | 0 |
| 0 | Link Partner Auto-Negotiation Able | 0 = Link partner is not auto-negotiation able<br>1 = Link partner is auto-negotiation able | RO | 0 |
| 1. RO = Read Only; RR = Clear on Read; W = Write; R/W = Read/Write | | | | |

**Table 102    Auto-Negotiation Next Page Transmit ($ Port Index + 0x67)**

| Bit | Name | Description | Type[1] | Default |
|---|---|---|---|---|
| | | | | 0x0000000 |
| 31:16 | Reserved | Reserved | RO | 0 |
| 15 | Next Page (NP) | 0 = Last page<br>1 = Additional Next Pages follow | RO | 0 |
| 14 | Reserved | Reserved | RO | 0 |
| 13 | Message Page (MP) | 0 = Unformatted page<br>1 = Message page | RO | 0 |
| 12 | Acknowledge 2 | 0 = Cannot comply with message<br>1 = Complies with message | RO | 0 |
| 11 | Toggle (T) | 0 = Previous value of the transmitted Link Code Word was logic one<br>1 = Previous value of the transmitted Link Code Word was logic zero | RO | 0 |
| 10:0 | Message/Unformatted Code Field | 11-bit message code field<br>See IEEE 802.3 Annex 28C | RO | 0 |
| 1. RO = Read Only; RR = Clear on Read; W = Write; R/W = Read/Write | | | | |

## 8.4.5    Global Status and Configuration Register Overview

Table 103 through Table 112, *JTAG ID ($0x50C), on page 185* provide an overview for the Global Control and Status Registers.

**Table 103      Port Enable ($0x500)**

| Bit | Name | Description | Type* | Default |
|---|---|---|---|---|
| **Register Description:** A control register for each port in the IXF1104 MAC. Port ID = bit position in the register. To make a port active, the bit must be set High. For example, Port 2 active implies a register value of 0000.0100. Setting the bit to 0 de-asserts the enable. The default state for this register is for all four ports to be disabled. | | | | 0x00000000 |
| 31:4 | Reserved | Reserved | RO | 0x0000000 |
| 3 | Port 3 Enable | Port 3<br>0 = Disable<br>1 = Enable | R/W | 0 |
| 2 | Port 2 Enable | Port 2<br>0 = Disable<br>1 = Enable | R/W | 0 |
| 1 | Port 1 Enable | Port 1<br>0 = Disable<br>1 = Enable | R/W | 0 |
| 0 | Port 0 Enable | Port 0<br>0 = Disable<br>1 = Enable | R/W | 0 |
| 1.  RO = Read Only, No clear on Read; R = Read, Clear on Read; W = Write only; R/W = Read/Write, No clear; R/W/C = Read/Write, Clear on Write | | | | |

**Table 104      Interface Mode ($0x501)**

| Bit | Name | Description | Type[1] | Default |
|---|---|---|---|---|
| **Register Description:** If_Mode – Four bits of this register determines the PHY interface mode.<br>0 = Fiber (SerDes/OMI interface)<br>1 = Copper (GMII or RGMII interface)<br>Changes to the data setting of this register must be made in conjunction with the *Clock and Interface Mode Change Enable Ports 0 - 3 ($0x794)* to ensure a safe transition to a new operational mode (see Section 6.1, *Change Port Mode Initialization Sequence*, on page 127).<br>The Enable clock mode change bit has to be set back to 1 after the configuration change takes effect. | | | | 0x00000000 |
| 31:4 | Reserved | Reserved | RO | 0x0000000 |
| 3 | Port 3 Interface Mode | 0 = Fiber mode<br>1 = Copper mode | R/W | 0 |
| 2 | Port 2 Interface Mode | 0 = Fiber mode<br>1 = Copper mode | R/W | 0 |
| 1 | Port 1 Interface Mode | 0 = Fiber mode<br>1 = Copper mode | R/W | 0 |
| 0 | Port 0 Interface Mode | 0 = Fiber mode<br>1 = Copper mode | R/W | 0 |
| 1.  RO = Read Only, No clear on Read; R = Read, Clear on Read; W = Write only; R/W = Read/Write, No clear; R/W/C = Read/Write, Clear on Write | | | | |

**Table 105    Link LED Enable ($0x502)**

| Bit | Name | Description | Type[1] | Default |
|---|---|---|---|---|
| **Register Description:** Per port bit should be set upon detection of link to enable proper operation of the link LEDs. | | | | 0x00000000 |
| 31:4 | Reserved | Reserved | R/W | 0x00000 |
| 3 | Link LED Enable Port 3 | Port 3 link<br>0 = No link<br>1 = Link | R/W | 0 |
| 2 | Link LED Enable Port 2 | Port 2 link<br>0 = No link<br>1 = Link | R/W | 0 |
| 1 | Link LED Enable Port 1 | Port 1 link<br>0 = No link<br>1 = Link | R/W | 0 |
| 0 | Link LED Enable Port 0 | Port 0 link<br>0 = No link<br>1 = Link | R/W | 0 |
| 1.   RO = Read Only, No clear on Read; R = Read, Clear on Read; W = Write only; R/W = Read/Write, No clear; R/W/C = Read/Write, Clear on Write | | | | |

**Table 106    MAC Soft Reset ($0x505)**

| Bit | Name | Description | Type[1] | Default |
|---|---|---|---|---|
| **Register Description:** Per-port software-activated reset of the MAC core. | | | | 0x00000000 |
| 31:4 | Reserved | Reserved | R/W | 0x00000 |
| 3 | Software Reset MAC 3 | Port 3<br>0 = Reset inactive<br>1 = Enable | R/W | 0 |
| 2 | Software Reset MAC 2 | Port 2<br>0 = Reset inactive<br>1 = Enable | R/W | 0 |
| 1 | Software Reset MAC 1 | Port 1<br>0 = Reset inactive<br>1 = Enable | R/W | 0 |
| 0 | Software Reset MAC 0 | Port 0<br>0 = Reset inactive<br>1 = Enable | R/W | 0 |
| 1.   RO = Read Only, No clear on Read; R = Read, Clear on Read; W = Write only; R/W = Read/Write, No clear; R/W/C = Read/Write, Clear on Write | | | | |

### Table 107    MDIO Soft Reset ($0x506)

| Bit | Name | Description | Type[1] | Default |
|---|---|---|---|---|
| **Register Description:** Software-activated reset of the MDIO module. | | | | 0x00000000 |
| 31:1 | Reserved | Reserved | RO | 0x00000000 |
| 0 | Software MDIO Reset | 0 = Reset inactive<br>1 = Reset active | R/W | 0 |
| 1.   RO = Read Only, No clear on Read; R = Read, Clear on Read; W = Write only; R/W = Read/Write, No clear; R/W/C = Read/Write, Clear on Write | | | | |

### Table 108    CPU Interface ($0x508)

| Bit | Name | Description | Type[1] | Default |
|---|---|---|---|---|
| **Register Description:** CPU Interface Endian select. Allows the user to select the Endian of the CPU interface to allow for various CPUs to be connected to the IXF1104 MAC. | | | | 0x00000000 |
| 31:25 | Reserved | Reserved | RO | 0x00 |
| 24 | CPU Endian | Reserved in Little Endian<br>Valid in Big endian<br>0 = Little Endian<br>1 = Big Endian | R/W | 0 |
| 23:1 | Reserved | Reserved | RO | 0x000000 |
| 0 | CPU Endian Control | Reserved in Big Endian<br>Valid in Little Endian<br>0 = Little Endian<br>1 = Big Endian | R/W | 0 |
| 1.   RO = Read Only, No clear on Read; R = Read, Clear on Read; W = Write only; R/W = Read/Write, No clear; R/W/C = Read/Write, Clear on Write<br>*Note:*    Since the Endianess of the bus is unknown when writing to this register, write 0x01000001 to set the bit and 0x0 to clear it. | | | | |

### Table 109    LED Control ($0x509)

| Bit | Name | Description | Type[1] | Default |
|---|---|---|---|---|
| **Register Description:** Global selection of LED mode. | | | | 0x00000000 |
| 31:2 | Reserved | Reserved | RO | 0x00000000 |
| 1 | LED Enable | 0 = Disable LED Block<br>1 = Enable LED Block | R/W | 0 |
| 0 | LED Control | 0 = Enable LED Mode 0 for use with SGS<br>      Thomson M5450 LED driver (Default)<br>1 = LED Mode 1 for use with Standard Octal Shift<br>      register | R/W | 0 |
| 1.   RO = Read Only, No clear on Read; R = Read, Clear on Read; W = Write only; R/W = Read/Write, No clear; R/W/C = Read/Write, Clear on Write | | | | |

**Table 110     LED Flash Rate ($0x50A)**

| Bit | Name | Description | Type[1] | Default |
|---|---|---|---|---|
| **Register Description:** Global selection of LED flash rate. | | | | 0x00000000 |
| 31:3 | Reserved | Reserved | RO | 0x00000000 |
| 2:0 | LED Flash Rate Control | 000 = 100 ms flash rate<br>001 = 200 ms flash rate<br>010 = 300 ms flash rate<br>011 = 400 ms flash rate<br>100 = 500 ms flash rate<br>101 = Reserved<br>110 = Reserved<br>111 = Reserved | R/W | 000 |
| 1. RO = Read Only, No clear on Read; R = Read, Clear on Read; W = Write only; R/W = Read/Write, No clear; R/W/C = Read/Write, Clear on Write | | | | |

**Table 111     LED Fault Disable ($0x50B)**

| Bit | Name | Description | Type[1] | Default |
|---|---|---|---|---|
| **Register Description:** Per-port fault disable. Disables the LED flashing for local or remote faults. | | | | 0x00000000 |
| 31:4 | Reserved | Reserved | RO | 0x0000000 |
| 3 | LED Port 3 Fault Control | Port 3<br>0 = Fault enabled<br>1 = Fault disabled | R/W | 0 |
| 2 | LED Port 2 Fault Control | Port 2<br>0 = Fault enabled<br>1 = Fault disabled | R/W | 0 |
| 1 | LED Port 1 Fault Control | Port 1<br>0 = Fault enabled<br>1 = Fault disabled | R/W | 0 |
| 0 | LED Port 0 Fault Control | Port 0<br>0 = Fault enabled<br>1 = Fault disabled | R/W | 0 |
| 1. RO = Read Only, No clear on Read; R = Read, Clear on Read; W = Write only; R/W = Read/Write, No clear; R/W/C = Read/Write, Clear on Write | | | | |

**Table 112    JTAG ID ($0x50C)**

| Bit | Name | Description | Type[1] | Default |
|-----|------|-------------|---------|---------|
| **Register Description:** The value of this register follows the same scheme as the device identification register found in the IEEE 1149.1 specification. The upper four bits correspond to silicon stepping.   The next 16 bits store a Part ID Number. The next 11 bits contain a JEDEC manufacturer ID. Bit zero = 1 if the chip is the first in a stack. The encoding scheme used for the Product ID field is implementation-dependent. | | | | 0x10450013 |
| 31:28 | Version | Version | RO | 0001[2] |
| 27:12 | Part ID | Part ID | RO | 0000010001 010000 |
| 11:8 | JEDEC Continuation Characters | JEDEC Continuation Characters | RO | 0000 |
| 7:1 | JEDEC ID | JEDEC ID | RO | 0001001 |
| 0 | Fixed | Fixed | RO | 1 |
| 1.  RO = Read Only, No clear on Read; R = Read, Clear on Read; W = Write only; R/W = Read/Write, No clear; R/W/C = Read/Write, Clear on Write<br>2.  These bits vary with stepping. | | | | |

## 8.4.6    RX FIFO Register Overview

Table 113 through Table 131 provide an overview of the RX FIFO registers, which include the RX FIFO High and Low watermarks.

**Table 113    RX FIFO High Watermark Port 0 ($0x580)**

| Bit | Name | Description | Type[1] | Default |
|-----|------|-------------|---------|---------|
| **Register Description:** The default value of 0x0E6 represents 230 eight-byte locations. This equates to 1840 bytes of data. A unit entry in this register equates to 8 bytes of data. When the amount of data stored in the RX FIFO exceeds the high watermark, flow control is automatically initiated within the MAC to avoid an overflow condition. | | | | 0x0E6 |
| 31:12 | Reserved | Reserved | RO | 0x00000 |
| 11: 0 | RX FIFO High Watermark Port 0 | The high water mark value.<br>*Note:*    Must be greater than the RX FIFO Low Watermark and RX FIFO transfer threshold. | R/W | 0x0E6 |
| 1.  RO = Read Only, No clear on Read; R = Read, Clear on Read; W = Write only; R/W = Read/Write, No clear; R/W/C = Read/Write, Clear on Write | | | | |

**Table 114    RX FIFO High Watermark Port 1 ($0x581)**

| Bit | Name | Description | Type[1] | Default |
|---|---|---|---|---|
| \* | **Register Description:** The default value of 0x0E6 represents 230 eight-byte locations. This equates to 1840 bytes of data. A unit entry in this register equates to 8 bytes of data. When the amount of data stored in the RX FIFO exceeds the high watermark, flow control is automatically initiated within the MAC to avoid an overflow condition. | | | 0x0E6 |
| 31:12 | Reserved | Reserved | RO | 0x00000 |
| 11: 0 | RX FIFO High Watermark Port 1 | The high water mark value.<br>***Note:*** Must be greater than the RX FIFO Low Watermark and RX FIFO transfer threshold. | R/W | 0x0E6 |
| 1. RO = Read Only, No clear on Read; R = Read, Clear on Read; W = Write only; R/W = Read/Write, No clear; R/W/C = Read/Write, Clear on Write | | | | |

