Sundance Technology Incorporated
Proprietary
DO NOT DISTRIBUTE
Design Specification
for
Rio
Gigabit Ethernet MAC

November 10, 1998

Last Update: December 8, 2000 4:38 pm

1.0 Introduction

2.0 Target Applications

The target application for Rio is the network interface portion (either embedded or as part of a NIC) of a file server operating in either a full or half duplex Gigabit Ethernet LAN.

3.0 Assumptions

3.1 Gigabit only

Assumption:

Rio must support Gigabit Ethernet, but will not