**Table 115    RX FIFO High Watermark Port 2 ($0x582)**

| Bit | Name | Description | Type[1] | Default |
|---|---|---|---|---|
| \* | **Register Description:** The default value of 0x0E6 represents 230 eight-byte locations. This equates to 1840 bytes of data. A unit entry in this register equates to 8 bytes of data. When the amount of data stored in the RX FIFO exceeds the high watermark, flow control is automatically initiated within the MAC to avoid an overflow condition. | | | 0x0E6 |
| 31:12 | Reserved | Reserved | RO | 0x00000 |
| 11: 0 | RX FIFO High Watermark Port 2 | The high water mark value.<br>***Note:*** Must be greater than the RX FIFO Low Watermark and RX FIFO transfer threshold. | R/W | 0x0E6 |
| 1. RO = Read Only, No clear on Read; R = Read, Clear on Read; W = Write only; R/W = Read/Write, No clear; R/W/C = Read/Write, Clear on Write | | | | |

**Table 116    RX FIFO High Watermark Port 3 ($0x583)**

| Bit | Name | Description | Type[1] | Default |
|---|---|---|---|---|
| \* | **Register Description:** The default value of 0x0E6 represents 230 eight-byte locations. This equates to 1840 bytes of data. A unit entry in this register equates to 8 bytes of data. When the amount of data stored in the RX FIFO exceeds the high watermark, flow control is automatically initiated within the MAC to avoid an overflow condition. | | | 0x0E6 |
| 31:12 | Reserved | Reserved | RO | 0x00000 |
| 11: 0 | RX FIFO High Watermark Port 3 | The high water mark value.<br>***Note:*** Must be greater than the RX FIFO Low Watermark and RX FIFO transfer threshold. | R/W | 0x0E6 |
| 1. RO = Read Only, No clear on Read; R = Read, Clear on Read; W = Write only; R/W = Read/Write, No clear; R/W/C = Read/Write, Clear on Write | | | | |

**Table 117      RX FIFO Low Watermark Port 0 ($0x58A)**

| Bit | Name | Description | Type[1] | Default |
|-----|------|-------------|---------|---------|
| | **Register Description:** The default value of 0x072 represents 114 eight-byte locations. This equates to 912 bytes of data. A unit entry in this register equates to 8 bytes of data. When the amount of data stored in the RX FIFO falls below the Low Watermark, flow control is automatically de-asserted within the MAC to allow more line-side data to be captured by the RX FIFO. | | 0x072 |
| 31:12 | Reserved | Reserved | RO | 0x00000 |
| 11: 0 | RX FIFO Low Watermark Port 0 | The High Watermark value<br>***Note:*** Should never be greater or equal to the High Watermark. | R/W | 0x072 |
| | 1.  RO = Read Only, No clear on Read; R = Read, Clear on Read; W = Write only; R/W = Read/Write, No clear; R/W/C = Read/Write, Clear on Write | | | |

**Table 118      RX FIFO Low Watermark Port 1 ($0x58B)**

| Bit | Name | Description | Type[1] | Default |
|-----|------|-------------|---------|---------|
| | **Register Description:** The default value of 0x072 represents 114 eight-byte locations. This equates to 912 bytes of data. A unit entry in this register equates to 8 bytes of data. When the amount of data stored in the RX FIFO falls below the Low Watermark, flow control is automatically de-asserted within the MAC to allow more line-side data to be captured by the RX FIFO. | | 0x072 |
| 31:12 | Reserved | Reserved | RO | 0x00000 |
| 11: 0 | RX FIFO Low Watermark Port 1 | The High Watermark value<br>***Note:*** Should never be greater or equal to the High Watermark. | R/W | 0x072 |
| | 1.  RO = Read Only, No clear on Read; R = Read, Clear on Read; W = Write only; R/W = Read/Write, No clear; R/W/C = Read/Write, Clear on Write | | | |

**Table 119      RX FIFO Low Watermark Port 2 ($0x58C)**

| Bit | Name | Description | Type[1] | Default |
|-----|------|-------------|---------|---------|
| | **Register Description:** The default value of 0x072 represents 114 eight-byte locations. This equates to 912 bytes of data. A unit entry in this register equates to 8 bytes of data. When the amount of data stored in the RX FIFO falls below the Low Watermark, flow control is automatically de-asserted within the MAC to allow more line-side data to be captured by the RX FIFO. | | 0x072 |
| 31:12 | Reserved | Reserved | RO | 0x00000 |
| 11: 0 | RX FIFO Low Watermark Port 2 | The High Watermark value<br>***Note:*** Should never be greater or equal to the High Watermark. | R/W | 0x072 |
| | 1.  RO = Read Only, No clear on Read; R = Read, Clear on Read; W = Write only; R/W = Read/Write, No clear; R/W/C = Read/Write, Clear on Write | | | |

**Table 120    RX FIFO Low Watermark Port 3 ($0x58D)**

| Bit | Name | Description | Type[1] | Default |
|-----|------|-------------|---------|---------|
| | **Register Description:** The default value of 0x072 represents 114 eight-byte locations. This equates to 912 bytes of data. A unit entry in this register equates to 8 bytes of data. When the amount of data stored in the RX FIFO falls below the Low watermark, flow control is automatically de-asserted within the MAC to allow more line-side data to be captured by the RX FIFO. | | 0x072 |
| 31:12 | Reserved | Reserved | RO | 0x00000 |
| 11: 0 | RX FIFO Low Watermark Port 3 | The High watermark value<br>*Note:* Should never be greater or equal to the High Watermark. | R/W | 0x072 |
| | 1.  RO = Read Only, No clear on Read; R = Read, Clear on Read; W = Write only; R/W = Read/Write, No clear; R/W/C = Read/Write, Clear on Write | | | |

**Table 121    RX FIFO Overflow Frame Drop Counter Ports 0 - 3 ($0x594 – 0x597)**

| Name | Description | Address | Type[1] | Default |
|------|-------------|---------|---------|---------|
| RX FIFO Overflow Frame Drop Counter on port 0 | When RX FIFO on port 0 becomes full or reset, the number of frames lost/dropped on this port are shown in this register. | 0x594 | R | 0x00000000 |
| RX FIFO Overflow Frame Drop Counter on port 1 | When RX FIFO on port 1 becomes full or reset, the number of frames lost/dropped on this port are shown in this register. | 0x595 | R | 0x00000000 |
| RX FIFO Overflow Frame Drop Counter on port 2 | When RX FIFO on port 2 becomes full or reset, the number of frames lost/dropped on this port are shown in this register. | 0x596 | R | 0x00000000 |
| RX FIFO Overflow Frame Drop Counter on port 3 | When RX FIFO on port 3 becomes full or reset, the number of frames lost/dropped on this port are shown in this register. | 0x597 | R | 0x00000000 |
| | 1.  RO = Read Only, No clear on Read; R = Read, Clear on Read; W = Write only; R/W = Read/Write, No clear; R/W/C = Read/Write, Clear on Write | | | |

**Table 122    RX FIFO Port Reset ($0x59E) (Sheet 1 of 2)**

| Bit | Name | Description | Type[1] | Default |
|-----|------|-------------|---------|---------|
| | **Register Description:** The soft reset register for each port in the RX block. Port ID = bit position in the register. To make the reset active, the bit must be set High. For example, reset of port 1 implies register value = 0000_0018. Setting the bit to 0 de-asserts the reset. | | 0x00000000 |
| 31:4 | Reserved | Reserved | RO | 0x0000000 |
| 3 | Reset RX FIFO for Port 3 | Port 3<br>0 = De-assert reset<br>1 = Reset | R/W | 0 |
| | 1.  RO = Read Only, No clear on Read; R = Read, Clear on Read; W = Write only; R/W = Read/Write, No clear; R/W/C = Read/Write, Clear on Write | | | |

**Table 122    RX FIFO Port Reset ($0x59E) (Sheet 2 of 2)**

| Bit | Name | Description | Type[1] | Default |
|---|---|---|---|---|
| 2 | Reset RX FIFO for Port 2 | Port 2<br>0 = De-assert reset<br>1 = Reset | R/W | 0 |
| 1 | Reset RX FIFO for Port 1 | Port 1<br>0 = De-assert reset<br>1 = Reset | R/W | 0 |
| 0 | Reset RX FIFO for Port 0 | Port 0<br>0 = De-assert reset<br>1 = Reset | R/W | 0 |
| 1. RO = Read Only, No clear on Read; R = Read, Clear on Read; W = Write only; R/W = Read/Write, No clear; R/W/C = Read/Write, Clear on Write ||||| 

**Table 123    RX FIFO Errored Frame Drop Enable ($0x59F)**

| Bit | Name | Description | Type[1] | Default |
|---|---|---|---|---|
| **Register Description:** This register configures the dropping of error packets (DEBAD).<br><br>*Note:*    Jumbo packets are not dropped. |||| 0x00000000 |
| 31:4 | Reserved | Reserved | RO | 0x0000000 |
| 3 | RX FIFO Errored Frame Drop Enable Port 3 | This bit is used in conjunction with MAC filter bits. This allows the user to select whether the errored packets are to be dropped or not.<br>1 = Frame Drop Enable<br>0 = Frame Drop Disable | R/W | 0 |
| 2 | RX FIFO Errored Frame Drop Enable Port 2 | This bit is used in conjunction with MAC filter bits. This allows the user to select whether the errored packets are to be dropped or not.<br>1 = Frame Drop Enable<br>0 = Frame Drop Disable | R/W | 0 |
| 1 | RX FIFO Errored Frame Drop Enable Port 1 | This bit is used in conjunction with MAC filter bits. This allows the user to select whether the errored packets are to be dropped or not.<br>1 = Frame Drop Enable<br>0 = Frame Drop Disable | R/W | 0 |
| 0 | RX FIFO Errored Frame Drop Enable Port 0 | This bit is used in conjunction with MAC filter bits. This allows the user to select whether the errored packets are to be dropped or not.<br>1 = Frame Drop Enable<br>0 = Frame Drop Disable | R/W | 0 |
| 1. RO = Read Only, No clear on Read; R = Read, Clear on Read; W = Write only; R/W = Read/Write, No clear; R/W/C = Read/Write, Clear on Write |||||

**Table 124    RX FIFO Overflow Event ($0x5A0)**

| Bit | Name | Description | Type[1] | Default |
|---|---|---|---|---|
| **Register Description:** This register provides a status if a FIFO-full situation occurs (for example, a FIFO overflow). The bit position equals the port number. This register is cleared on Read. ||| 0x00000000 |
| 31:4 | Reserved | Reserved | RO | 0x0000000 |
| 3 | RX FIFO Overflow Event on Port 3 | Port 3<br>0 = FIFO overflow event did not occur<br>1 = FIFO overflow event occurred | R | 0 |
| 2 | RX FIFO Overflow Event on Port 2 | Port 2<br>0 = FIFO overflow event did not occur<br>1 = FIFO overflow event occurred | R | 0 |
| 1 | RX FIFO Overflow Event on Port 1 | Port 1<br>0 = FIFO overflow event did not occur<br>1 = FIFO overflow event occurred | R | 0 |
| 0 | RX FIFO Overflow Event on Port 0 | Port 0<br>0 = FIFO overflow event did not occur<br>1 = FIFO overflow event occurred | R | 0 |
| 1. RO = Read Only, No clear on Read; R = Read, Clear on Read; W = Write only; R/W = Read/Write, No clear; R/W/C = Read/Write, Clear on Write |||||

**Table 125    RX FIFO Errored Frame Drop Counter Ports 0 - 3 ($0x5A2 - 0x5A5)**

| Name | Description | Address | Type | Default |
|------|-------------|---------|------|---------|
| RX FIFO Errored Frame Drop Counter on Port 0 | This register counts all frames dropped from the RX FIFO for port 0 by meeting one of the following conditions:<br>• Frames are removed in conjunction with the Section 123, *RX FIFO Errored Frame Drop Enable ($0x59F)* and the Section 91, *RX Packet Filter Control ($ Port_Index + 0x19)*.<br>• Frames are greater than the Section 81, *Max Frame Size (Addr: Port_Index + 0x0F)*.<br>This register is cleared on Read. | 0x5A2 | R | 0x00000000 |
| RX FIFO Errored Frame Drop Counter on Port 1 | This register counts all frames dropped from the RX FIFO for port 1 by meeting one of the following conditions:<br>• Frames are removed in conjunction with the Section 123, *RX FIFO Errored Frame Drop Enable ($0x59F)* and the Section 91, *RX Packet Filter Control ($ Port_Index + 0x19)*.<br>• Frames are greater than the Section 81, *Max Frame Size (Addr: Port_Index + 0x0F)*.<br>This register is cleared on Read. | 0x5A3 | R | 0x00000000 |
| RX FIFO Errored Frame Drop Counter on Port 2 | This register counts all frames dropped from the RX FIFO for port 2 by meeting one of the following conditions:<br>• Frames are removed in conjunction with the Section 123, *RX FIFO Errored Frame Drop Enable ($0x59F)* and the Section 91, *RX Packet Filter Control ($ Port_Index + 0x19)*.<br>• Frames are greater than the Section 81, *Max Frame Size (Addr: Port_Index + 0x0F)*.<br>This register is cleared on Read. | 0x5A4 | R | 0x00000000 |
| RX FIFO Errored Frame Drop Counter on Port 3 | This register counts all frames dropped from the RX FIFO for port 3 by meeting one of the following conditions:<br>• Frames are removed in conjunction with the Section 123, *RX FIFO Errored Frame Drop Enable ($0x59F)* and the Section 91, *RX Packet Filter Control ($ Port_Index + 0x19)*.<br>• Frames are greater than the Section 81, *Max Frame Size (Addr: Port_Index + 0x0F)*.<br>This register is cleared on Read. | 0x5A5 | R | 0x00000000 |

1. RO = Read Only, No clear on Read; R = Read, Clear on Read; W = Write only; R/W = Read/Write, No clear; R/W/C = Read/Write, Clear on Write

**Table 126     RX FIFO SPI3 Loopback Enable for Ports 0 - 3 ($0x5B2)**

| Bit | Name | Description | Type[1] | Default |
|-----|------|-------------|---------|---------|
| **Register Description:** Enables the TX SPI3 port to send packets into the RX_FIFO instead of into the TX FIFO, creating a SPI3 loopback. | | | | 0x00000000 |
| 31:12 | Reserved | Reserved | RO | 0x00000 |
| 11 | SPI3 loopback enable for Port 3 | 0 = Disabled<br>1 = Enabled | R/W | 0x0 |
| 10 | SPI3 loopback enable for Port 2 | 0 = Disabled<br>1 = Enabled | R/W | 0x0 |
| 9 | SPI3 loopback enable for Port 1 | 0 = Disabled<br>1 = Enabled | R/W | 0x0 |
| 8 | SPI3 loopback enable for Port 0 | 0 = Disabled<br>1 = Enabled | R/W | 0x0 |
| 7:0 | Reserved | Write as 0, ignore on Read. | R/W | 0x00 |
| 1. RO = Read Only, No clear on Read; R = Read, Clear on Read; W = Write only; R/W = Read/Write, No clear; R/W/C = Read/Write, Clear on Write | | | | |

**Table 127     RX FIFO Padding and CRC Strip Enable ($0x5B3) (Sheet 1 of 2)**

| Bit | Name | Description | Type[1] | Default |
|-----|------|-------------|---------|---------|
| **Register Description:** This control register enables to pre-pend every packet with two extra bytes and also enables the CRC stripping of a packet. | | | | 0x00000000 |
| 31:8 | Reserved | Reserved | RO | 0x000000 |
| 7 | CRC Stripping Enable for Port 3 | CRC stripping is enabled for Port 3.<br>0 = Disabled<br>1 = Enabled | R/W | 0 |
| 6 | CRC Stripping Enable for Port 2 | CRC stripping is enabled for Port 2.<br>0 = Disabled<br>1 = Enabled | R/W | 0 |
| 5 | CRC Stripping Enable for Port 1 | CRC stripping is enabled for Port 1.<br>0 = Disabled<br>1 = Enabled | R/W | 0 |
| 4 | CRC Stripping Enable for Port 0 | CRC stripping is enabled for Port 0.<br>0 = Pre-pending Disabled<br>1 = Pre-pending Enabled | R/W | 0 |
| 3 | Pre-pending Enable[2] Port 3 | Enables pre-pending of two bytes at the start of every packet – Port 3.<br>0 = Disabled<br>1 = Enabled | R/W | 0 |
| 1. RO = Read Only, No clear on Read; R = Read, Clear on Read; W = Write only; R/W = Read/Write, No clear; R/W/C = Read/Write, Clear on Write<br>2. Pre-pending should not be enabled in loopback mode. | | | | |

**Table 127    RX FIFO Padding and CRC Strip Enable ($0x5B3) (Sheet 2 of 2)**

| Bit | Name | Description | Type[1] | Default |
|-----|------|-------------|---------|---------|
| 2 | Pre-pending Enable[2] Port 2 | Enables pre-pending of two bytes at the start of every packet – Port 2.<br>0 = Disabled<br>1 = Enabled | R/W | 0 |
| 1 | Pre-pending Enable[2] Port 1 | Enables pre-pending of two bytes at the start of every packet – Port 1.<br>0 = Disabled<br>1 = Enabled | R/W | 0 |
| 0 | Pre-pending Enable[2] Port 0 | Enables pre-pending of two bytes at the start of every packet – Port 0.<br>0 = Disabled<br>1 = Enabled | R/W | 0 |

1. RO = Read Only, No clear on Read; R = Read, Clear on Read; W = Write only; R/W = Read/Write, No clear; R/W/C = Read/Write, Clear on Write
2. Pre-pending should not be enabled in loopback mode.

**Table 128    RX FIFO Transfer Threshold Port 0 ($0x5B8)**

| Bit | Name | Description | Type | Default |
|-----|------|-------------|------|---------|
| **Register Description:** RX FIFO transfer threshold for port 0 in 8-byte location. | | | | 0x000000BE |
| 31:12 | Reserved | Reserved | RO | 0x00000 |
| 11:0 | RX FIFO Transfer Threshold - Port 0 | RX FIFO transfer threshold for port 0. This must be less than the RX FIFO High water mark.<br>User definable control register that sets the threshold where a packet starts transitioning to the SPI3 interface from the RX FIFO before the EOP is received. Packets received in the RX FIFO below this threshold are treated as store and forward.<br>*Note:* Do not program the RX FIFO transfer threshold below a setting of 0xBE (1520bytes). | R/W | 0x0BE |

1. RO = Read Only, No clear on Read; R = Read, Clear on Read; W = Write only; R/W = Read/Write, No clear; R/W/C = Read/Write, Clear on Write

**Table 129    RX FIFO Transfer Threshold Port 1 ($0x5B9) (Sheet 1 of 2)**

| Bit | Name | Description | Type | Default |
|-----|------|-------------|------|---------|
| **Register Description:** RX FIFO transfer threshold for port 1in 8-byte location. | | | | 0x000000BE |

**Table 129     RX FIFO Transfer Threshold Port 1 ($0x5B9) (Sheet 2 of 2)**

| | | | | |
|---|---|---|---|---|
| 31:12 | Reserved | Reserved | RO | 0x00000 |
| 11:0 | RX FIFO Transfer Threshold - Port 1 | RX FIFO transfer threshold for port 1. This must be less than the RX FIFO High watermark.<br><br>User definable control register that sets the threshold where a packet starts transitioning to the SPI3 interface from the RX FIFO before the EOP is received. Packets received in the RX FIFO below this threshold are treated as store and forward.<br>*Note:*  Do not program the RX FIFO transfer threshold below a setting of 0xBE (1520bytes). | R/W | 0x0BE |
| 1.  RO = Read Only, No clear on Read; R = Read, Clear on Read; W = Write only; R/W = Read/Write, No clear; R/W/C = Read/Write, Clear on Write | | | | |

**Table 130     RX FIFO Transfer Threshold Port 2 ($0x5BA)**

| Bit | Name | Description | Type | Default |
|---|---|---|---|---|
| **Register Description:** RX FIFO transfer threshold for port 2 in 8-byte location. | | | | 0x000000BE |
| 31:12 | Reserved | Reserved | RO | 0x00000 |
| 11:0 | RX FIFO Transfer Threshold - Port 2 | RX FIFO transfer threshold for port 2. This must be less than the RX FIFO High water mark.<br><br>User definable control register that sets the threshold where a packet starts transitioning to the SPI3 interface from the RX FIFO before the EOP is received. Packets received in the RX FIFO below this threshold are treated as store and forward.<br>*Note:*  Do not program the RX FIFO transfer threshold below a setting of 0xBE (1520bytes). | R/W | 0x0BE |
| 1.  RO = Read Only, No clear on Read; R = Read, Clear on Read; W = Write only; R/W = Read/Write, No clear; R/W/C = Read/Write, Clear on Write | | | | |

**Table 131     RX FIFO Transfer Threshold Port 3 ($0x5BB)**

| Bit | Name | Description | Type | Default |
|---|---|---|---|---|
| **Register Description:** RX FIFO transfer threshold for port 3 in 8-byte location. | | | | 0x000000BE |
| 31:12 | Reserved | Reserved | RO | 0x00000 |
| 11:0 | RX FIFO Transfer Threshold - Port 3 | RX FIFO transfer threshold for port 3. This must be less than the RX FIFO High water mark.<br><br>User definable control register that sets the threshold where a packet starts transitioning to the SPI3 interface from the RX FIFO before the EOP is received. Packets received in the RX FIFO below this threshold are treated as store and forward.<br>*Note:*  Do not program the RX FIFO transfer threshold below a setting of 0xBE (1520bytes). | R/W | 0x0BE |
| 1.  RO = Read Only, No clear on Read; R = Read, Clear on Read; W = Write only; R/W = Read/Write, No clear; R/W/C = Read/Write, Clear on Write | | | | |

## 8.4.7　TX FIFO Register Overview

Table 132 through Table 139 provide an overview of the TX FIFO registers, which include the TX FIFO High and Low watermark.

### Table 132　TX FIFO High Watermark Ports 0 - 3 ($0x600 – 0x603)

| Name | Description | Address | Type[1] | Default |
|------|-------------|---------|---------|---------|
| TX FIFO High Watermark Port 0 | **High watermark for TX FIFO Port 0.** The default value of 0x3E0 represents 992 8-byte locations. This equates to 7936 bytes of data. A unit entry in this register equates to 8 bytes of data. When the amount of data stored in the TX FIFO exceeds the high watermark, flow control is automatically initiated on the SPI3 interface to request that the switch fabric stops data transfers to avoid an overflow condition. | 0x600 | R/W | 0x000003E0 |
| TX FIFO High Watermark Port 1 | **High watermark for TX FIFO Port 1.** The default value of 0x3E0 represents 992 8-byte locations. This equates to 7936 bytes of data. A unit entry in this register equates to 8 bytes of data. When the amount of data stored in the TX FIFO exceeds the high watermark, flow control is automatically initiated on the SPI3 interface to request that the switch fabric stops data transfers to avoid an overflow condition. | 0x601 | R/W | 0x000003E0 |
| TX FIFO High Watermark Port 2 | **High watermark for TX FIFO Port 2.** The default value of 0x3E0 represents 992 8-byte locations. This equates to 7936 bytes of data. A unit entry in this register equates to 8 bytes of data. When the amount of data stored in the TX FIFO exceeds the high watermark, flow control is automatically initiated on the SPI3 interface to request that the switch fabric stops data transfers to avoid an overflow condition. | 0x602 | R/W | 0x000003E0 |
| TX FIFO High Watermark Port 3 | **High watermark for TX FIFO Port 3.** The default value of 0x3E0 represents 992 8-byte locations. This equates to 7936 bytes of data. A unit entry in this register equates to 8 bytes of data. When the amount of data stored in the TX FIFO exceeds the high watermark, flow control is automatically initiated on the SPI3 interface to request that the switch fabric stops data transfers to avoid an overflow condition. | 0x603 | R/W | 0x000003E0 |
| 1. RO = Read Only, No clear on Read; R = Read, Clear on Read; W = Write only; R/W = Read/Write, No clear; R/W/C = Read/Write, Clear on Write | | | | |

**Table 133     TX FIFO Low Watermark Register Ports 0 - 3 ($0x60A – 0x60D)**

| Name | Description | Address | Type[1] | Default |
|---|---|---|---|---|
| TX FIFO Low Watermark Port 0 | **Low watermark for TX FIFO Port 0.** The default value of 0x0D0 represents 208 8-byte locations. This equates to 1664 bytes of data. A unit entry in this register equates to 8 bytes of data. When the amount of data stored in the TX FIFO falls below the low watermark, flow control is automatically de-asserted on the SPI3 interface to allow further data to be sent by the switch fabric to the IXF1104 MAC. | 0x60A | R/W | 0x000000D0 |
| TX FIFO Low Watermark Port 1 | **Low watermark for TX FIFO Port 1.** The default value of 0x0D0 represents 208 8-byte locations. This equates to 1664 bytes of data. A unit entry in this register equates to 8 bytes of data. When the amount of data stored in the TX FIFO falls below the low watermark, flow control is automatically de-asserted on the SPI3 interface to allow further data to be sent by the switch fabric to the IXF1104 MAC. | 0x60B | R/W | 0x000000D0 |
| TX FIFO Low Watermark Port 2 | **Low watermark for TX FIFO Port 2.** The default value of 0x0D0 represents 208 8-byte locations. This equates to 1664 bytes of data. A unit entry in this register equates to 8 bytes of data. When the amount of data stored in the TX FIFO falls below the low watermark, flow control is automatically de-asserted on the SPI3 interface to allow further data to be sent by the switch fabric to the IXF1104 MAC. | 0x60C | R/W | 0x000000D0 |
| TX FIFO Low Watermark Port 3 | **Low watermark for TX FIFO Port 3.** The default value of 0x0D0 represents 208 8-byte locations. This equates to 1664 bytes of data. A unit entry in this register equates to 8 bytes of data. When the amount of data stored in the TX FIFO falls below the low watermark, flow control is automatically de-asserted on the SPI3 interface to allow further data to be sent by the switch fabric to the IXF1104 MAC. | 0x60D | R/W | 0x000000D0 |
| 1. RO = Read Only, No clear on Read; R = Read, Clear on Read; W = Write only; R/W = Read/Write, No clear; R/W/C = Read/Write, Clear on Write | | | | |

**Table 134    TX FIFO MAC Threshold Register Ports 0 - 3 ($0x614 – 0x617)**

| Name | Description | Address | Type[1] | Default |
|------|-------------|---------|---------|---------|
| TX FIFO MAC Threshold Port 0 | **MAC threshold for TX FIFO Port 0.** The default value of 0x1BE represents 446 8-byte locations. This equates to 3568 bytes of data. A unit entry in this register equates to 8 bytes of data. When the amount of data stored in the TX FIFO reaches this threshold, data is forwarded to the MAC core and line-side interfaces for onward transmission. By setting the threshold to an appropriate value, the user can configure the TX FIFO to operate in a "cut-through" mode rather than the default "store and forward" operation mode. | 0x614 | R/W | 0x000001BE |
| TX FIFO MAC Threshold Port 1 | **MAC threshold for TX FIFO Port 1.** The default value of 0x1BE represents 446 8-byte locations. This equates to 3568 bytes of data. A unit entry in this register equates to 8 bytes of data. When the amount of data stored in the TX FIFO reaches this threshold, data is forwarded to the MAC core and line-side interfaces for onward transmission. By setting the threshold to an appropriate value, the user can configure the TX FIFO to operate in a "cut-through" mode rather than the default "store and forward" operation mode. | 0x615 | R/W | 0x000001BE |
| TX FIFO MAC Threshold Port 2 | **MAC threshold for TX FIFO Port 2.** The default value of 0x1BE represents 446 8-byte locations. This equates to 3568 bytes of data. A unit entry in this register equates to 8 bytes of data. When the amount of data stored in the TX FIFO reaches this threshold, data is forwarded to the MAC core and line-side interfaces for onward transmission. By setting the threshold to an appropriate value, the user can configure the TX FIFO to operate in a "cut-through" mode rather than the default "store and forward" operation mode. | 0x616 | R/W | 0x000001BE |
| TX FIFO MAC Threshold Port 3 | **MAC threshold for TX FIFO Port 3.** The default value of 0x1BE represents 446 8-byte locations. This equates to 3568 bytes of data. A unit entry in this register equates to 8 bytes of data. When the amount of data stored in the TX FIFO reaches this threshold, data is forwarded to the MAC core and line-side interfaces for onward transmission. By setting the threshold to an appropriate value, the user can configure the TX FIFO to operate in a "cut-through" mode rather than the default "store and forward" operation mode. | 0x617 | R/W | 0x000001BE |

1. RO = Read Only, No clear on Read; R = Read, Clear on Read; W = Write only; R/W = Read/Write, No clear; R/W/C = Read/Write, Clear on Write

**Table 135    TX FIFO Overflow/Underflow/Out of Sequence Event ($0x61E) (Sheet 1 of 2)**

| Bit | Name | Description | Type[1] | Default |
|-----|------|-------------|---------|---------|
| **Register Description: TX FIFO Out of Sequence Event:** These register bits provide status information, and indicate if out-of-sequence data has been received. The bit position equals the port number + 8. These bits are cleared on Read. | | | | 0x0 |
| **Register Description: TX FIFO Underflow Event:** This register provides a status that a FIFO Empty situation has occurred (for example, a FIFO under-run). The bit position equals the port number + 4. This register is cleared on Read. | | | | 0x0 |
| **Register Description: TX FIFO Overflow Event:** This register provides a status that a FIFO full situation has occurred (for example, a FIFO overflow). The bit position equals the port number. This register is cleared on Read. | | | | 0x0 |
| 31:12 | Reserved | Reserved | RO | 0x00000 |
| 11 | FOSE3 | Port 3<br>0 = FIFO out of sequence event did not occur<br>1 = FIFO out of sequence event occurred | R | 0 |
| 10 | FOSE2 | Port 2<br>0 = FIFO out of sequence event did not occur<br>1 = FIFO out of sequence event occurred | R | 0 |
| 9 | FOSE1 | Port 1<br>0 = FIFO out of sequence event did not occur<br>1 = FIFO out of sequence event occurred | R | 0 |
| 8 | FOSE0 | Port 0<br>0 = FIFO out of sequence event did not occur<br>1 = FIFO out of sequence event occurred | R | 0 |
| 7 | FUE3 | Port 3<br>0 = FIFO underflow event did not occur<br>1 = FIFO underflow event occurred | R | 0 |
| 6 | FUE2 | Port 2<br>0 = FIFO underflow event did not occur<br>1 = FIFO underflow event occurred | R | 0 |
| 5 | FUE1 | Port 1<br>0 = FIFO underflow event did not occur<br>1 = FIFO underflow event occurred | R | 0 |
| 4 | FUE0 | Port 0<br>0 = FIFO underflow event did not occur<br>1 = FIFO underflow event occurred | R | 0 |
| 3 | FOE3 | Port 3<br>0 = FIFO overflow event did not occur<br>1 = FIFO overflow event occurred | R | 0 |
| 1.  RO = Read Only, No clear on Read; R = Read, Clear on Read; W = Write only; R/W = Read/Write, No clear; R/W/C = Read/Write, Clear on Write | | | | |

**Table 135    TX FIFO Overflow/Underflow/Out of Sequence Event ($0x61E) (Sheet 2 of 2)**

| Bit | Name | Description | Type[1] | Default |
|-----|------|-------------|---------|---------|
| 2 | FOE2 | Port 2<br>0 = FIFO overflow event did not occur<br>1 = FIFO overflow event occurred | R | 0 |
| 1 | FOE1 | Port 1<br>0 = FIFO overflow event did not occur<br>1 = FIFO overflow event occurred | R | 0 |
| 0 | FOE0 | Port 0<br>0 = FIFO overflow event did not occur<br>1 = FIFO overflow event occurred | R | 0 |
| 1. RO = Read Only, No clear on Read; R = Read, Clear on Read; W = Write only; R/W = Read/Write, No clear; R/W/C = Read/Write, Clear on Write | | | | |

**Table 136    Loop RX Data to TX FIFO (Line-Side Loopback) Ports 0 - 3 ($0x61F)**

| Bit | Name | Description | Type[1] | Default |
|-----|------|-------------|---------|---------|
| **Register Description:** This register enables data received from the line-side receive interface through the MAC to be sent to the TX FIFO and back to the line-side transmit interface. | | | | 0x00000000 |
| 31:4 | Reserved | Reserved | RO | 0x0000000 |
| 3 | Port 3 Line-Side Loopback | 0 = Disable line-side loopback<br>1 = Enable line-side loopback | R/W | 0 |
| 2 | Port 2 Line-Side Loopback | 0 = Disable line-side loopback<br>1 = Enable line-side loopback | R/W | 0 |
| 1 | Port 1 Line-Side Loopback | 0 = Disable line-side loopback<br>1 = Enable line-side loopback | R/W | 0 |
| 0 | Port 0 Line-Side Loopback | 0 = Disable line-side loopback<br>1 = Enable line-side loopback | R/W | 0 |
| 1. RO = Read Only, No clear on Read; R = Read, Clear on Read; W = Write only; R/W = Read/Write, No clear; R/W/C = Read/Write, Clear on Write | | | | |

**Table 137    TX FIFO Port Reset ($0x620) (Sheet 1 of 2)**

| Bit | Name | Description | Type[1] | Default |
|-----|------|-------------|---------|---------|
| **Register Description:** This is a port reset register for each port in the TX block. Port ID = bit position in the register. To make the port active, the bit must be set to Low. (For example, reset of Port 3 implies register value = 1000, setting the bit to 1 asserts the port reset). | | | | 0x00000000 |
| 31:4 | Reserved | Reserved | RO | 0x0000000 |
| 3 | Port 3 Reset | Port 3<br>0 = De-assert Reset<br>1 = Assert Reset | R/W | 0 |
| 1. RO = Read Only, No clear on Read; R = Read, Clear on Read; W = Write only; R/W = Read/Write, No clear; R/W/C = Read/Write, Clear on Write | | | | |

**Table 137**   **TX FIFO Port Reset ($0x620) (Sheet 2 of 2)**

| Bit | Name | Description | Type[1] | Default |
|---|---|---|---|---|
| 2 | Port 2 Reset | Port 2<br>0 = De-assert Reset<br>1 = Assert Reset | R/W | 0 |
| 1 | Port 1 Reset | Port 1<br>0 = De-assert Reset<br>1 = Assert Reset | R/W | 0 |
| 0 | Port 0 Reset | Port 0<br>0 = De-assert Reset<br>1 = Assert Reset | R/W | 0 |
| 1. RO = Read Only, No clear on Read; R = Read, Clear on Read; W = Write only; R/W = Read/Write, No clear; R/W/C = Read/Write, Clear on Write ||||||

**Table 138**   **TX FIFO Overflow Frame Drop Counter Ports 0 - 3 ($0x621 – 0x624)**

| Name | Description | Address | Type[*] | Default |
|---|---|---|---|---|
| TX FIFO overflow frame drop counter on Port 0 | When TX FIFO on Port 0 becomes full or reset, the number of frames lost or removed on this port is shown in this register. This register is cleared on Read. | 0x621 | R | 0x00000000 |
| TX FIFO overflow frame drop counter on Port 1 | When TX FIFO on Port 1 becomes full or reset, the number of frames lost or removed on this port is shown in this register. This register is cleared on Read. | 0x622 | R | 0x00000000 |
| TX FIFO overflow frame drop counter on Port 2 | When TX FIFO on Port 2 becomes full or reset, the number of frames lost or removed on this port is shown in this register. This register is cleared on Read. | 0x623 | R | 0x00000000 |
| TX FIFO overflow frame drop counter on Port 3 | When TX FIFO on Port 3 becomes full or reset, the number of frames lost or removed on this port is shown in this register. This register is cleared on Read. | 0x624 | R | 0x00000000 |
| 1. RO = Read Only, No clear on Read; R = Read, Clear on Read; W = Write only; R/W = Read/Write, No clear; R/W/C = Read/Write, Clear on Write |||||

**Table 139    TX FIFO Errored Frame Drop Counter Ports 0 - 3 ($0x625 – 0x629)**

| Name | Description | Address | Type[*] | Default |
|---|---|---|---|---|
| TX FIFO errored frame drop counter on Port 0 | This register provides the number of packets dropped by the TX FIFO due to the following: Data Parity Errors Short SOPs (two consecutive SOPs for a port with no EOP) Small Packets (9-14 bytes) Frames received that are signaled with TERR on the SPI3 TX interface. *Note:*    This register is cleared on Read. | 0x625 | R | 0x00000000 |
| TX FIFO errored frame drop counter on Port 1 | This register provides the number of packets dropped by the TX FIFO due to the following: Data Parity Errors Short SOPs (two consecutive SOPs for a port with no EOP) Small Packets (9-14 bytes) Frames received that are signaled with TERR on the SPI3 TX interface. *Note:*    This register is cleared on Read. | 0x626 | R | 0x00000000 |
| TX FIFO errored frame drop counter on Port 2 | This register provides the number of packets dropped by the TX FIFO due to the following: Data Parity Errors Short SOPs (two consecutive SOPs for a port with no EOP) Small Packets (9-14 bytes) Frames received that are signaled with TERR on the SPI3 TX interface. *Note:*    This register is cleared on Read. | 0x627 | R | 0x00000000 |
| TX FIFO errored frame drop counter on Port 3 | This register provides the number of packets dropped by the TX FIFO due to the following: Data Parity Errors Short SOPs (two consecutive SOPs for a port with no EOP) Small Packets (9-14 bytes) Frames received that are signaled with TERR on the SPI3 TX interface. *Note:*    This register is cleared on Read. | 0x628 | R | 0x00000000 |
| 1.   RO = Read Only, No clear on Read; R = Read, Clear on Read; W = Write only; R/W = Read/Write, No clear; R/W/C = Read/Write, Clear on Write | | | | |

**Table 140      TX FIFO Occupancy Counter for Ports 0 - 3 ($0x62D – 0x630)**

| Name | Description | Address | Type | Default |
|------|-------------|---------|------|---------|
| Occupancy for Tx FIFO Port 0 | This register gives the Occupancy for TX FIFO Port 0. This is a Read only register | 0x62D | R | 0x00000000 |
| Occupancy for Tx FIFO Port 1 | This register gives the Occupancy for TX FIFO Port 1. This is a Read only register | 0x62E | R | 0x00000000 |
| Occupancy for Tx FIFO Port 2 | This register gives the Occupancy for TX FIFO Port 2. This is a Read only register | 0x62F | R | 0x00000000 |
| Occupancy for Tx FIFO Port 3 | This register gives the Occupancy for TX FIFO Port 3. This is a Read only register | 0x630 | R | 0x00000000 |
| 1. RO = Read Only, No clear on Read; R = Read, Clear on Read; W = Write only; R/W = Read/Write, No clear; R/W/C = Read/Write, Clear on Write | | | | |

**Table 141      TX FIFO Port Drop Enable ($0x63D)**

| Bit | Name | Description | Type | Default |
|-----|------|-------------|------|---------|
| **Register Description:** Independently enables the individual TX FIFOs to drop erroneous packets. | | | | 0x0000000f |
| 31:4 | Reserved | Reserved | RO | 0x000000 |
| 3 | Port 3 Drop | 0 = Disable the TXFIFO from dropping erroneous packets<br>1 = Enable the TXFIFO to drop erroneous packets | R/W | 1 |
| 2 | Port 2 Drop | 0 = Disable the TXFIFO from dropping erroneous packets<br>1 = Enable the TXFIFO to drop erroneous packets | R/W | 1 |
| 1 | Port 1 Drop | 0 = Disable the TXFIFO from dropping erroneous packets<br>1 = Enable the TXFIFO to drop erroneous packets | R/W | 1 |
| 0 | Port 0 Drop | 0 = Disable the TXFIFO from dropping erroneous packets<br>1 = Enable the TXFIFO to drop erroneous packets | R/W | 1 |
| 1. RO = Read Only, No clear on Read; R = Read, Clear on Read; W = Write only; R/W = Read/Write, No clear; R/W/C = Read/Write, Clear on Write | | | | |

## 8.4.8      MDIO Register Overview

Table 142 through Table 145 provide an overview of the MDIO registers.

**Table 142      MDIO Single Command ($0x680) (Sheet 1 of 2)**

| Bit | Name | Description | Type[1] | Default |
|-----|------|-------------|---------|---------|
| **Register Description:** Gives the CPU the ability to perform single MDIO read and write accesses to the external PHY for ports that are configured in copper mode. | | | | 0x00010000 |
| 31:21 | Reserved | Reserved | RO | 00000000000 |
| 20 | MDIO Command | Performs the MDIO operation. Cleared when done.<br>0 = MDIO ready, operation complete<br>1 = Perform operation | R/W | 0 |
| 19:18 | Reserved | Reserved | RO | 00 |
| 1. RO = Read Only, No clear on Read; R = Read, Clear on Read; W = Write only; R/W = Read/Write, No clear; R/W/C = Read/Write, Clear on Write | | | | |

**Table 142      MDIO Single Command ($0x680) (Sheet 2 of 2)**

| Bit | Name | Description | Type[1] | Default |
|-----|------|-------------|---------|---------|
| 17:16 | OP Code | MDIO Op Code; two bits identify operation to be performed:<br>00 = Reserved<br>01 = Write operation (as defined in IEEE 802.3, clause 22.2.4.5)<br>10 = Read operation (as defined in IEEE 802.3, clause 22.2.4.5)<br>11 = Reserved | R/W | 01 |
| 15:10 | Reserved | Reserved | RO | 000000 |
| 9:8 | PHY Address | Sets bits 1:0 of the external PHY address. Bits 4:2 of the PHY address are fixed at 000. | R/W | 00 |
| 7:5 | Reserved | Reserved | RO | 000 |
| 4:0 | REG Address | Five-bit address to one among 32 registers in an addressed PHY device. | R/W | 00000 |
| 1. RO = Read Only, No clear on Read; R = Read, Clear on Read; W = Write only; R/W = Read/Write, No clear; R/W/C = Read/Write, Clear on Write | | | | |

**Table 143      MDIO Single Read and Write Data ($0x681)**

| Bit | Name | Description | Type[1] | Default |
|-----|------|-------------|---------|---------|
| **Register Description:** MDIO read and write data. | | | | 0x00000000 |
| 31:16 | MDIO Read Data | MDIO Read data from external device. | RO | 0x0000 |
| 15:0 | MDIO Write Data | MDIO Write data to external device. | R/W | 0x0000 |
| 1. RO = Read Only, No clear on Read; R = Read, Clear on Read; W = Write only; R/W = Read/Write, No clear; R/W/C = Read/Write, Clear on Write | | | | |

**Table 144      Autoscan PHY Address Enable ($0x682)**

| Bit | Name | Description | Type[1] | Default |
|-----|------|-------------|---------|---------|
| **Register Description:** Defines valid PHY addresses. Each bit enables the corresponding PHY address.<br>0 = Disable the PHY address<br>1 = Enable the PHY address<br>*Note:*    Autoscan is only applicable for the ports in copper mode. | | | | 0x00000000 |
| 31:4 | Reserved | Reserved | RO | 0x0000000 |
| 3:0 | Autoscan PHY Address | Autoscan PHY address enable<br>0 = Disable address<br>1 = Enable address | R/W | 1111 |
| 1. RO = Read Only, No clear on Read; R = Read, Clear on Read; W = Write only; R/W = Read/Write, No clear; R/W/C = Read/Write, Clear on Write | | | | |

**Table 145    MDIO Control ($0x683)**

| Bit | Name | Description | Type[1] | Default |
|---|---|---|---|---|
| **Register Description:** Miscellaneous control bits. | | | | 0x00000000 |
| 31:4 | Reserved | Reserved | RO | 0x000 |
| 3 | MDIO in Progress | MDIO progress. This bit reflects the status of MDIO transaction<br>0 = MDIO Single command not in progress<br>1 = MDIO Single Command in progress | RO | 0 |
| 2 | MDIO in Progress Enable | Enables the MDIO in progress bit<br>0 = Disable MDIO in progress register bit<br>1 = Enable MDIO in progress register bit | R/W | 0 |
| 1 | Autoscan Enable | Autoscan enable<br>0 = Disable Autoscan<br>1 = Enable Autoscan | R/W | 0 |
| 0 | MDC Speed | MDC speed<br>0 = MDC runs at 2.5 MHz<br>1 = MDC runs at 18 MHz | R/W | 0 |
| 1. RO = Read Only, No clear on Read; R = Read, Clear on Read; W = Write only; R/W = Read/Write, No clear; R/W/C = Read/Write, Clear on Write | | | | |

## 8.4.9    SPI3 Register Overview

Table 146 through Table 148, *Address Parity Error Packet Drop Counter ($0x70A)*, on page 210 provide an overview of the SPI3 registers.

**Table 146    SPI3 Transmit and Global Configuration ($0x700) (Sheet 1 of 3)**

| Bit | Name | Description | Type[1] | Default |
|---|---|---|---|---|
| **Register Description:** This register gives the configuration related to the SPI3 Transmitter and Global configuration (4 x 8 mode). | | | | 0x00200000 |
| 31:24 | Reserved | Reserved | RO | 0x00 |
| 23 | SPI3 Transmitter Soft Reset | 1 = The SPI3 TX block is reset. | R/W | 0 |
| 22 | SPI3 Receiver Soft Reset | 1 = The SPI3 RX block is reset. | R/W | 0 |
| 21 | SPHY/MPHY Mode | 0 = Indicates that SPI3 block operates in 32-bit MPHY mode.<br>1 = Indicates that the SPI3 block operates in 4 x 8 SPHY mode.<br>This configuration affects both the SPI3 transmitter and receiver functionality. | R/W | 1 |
| 1. RO = Read Only, No clear on Read; R = Read, Clear on Read; W = Write only; R/W = Read/Write, No clear; R/W/C = Read/Write, Clear on Write | | | | |

**Table 146    SPI3 Transmit and Global Configuration ($0x700) (Sheet 2 of 3)**

| Bit | Name | Description | Type[1] | Default |
|---|---|---|---|---|
| 20 | Tx_ad_prtyer_drop | Indicates whether to drop packets received with parity error during the address selection phase (Tsx and nTenb High) should be dropped.<br>0 = Do not drop packets with address parity error<br>1 = Drop packets with address parity error<br>This is applicable only in MPHY mode of operation. This bit is ignored in SPHY (4 x 8) mode as there will be no address selection. | R/W | 0 |
| 19 | Dat_prtyer_drp Port 3 | **SPHY/MPHY Mode:**<br>Indicates whether to drop packets with data parity error for port 3.<br>0 = Do not drop packets with data parity error (default)<br>1 = Drop packets with data parity error | R/W | 0x0 |
| 18 | Dat_prtyer_drp Port 2 | **SPHY/MPHY Mode:**<br>Indicates whether to drop packets with data parity error for port 2.<br>0 = Do not drop packets with data parity error (default)<br>1 = Drop packets with data parity error | R/W | 0 |
| 17 | Dat_prtyer_drp Port 1 | **SPHY/MPHY Mode:**<br>Indicates whether to drop packets with data parity error for port 1.<br>0 = Do not drop packets with data parity error (default)<br>1 = Drop packets with data parity error | R/W | 0 |
| 16 | Dat_prtyer_drp Port 0 | **SPHY/MPHY Mode:**<br>Indicates whether to drop packets with data parity error for port 0.<br>0 = Do not drop packets with data parity error (default)<br>1 = Drop packets with data parity error | R/W | 0 |
| 15:8 | Reserved | Write as 0, ignore on Read. | R/W | 00000000 |
| 7 | Tx_parity_sense Port 3 | **SPHY Mode:**<br>Indicates the parity sense to check the parity on TDAT bus for port 3.<br>0 = Odd Parity<br>1 = Even Parity<br>**MPHY Mode:**<br>NA | R/W | 0 |
| 6 | Tx_parity_sense Port 2 | **SPHY Mode:**<br>Indicates the parity sense to check the parity on TDAT bus for port 2.<br>0 = Odd Parity<br>1 = Even Parity<br>**MPHY Mode:**<br>NA | R/W | 0 |

1. RO = Read Only, No clear on Read; R = Read, Clear on Read; W = Write only; R/W = Read/Write, No clear; R/W/C = Read/Write, Clear on Write

**Table 146    SPI3 Transmit and Global Configuration ($0x700) (Sheet 3 of 3)**

| Bit | Name | Description | Type[1] | Default |
|-----|------|-------------|---------|---------|
| 5 | Tx_parity_sense Port 1 | **SPHY Mode:**<br>Indicates the parity sense to check the parity on TDAT bus for port 1.<br>0 = Odd Parity<br>1 = Even Parity<br>**MPHY Mode:**<br>NA | R/W | 0 |
| 4 | Tx_parity_sense Port 0 | **SPHY Mode:**<br>Indicates the parity sense to check the parity on TDAT bus for port 0.<br>0 = Odd Parity<br>1 = Even Parity<br>**MPHY Mode:**<br>Indicates the parity sense to check the parity on TDAT bus for all ports.<br>0 = Odd Parity<br>1 = Even Parity | R/W | 0 |
| 3 | Tx_port_enable Port 3 | **SPHY Mode:**<br>0 = Disables the selected SPI3TX port 3.<br>1 = Enables the selected SPI3 TX port 3.<br>**MPHY Mode:**<br>0 = Disables the selected SPI3 TX port 3.<br>1 = Enables the selected SPI3 TX port 3. | R/W | 1 |
| 2 | Tx_port_enable Port 2 | **SPHY Mode:**<br>0 = Disables the selected SPI3 TX port 2<br>1 = Enables the selected SPI3 TX port 2<br>**MPHY Mode:**<br>0 = Disables the selected SPI3 TX port 2<br>1 = Enables the selected SPI3 TX port 2 | R/W | 1 |
| 1 | Tx_port_enable Port 1 | **SPHY Mode:**<br>0 = Disables the selected SPI3 TX port 1<br>1 = Enables the selected SPI3 TX port 1<br>**MPHY Mode:**<br>0 = Disables the selected SPI3 TX port 1<br>1 = Enables the selected SPI3 TX port 1 | R/W | 1 |
| 0 | Tx_port_enable Port 0 | **SPHY Mode:**<br>0 = Disables the selected SPI3 TX port 0<br>1 = Enables the selected SPI3 TX port 0<br>**MPHY Mode:**<br>0 = Disables the selected SPI3 TX port 0<br>1 = Enables the selected SPI3 TX port 0 | R/W | 1 |
| 1. RO = Read Only, No clear on Read; R = Read, Clear on Read; W = Write only; R/W = Read/Write, No clear; R/W/C = Read/Write, Clear on Write | | | | |

**Table 147    SPI3 Receive Configuration ($0x701)  (Sheet 1 of 3)**

| Bit | Name | Description | Type[1] | Default |
|---|---|---|---|---|
| **Register Description:** This register gives the configuration related to the SPI3 receiver. | | | | 0x00000F80 |
| 31:28 | Reserved | Reserved | RO | 0x0 |
| 27 | B2B_PAUSE Port 3 | **SPHY Mode:**<br>Indicates the number of pause cycles to be introduced between back-to-back transfers for port 3.<br>0 = Zero pause cycles<br>1 = Two pause cycles<br>**MPHY Mode:**<br>NA | R/W | 0 |
| 26 | B2B_PAUSE Port 2 | **SPHY Mode:**<br>Indicates the number of pause cycles to be introduced between back-to-back transfers for port 2.<br>0 = Zero pause cycles<br>1 = Two pause cycles<br>**MPHY Mode:**<br>NA | R/W | 0 |
| 25 | B2B_PAUSE Port 1 | **SPHY Mode:**<br>Indicates the number of pause cycles to be introduced between back-to-back transfers for port 1.<br>0 = Zero pause cycles<br>1 = Two pause cycles<br>**MPHY Mode:**<br>NA | R/W | 0 |
| 24 | B2B_PAUSE Port 0 | **SPHY Mode:**<br>Indicates the number of pause cycles to be introduced between back-to-back transfers for port 0.<br>0 = Zero pause cycles<br>1 = Two pause cycles<br>**MPHY Mode:**<br>Indicates the number of pause cycles to be introduced between back-to-back transfers for all ports.<br>0 = Zero pause cycles<br>1 = Two pause cycles | R/W | 0 |
| 23:22 | RX_BURST Port 3 | **SPHY Mode:**<br>NA<br>**MPHY Mode:**<br>NA | R/W | 0x0 |
| 21:20 | RX_BURST Port 2 | **SPHY Mode:**<br>NA<br>**MPHY Mode:**<br>NA | R/W | 0x0 |
| 1.  RO = Read Only, No clear on Read; R = Read, Clear on Read; W = Write only; R/W = Read/Write, No clear; R/W/C = Read/Write, Clear on Write | | | | |

**Table 147    SPI3 Receive Configuration ($0x701) (Continued) (Sheet 2 of 3)**

| Bit | Name | Description | Type[1] | Default |
|---|---|---|---|---|
| 19:18 | RX_BURST Port 1 | **SPHY Mode:**<br>NA<br>**MPHY Mode:**<br>NA | R/W | 0x0 |
| 17:16 | RX_BURST Port 0 | **SPHY Mode:**<br>NA<br>**MPHY Mode:**<br>Selects the maximum burst size on the RX path for all ports.<br>0x = 64 bytes maximum burst size<br>10 = 128 bytes maximum burst size<br>11 = 256 bytes maximum burst size | R/W | 0x0 |
| 15 | Rx_parity_sense Port 3 | **SPHY Mode:**<br>Indicates the parity sense to check the parity on RDAT bus for port 3.<br>0 = Odd Parity<br>1 = Even Parity<br>**MPHY Mode:**<br>NA | R/W | 0x0 |
| 14 | Rx_parity_sense Port 2 | **SPHY Mode:**<br>Indicates the parity sense to check the parity on RDAT bus for port 2.<br>0 = Odd Parity<br>1 = Even Parity<br>**MPHY Mode:**<br>NA | R/W | 0x0 |
| 13 | Rx_parity_sense Port 1 | **SPHY Mode:**<br>Indicates the parity sense to check the parity on RDAT bus for port 1.<br>0 = Odd Parity<br>1 = Even Parity<br>**MPHY Mode:**<br>NA | R/W | 0x0 |
| 12 | Rx_parity_sense Port 0 | **SPHY Mode:**<br>Indicates the parity sense to check the parity on RDAT bus for port 0.<br>0 = Odd Parity<br>1 = Even Parity<br>**MPHY Mode:**<br>Indicates the parity sense to check the parity on RDAT bus for all ports.<br>0 = Odd Parity<br>1 = Even Parity | R/W | 0x0 |

1. RO = Read Only, No clear on Read; R = Read, Clear on Read; W = Write only; R/W = Read/Write, No clear; R/W/C = Read/Write, Clear on Write

**Table 147    SPI3 Receive Configuration ($0x701) (Continued) (Sheet 3 of 3)**

| Bit | Name | Description | Type[1] | Default |
|-----|------|-------------|---------|---------|
| 11 | Rx_port_enable Port 3 | **SPHY Mode:**<br>0 = Disables the selected SPI3 RX port.<br>1 = Enables the selected SPI3 RX port.<br>**MPHY Mode:**<br>0 = Disables the selected SPI3 RX port.<br>1 = Enables the selected SPI3 RX port. | R/W | 0x1 |
| 10 | Rx_port_enable Port 2 | **SPHY Mode:**<br>0 = Disables the selected SPI3 RX port.<br>1 = Enables the selected SPI3 RX port.<br>**MPHY Mode:**<br>0 = Disables the selected SPI3 RX port.<br>1 = Enables the selected SPI3 RX port. | R/W | 0x1 |
| 9 | Rx_port_enable Port 1 | **SPHY Mode:**<br>0 = Disables the selected SPI3 RX port.<br>1 = Enables the selected SPI3 RX port.<br>**MPHY Mode:**<br>0 = Disables the selected SPI3 RX port.<br>1 = Enables the selected SPI3 RX port. | R/W | 0x1 |
| 8 | Rx_port_enable Port 0 | **SPHY Mode:**<br>0 = Disables the selected SPI3 RX port.<br>1 = Enables the selected SPI3 RX port.<br>**MPHY Mode:**<br>0 = Disables the selected SPI3 RX port.<br>1 = Enables the selected SPI3 RX port. | R/W | 0x1 |
| 7 | Rx_core_enable | **SPHY Mode:**<br>NA. Write as 1, ignore on Read.<br>**MPHY Mode:**<br>0 = Disables the RX SPI3 core.<br>1 = Enables the RX SPI3 core. | R/W | 0x1 |
| 6:1 | IBA[5:0] | **SPHY Mode:**<br>NA. Write as 0, ignore on Read.<br>**MPHY Mode:**<br>Sets the 6-bit value appended to the 2-bit address during the port address selection. | R/W | 0x00 |
| 0 | RERR_enable | **SPHY Mode/MPHY Mode:**<br>Frames marked to be filtered (based on the settings in the Section 91, *RX Packet Filter Control ($ Port_Index + 0x19)*) or frames above the Section 81, *Max Frame Size (Addr: Port_Index + 0x0F)* that are not dropped in the RX FIFO (see Section 123, *RX FIFO Errored Frame Drop Enable ($0x59F)*can be optionally indicated with an RERR when sent out the SPI3 interface.<br>0 = Packets not indicated with RERR.<br>1 = Packets indicated with RERR. | R/W | 0 |

1.  RO = Read Only, No clear on Read; R = Read, Clear on Read; W = Write only; R/W = Read/Write, No clear; R/W/C = Read/Write, Clear on Write

**Table 148      Address Parity Error Packet Drop Counter ($0x70A)**

| Bit | Name | Description | Type[1] | Default |
|-----|------|-------------|---------|---------|
| Register Description: This register counts the number of packets dropped due to parity error detection during the address selection cycle. | | | | 0x00000000 |
| 31:8 | Reserved | Reserved | RO | 0x000000 |
| 7:0 | Address Parity Error Packet Drop Counter | This is an 8-bit counter that counts the number of packets dropped due to parity error detection during the address selection cycle. This gets cleared when read and saturates at 8'hFF. There is only one counter for address parity drop as address will be used only in MPHY mode of operation. The counter gets cleared once the register is read. | R | 0x00 |
| 1. RO = Read Only, No clear on Read; R = Read, Clear on Read; W = Write only; R/W = Read/Write, No clear; R/W/C = Read/Write, Clear on Write | | | | |

## 8.4.10      SerDes Register Overview

Table 149 through Table 152, *Clock and Interface Mode Change Enable Ports 0 - 3 ($0x794), on page 211* define the contents of the SerDes registers at base location 0x780, which contain the control and status for the four SerDes interfaces on the IXF1104 MAC.

**Table 149      TX Driver Power Level Ports 0 - 3 ($0x784)**

| Bit | Name | Description | Type | Default |
|-----|------|-------------|------|---------|
| Register Description: Allows selection of various programmable drive strengths on each SerDes port. Refer to Section 5.6.2.2, *Transmitter Programmable Driver-Power Levels*, on page 103. | | | | 0x0000dddd |
| 31:16 | Reserved | Reserved | RO | 0x0000 |
| 15:12 | DRVPWR3[3:0] | Encoded input that sets Power Level for Port 3 | R/W | 1101 |
| 11:8 | DRVPWR2[3:0] | Encoded input that sets Power Level for Port 2 | R/W | 1101 |
| 7:4 | DRVPWR1[3:0] | Encoded input that sets Power Level for Port 1 | R/W | 1101 |
| 3:0 | DRVPWR0[3:0] | Encoded input that sets Power Level for Port 0 | R/W | 1101 |
| 1. RO = Read Only, No clear on Read; R = Read, Clear on Read; W = Write only; R/W = Read/Write, No clear; R/W/C = Read/Write, Clear on Write | | | | |

**Table 150      TX and RX Power-Down ($0x787)**

| Bit | Name | Description | Type | Default |
|-----|------|-------------|------|---------|
| Register Description: TX and RX power-down bits to allow per-port power-down of unused ports | | | | 0x00000000 |
| 31:14 | Reserved | Reserved | RO | 0x0000000 |
| 13:10 | TPWRDWN[3:0] | TX power-down for Ports 3-0 (1 = Power-down) | R/W | 0000 |
| 9:4 | Reserved | Reserved | RO | 0x00 |
| 3:0 | RPWRDWN[3:0] | RX Power-down for Ports 3-0 (1 = Power-down) | R/W | 0000 |
| 1. RO = Read Only, No clear on Read; R = Read, Clear on Read; W = Write only; R/W = Read/Write, No clear; R/W/C = Read/Write, Clear on Write | | | | |

**Table 151** **RX Signal Detect Level Ports 0 - 3 ($0x793)**

| Bit | Name | Description | Type[1] | Default |
|-----|------|-------------|---------|---------|
| **Register Description:** This register shows the status of the Rx input in relation to the level of the signal being received from the line. This register is meant for debug and test use. | | | | 0x00000000 |
| 31:4 | Reserved | Reserved | RO | 0x0000000 |
| 3:0 | SIGDET[3:0] | Signal Detect for Ports 0-3<br>0 = Noise<br>1 = Signal | RO | 0x0 |
| 1.  RO = Read Only, No clear on Read; R = Read, Clear on Read; W = Write only; R/W = Read/Write, No clear; R/W/C = Read/Write, Clear on Write | | | | |

**Table 152** **Clock and Interface Mode Change Enable Ports 0 - 3 ($0x794) (Sheet 1 of 2)**

| Bit | Name | Description | Type[1] | Default |
|-----|------|-------------|---------|---------|
| **Register Description:** This register is used when a change to the operational mode or speed of the IXF1104 MAC is required. This register ensures that when a change is made that the internal clocking of the IXF1104 MAC is managed correctly and no unexpected effects of the operational or speed change are observable on the line interfaces. | | | | 0x00000000 |
| 31:4 | Reserved | Reserved | RO | 0x0000000 |
| 3 | Clock and Interface Mode Change Enable Port 3[2] | Enables internal clock generator for Port 3 to sample the *MAC IF Mode and RGMII Speed ($ Port_Index + 0x10)* and the *Interface Mode ($0x501)*.<br>0 = Set to zero when changes are being made to the *MAC IF Mode and RGMII Speed ($ Port_Index + 0x10)* and the *Interface Mode ($0x501)*.<br>1 = Set to 1 for the configuration changes to take effect. | R/W | 0 |
| 1.  RO = Read Only, No clear on Read; R = Read, Clear on Read; W = Write only; R/W = Read/Write, No clear; R/W/C = Read/Write, Clear on Write<br>2.  Refer to *Section 6.1, Change Port Mode Initialization Sequence*, on page 127 for the proper sequence to change the port mode and speed in conjunction with this register. | | | | |

**Table 152    Clock and Interface Mode Change Enable Ports 0 - 3 ($0x794) (Sheet 2 of 2)**

| Bit | Name | Description | Type[1] | Default |
|---|---|---|---|---|
| 2 | Clock and Interface Mode Change Enable Port 2[2] | Enables internal clock generator for Port 2 to sample the *MAC IF Mode and RGMII Speed ($ Port_Index + 0x10)* and the *Interface Mode ($0x501)*.<br>0 = Set to zero when changes are being made to the *MAC IF Mode and RGMII Speed ($ Port_Index + 0x10)* and the *Interface Mode ($0x501)*.<br>1 = Set to 1 for the configuration changes to take effect. | R/W | 0 |
| 1 | Clock and Interface Mode Change Enable Port 1[2] | Enables internal clock generator for Port 1 to sample the *MAC IF Mode and RGMII Speed ($ Port_Index + 0x10)* and the *Interface Mode ($0x501)*.<br>0 = Set to zero when changes are being made to the *MAC IF Mode and RGMII Speed ($ Port_Index + 0x10)* and the *Interface Mode ($0x501)*.<br>1 = Set to 1 for the configuration changes to take effect. | R/W | 0 |
| 0 | Clock and Interface Mode Change Enable Port 0[2] | Enables internal clock generator for Port 0 to sample the *MAC IF Mode and RGMII Speed ($ Port_Index + 0x10)* and the *Interface Mode ($0x501)*.<br>0 = Set to zero when changes are being made to the *MAC IF Mode and RGMII Speed ($ Port_Index + 0x10)* and the *Interface Mode ($0x501)*.<br>1 = Set to 1 for the configuration changes to take effect. | R/W | 0 |

1. RO = Read Only, No clear on Read; R = Read, Clear on Read; W = Write only; R/W = Read/Write, No clear; R/W/C = Read/Write, Clear on Write
2. Refer to Section 6.1, *Change Port Mode Initialization Sequence*, on page 127 for the proper sequence to change the port mode and speed in conjunction with this register.

## 8.4.11    Optical Module Register Overview

Table 153 through Table 156, *I$^2$C Data Ports 0 - 3 ($0x79F)*, on page 214 provide an overview of the Optical Module Registers.

**Note:**    All registers in this section are only applicable to ports that are configured in fiber mode.

**Table 153    Optical Module Status Ports 0-3 ($0x799)  (Sheet 1 of 2)**

| Bit | Name | Description | Type[1] | Default |
|---|---|---|---|---|
| **Register Description:** This register provides a means to control and monitor the interface to the optical modules when a port is used in fiber mode. | | | | 0x00000000 |
| 31:24 | Reserved | Reserved | RO | 0x00 |
| 23:20 | Rx_LOS_3:0 | Rx_LOS inputs for Ports 0-3 | R | 0x0 |
| 19:14 | Reserved | Reserved | | 0X00 |

1. RO = Read Only, No clear on Read; R = Read, Clear on Read; W = Write only; R/W = Read/Write, No clear; R/W/C = Read/Write, Clear on Write

**Table 153     Optical Module Status Ports 0-3 ($0x799) (Continued) (Sheet 2 of 2)**

| Bit | Name | Description | Type[1] | Default |
|-----|------|-------------|---------|---------|
| 13:10 | Tx_FAULT_3:0 | Tx_FAULT inputs for Ports 0-3 | R | 0x0 |
| 9:4 | Reserved | Reserved | | 0X00 |
| 3:0 | MOD_DEF_3:0 | MOD_DEF inputs for Ports 0-3 | R | 0x0 |
| 1. RO = Read Only, No clear on Read; R = Read, Clear on Read; W = Write only; R/W = Read/Write, No clear; R/W/C = Read/Write, Clear on Write | | | | |

**Table 154     Optical Module Control Ports 0 - 3 ($0x79A)**

| Bit | Name | Description | Type[1] | Default |
|-----|------|-------------|---------|---------|
| **Register Description:** This register provides access to optical module interrupt enables and sets the TX_DISABLE output for the ports configured in fiber mode. | | | | 0x1E000 |
| 31:17 | Reserved | Reserved | RO | 0x0000 |
| 16:13 | $I^2$C_port_enable | When set, individually enables the four $I^2$C ports. | R/W | 0xF |
| 12 | RX_LOS_EN | Enable for RX_LOS_INT operation<br>1 = Enabled | R/W | 0 |
| 11 | TX_FAULT_EN | Enable for TX_FAULT_INT operation<br>1 = Enabled | R/W | 0 |
| 10 | MOD_DEF_EN | Enable for MOD_DEF_INT operation<br>1 = Enabled | R/W | 0 |
| 9:4 | Reserved | Reserved | RO | 0X00 |
| 3:0 | TX_DISABLE_3:0 | Tx_DISABLE outputs for Ports 0-3 | R/W | 0x0 |
| 1. RO = Read Only, No clear on Read; R = Read, Clear on Read; W = Write only; R/W = Read/Write, No clear; R/W/C = Read/Write, Clear on Write | | | | |

**Table 155    I$^2$C Control Ports 0 - 3 ($0x79B)**

| Bit | Name | Description | Type[1] | Default |
|-----|------|-------------|---------|---------|
| **Register Description:** This register controls and monitors the interface to the optical modules when used in fiber mode. | | | | 0x00000000 |
| 31:29 | Reserved | Reserved | RO | 0x0 |
| 28 | Port address Err (R) | Port addressing error. | R | 0 |
| 27 | wp_err | An attempt to write to the protected E$^2$PROM has occurred. | R | 0 |
| 26 | no_ack_err | This bit is set to 1 when a write and subsequent read from an Optical Module Interface has failed. This signal should be used to validate the data being read. Data is only valid if this bit is equal to zero. | R | 0 |
| 25 | I$^2$C_enable | Enable the I$^2$C block. | R/W | 0 |
| 24 | I$^2$C_start | Start the I$^2$C transfer. | R/W | 0 |
| 23 | Reserved | Reserved | RO | 0 |
| 22 | write_complete | Bit is asserted when write access is complete. | R | 0 |
| 21 | Reserved | Reserved | RO | 0 |
| 20 | Read_complete | Bit asserted when read access is complete. | R | 0 |
| 19:18 | Reserved | Reserved | RO | 0 |
| 17:16 | Port Select | Selects the port for which the I$^2$C transaction is targeted. Valid range is 0 to 3. | R/W | 00 |
| 15 | Read/Write | 0 = Write transaction<br>1 = Read transaction | R/W | 0 |
| 14:11 | Device ID | Most-significant four bits of device address field. | R/W | 0x0 |
| 10:0 | Register Address | Bits 10:8 select the least-significant three bits of the device address field<br>Bits 7:0 select the word/register address | R/W | 0x000 |
| 1.  RO = Read Only, No clear on Read; R = Read, Clear on Read; W = Write only; R/W = Read/Write, No clear; R/W/C = Read/Write, Clear on Write | | | | |

**Table 156    I$^2$C Data Ports 0 - 3 ($0x79F)**

| Bit | Name | Description | Type[1] | Default |
|-----|------|-------------|---------|---------|
| **Register Description:** These registers hold data bytes that are read and written using the I$^2$C interface to Optical Module Interfaces connected to each port of the Cortina Systems® IXF1104 4-Port Gigabit Ethernet Media Access Controller. | | | | 0x00000000 |
| 31:24 | Reserved | Reserved | RO | 0x00 |
| 23:16 | Write Data | Bit 23=MSB, Bit 16 = LSB<br>Data to be written to the Optical Module Interface. | R/W | 0X00 |
| 15:8 | Reserved | Reserved | RO | 0x00 |
| 7:0 | Read Data | Bit 7 = MSB, Bit 0 = LSB<br>Data read from the Optical Module Interface. | R/W | 0X00 |
| 1.  RO = Read Only, No clear on Read; R = Read, Clear on Read; W = Write only; R/W = Read/Write, No clear; R/W/C = Read/Write, Clear on Write | | | | |

# 9.0    Mechanical Specifications

The IXF1104 MAC is packaged in a 576-ball BGA package with 6 balls removed diagonally from each corner, for a total of 552 balls used measuring 25 mm x 25 mm. The pitch of the package balls is 1 mm.

## 9.1    Overview

CBGA (standard and RoHS-compliant) and FC-PBGA packages are suited for applications requiring high I/O counts and high electrical performance, and are recommended for high-power applications with high noise immunity requirements.

**Note:**    The FC-PBGA package will not be available until mid-2006. Please see your field sales representative for more detailed information.

### 9.1.1    Features

- Flip chip die attach; surface mount second-level interconnect
- High electrical performance
- High I/O counts
- Area array I/O options
- Multiple power-zone offering supports core and four additional voltages
- JEDEC-compliant package

## 9.2    Package Specifics

The IXF1104 MAC uses the following package:

- 576-ball BGA package with 6 balls removed diagonally from each corner, for a total of 552 balls used
- Ball pitch of 1.0 mm
- Overall package dimensions of 25 mm x 25 mm

## 9.3 Package Information

### 9.3.1 CBGA Package Diagrams

Figure 54 and Figure 55 illustrate the CBGA top, bottom, and side package views.

**Figure 54    CBGA Package Diagram**
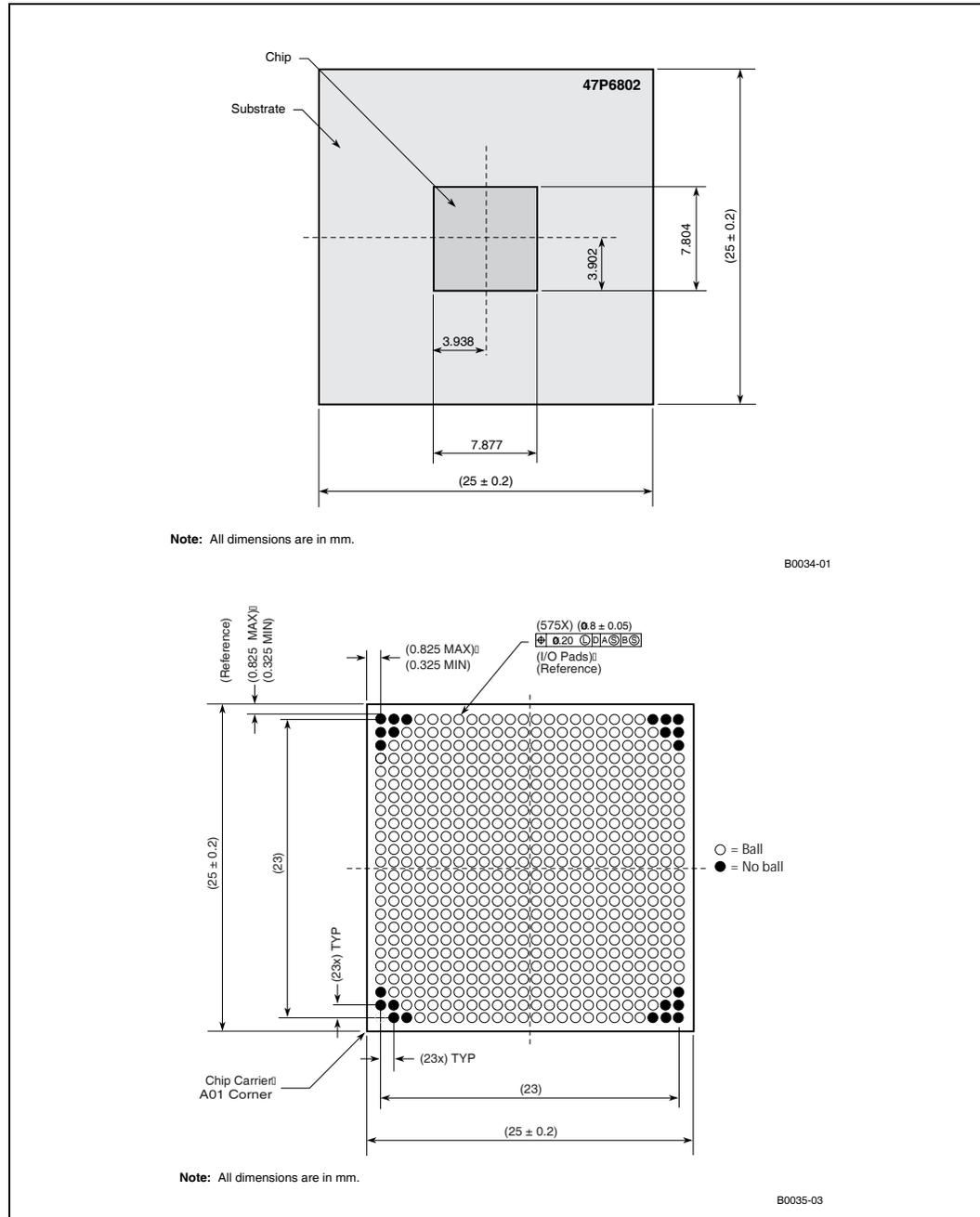
**Figure 55    CBGA Package Side View Diagram**



45L4867  (552)
Solder ball

C4 Encapsulant
Fillet

Chip

0.81 ± 0.1

(2.47  Max)
(2.03  Min)

(6X)  (0.77  Max)
(0.69  Min)

(6X)  (3.24  Max)
(2.72  Min)

(6X)  (4.16  Max)
(3.43  Min)

(0.857  Max)
(0.779  Min)

(3.327  Max)
(2.809  Min)

(4.237  Max)
(3.619  Min)

0.15  C

Seating Plane

**Note:**  All dimensions are in mm.

B0555-01
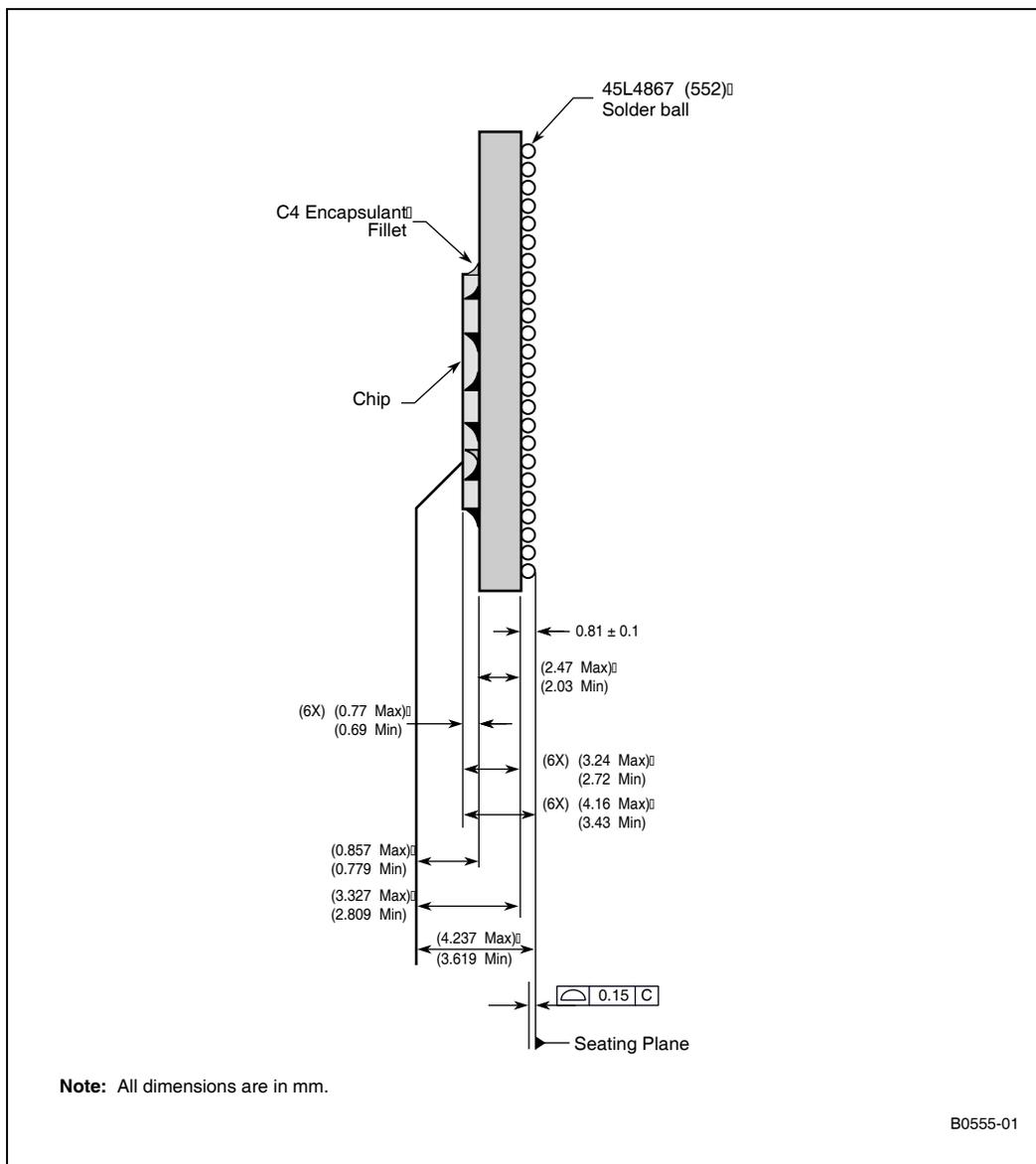
## 9.3.2    Flip Chip–Plastic Ball Grid Array Package Diagram

Figure 56 illustrates the FC-PBGA top and bottom package views and Figure 57 lists the FC-PBGA mechanical specifications.

**Note:**        Please contact your field sales representative for more information on the FC-PBGA package.

**Figure 56     FC-PBGA Package (Top and Bottom Views)**

**Figure 57**      **FC-PBGA Mechanical Specifications**
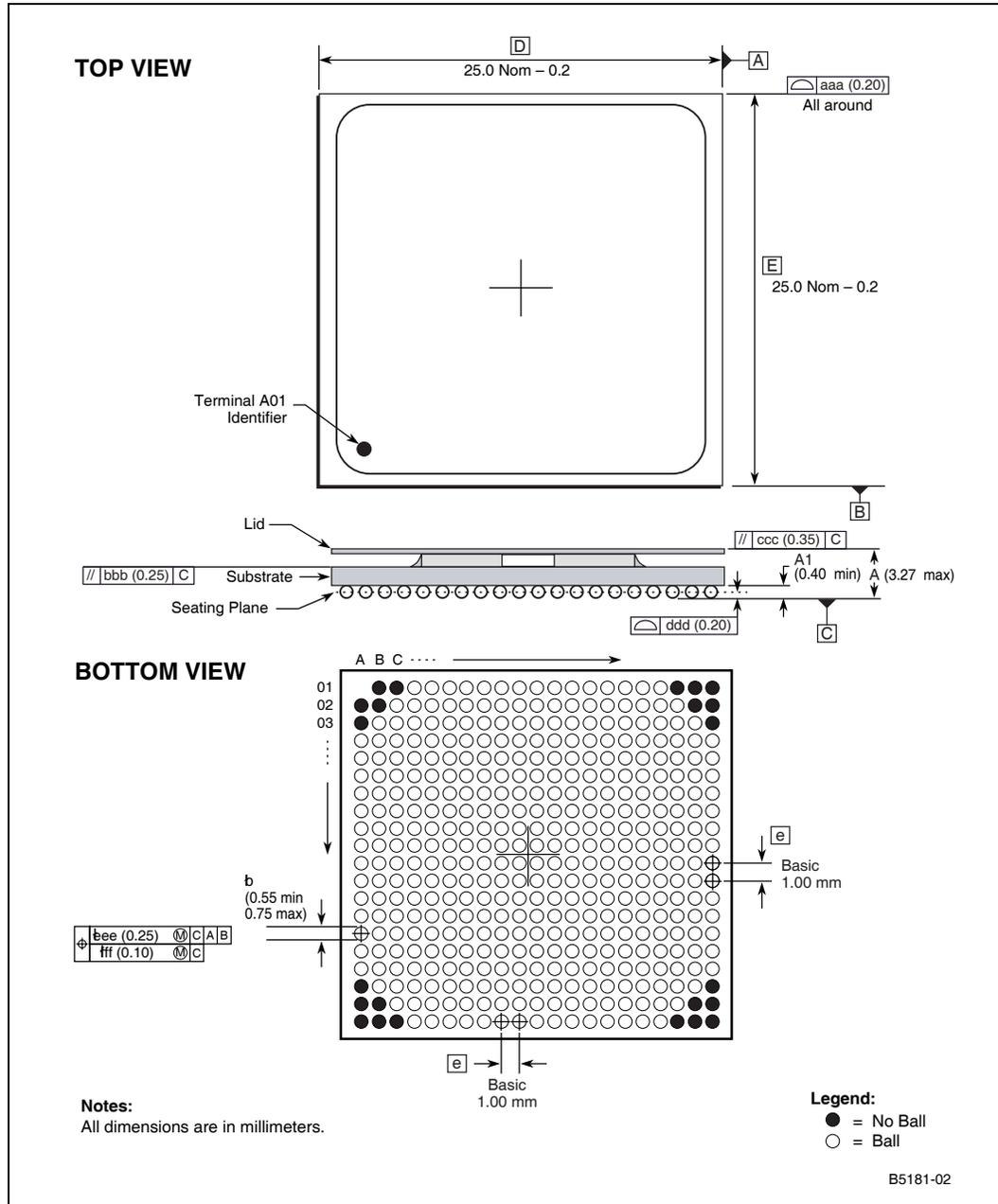
## Mechanical Specifications
### FC-PBGA Mechanical Specifications

| Dimensions[1] | | Package BGA Pitch |
|---|---|---|
| | | 1.00 mm Pitch |
| A | Max | 3.27 |
| A1 | Min | 0.40 |
| D and E | Nom ± 0.2 | 25.0 x 25.0 |
| b | Min | 0.55 |
| | Max | 0.75[2] |
| e | Basic | 1.00 |
| aaa | | 0.20 |
| bbb | | 0.25 |
| ccc | | 0.35 |
| ddd | | 0.20 |
| eee | | 0.25 |
| fff | | 0.10 |

Notes:
1. All dimensions are in millimeters.
2. Indicated feature does not conform to JEDEC MS-034A.

## 9.3.3      Top Label Marking Example

Figure 58 and Figure 59 show example marking labels.

**Note:**      In contrast to the Pb-Free (RoHS-compliant) package, the non-RoHS-compliant package does not have the "e1" symbol.

**Figure 58    Package Marking Example−Cortina Systems Marking**



AAAOOOAAA ← Device Name
AywwX00a ← FPO Traceability Code
Country of Origin

*Note:* This is a generic example. Markings for specific devices will vary.

**Figure 59    Package Marking Example−Intel* Marking**
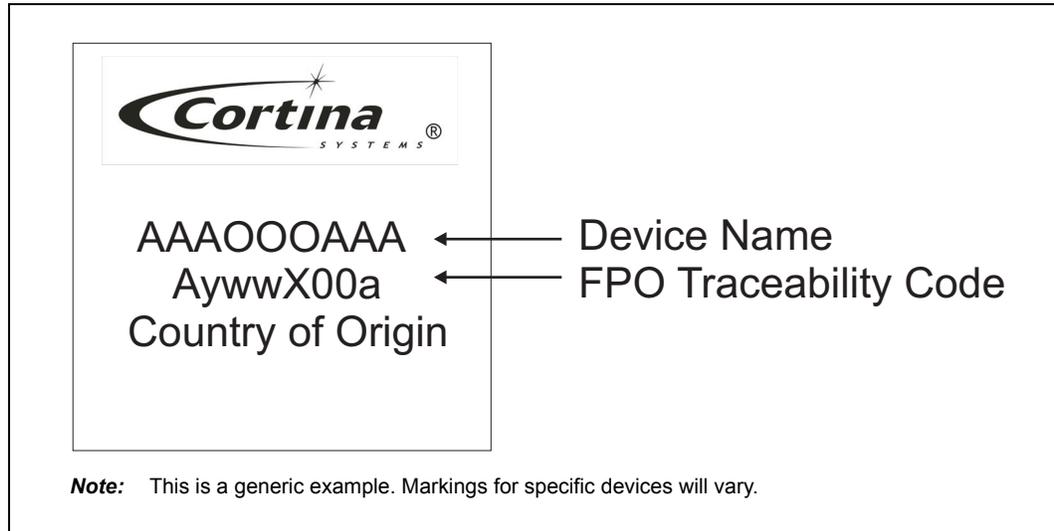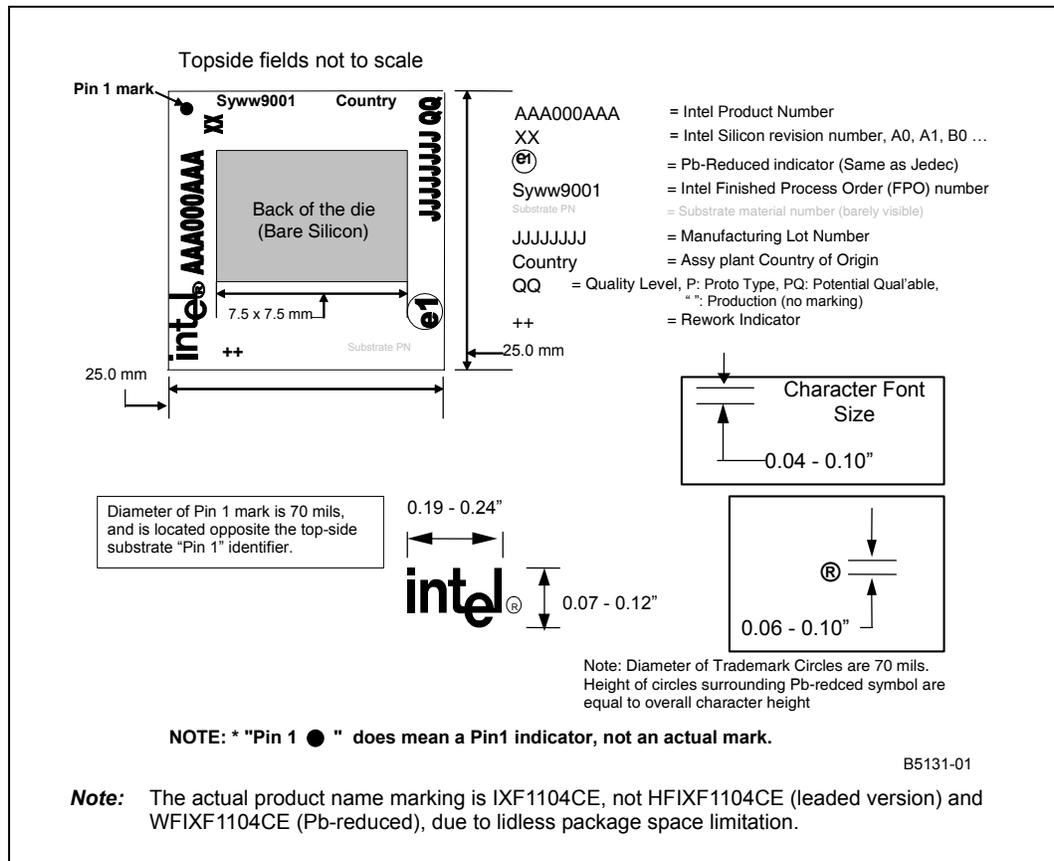


Topside fields not to scale

Pin 1 mark.    Syww9001    Country

XX

Back of the die
(Bare Silicon)

7.5 x 7.5 mm

AAA000AAA

intel ®

++    Substrate PN

25.0 mm

25.0 mm

AAA000AAA    = Intel Product Number
XX    = Intel Silicon revision number, A0, A1, B0 …
(e1)    = Pb-Reduced indicator (Same as Jedec)
Syww9001    = Intel Finished Process Order (FPO) number
Substrate PN    = Substrate material number (barely visible)
JJJJJJJJJ    = Manufacturing Lot Number
Country    = Assy plant Country of Origin
QQ    = Quality Level, P: Proto Type, PQ: Potential Qual'able,
    " ": Production (no marking)
++    = Rework Indicator

Character Font Size
0.04 - 0.10"

Diameter of Pin 1 mark is 70 mils, and is located opposite the top-side substrate "Pin 1" identifier.

0.19 - 0.24"

intel ®    0.07 - 0.12"

® 0.06 - 0.10"

Note: Diameter of Trademark Circles are 70 mils. Height of circles surrounding Pb-redced symbol are equal to overall character height

NOTE: * "Pin 1 ● " does mean a Pin1 indicator, not an actual mark.

B5131-01

*Note:* The actual product name marking is IXF1104CE, not HFIXF1104CE (leaded version) and WFIXF1104CE (Pb-reduced), due to lidless package space limitation.
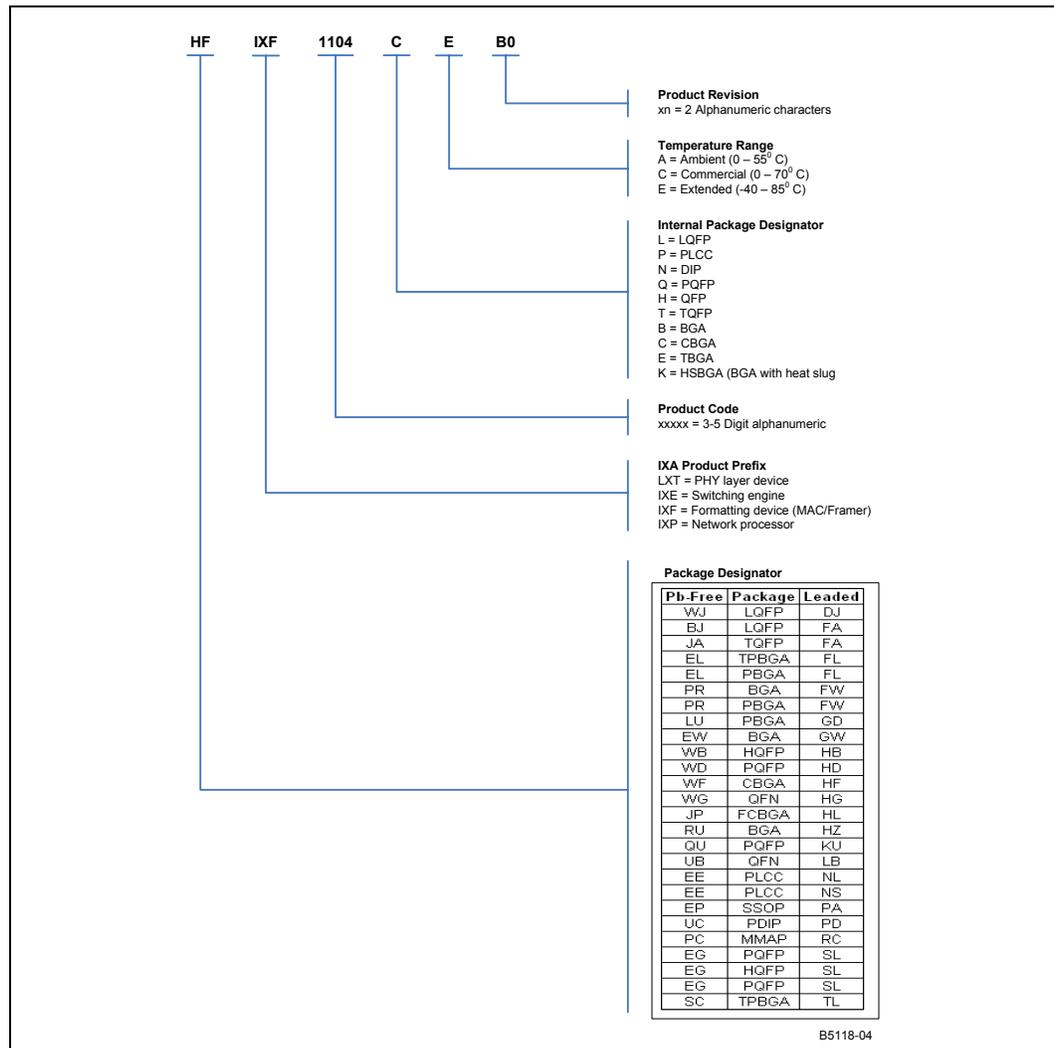
# 10.0 Product Ordering Information

Table 157 and Figure 60 provide IXF1104 MAC product ordering information.

### Table 157 Product Information

| Product Number | Revision | Package Type | RoHS-Compliant |
|---|---|---|---|
| HFIXF1104CE.B0 | B0 | CBGA | No |
| WFIXF1104CE.B0 | B0 | CBGA | Yes |
| HPIXF1104BE.B0[1] | B0 | PBGA | No |

*Note:*
1. Please contact your field sales representative for detailed information on the FC-PBGA package.

### Figure 60 Ordering Information – Sample



HF    IXF    1104    C    E    B0

**Product Revision**
xn = 2 Alphanumeric characters

**Temperature Range**
A = Ambient (0 – 55$^0$ C)
C = Commercial (0 – 70$^0$ C)
E = Extended (-40 – 85$^0$ C)

**Internal Package Designator**
L = LQFP
P = PLCC
N = DIP
Q = PQFP
H = QFP
T = TQFP
B = BGA
C = CBGA
E = TBGA
K = HSBGA (BGA with heat slug)

**Product Code**
xxxxx = 3-5 Digit alphanumeric

**IXA Product Prefix**
LXT = PHY layer device
IXE = Switching engine
IXF = Formatting device (MAC/Framer)
IXP = Network processor

**Package Designator**

| Pb-Free | Package | Leaded |
|---|---|---|
| WJ | LQFP | DJ |
| BJ | LQFP | FA |
| JA | TQFP | FA |
| EL | TPBGA | FL |
| EL | PBGA | FL |
| PR | BGA | FW |
| PR | PBGA | FW |
| LU | PBGA | GD |
| EW | BGA | GW |
| WB | HQFP | HB |
| WD | PQFP | HD |
| WF | CBGA | HF |
| WG | QFN | HG |
| JP | FCBGA | HL |
| RU | BGA | HZ |
| QU | PQFP | KU |
| UB | QFN | LB |
| EE | PLCC | NL |
| EE | PLCC | NS |
| EP | SSOP | PA |
| UC | PDIP | PD |
| PC | MMAP | RC |
| EG | PQFP | SL |
| EG | HQFP | SL |
| EG | PQFP | SL |
| SC | TPBGA | TL |

B5118-04

# Contact Information

**Cortina Systems, Inc.**
**840 W. California Ave**
**Sunnyvale, CA 94086**

**408-481-2300**

**sales@cortina-systems.com**

**www.cortina-systems.com**

**End of Document**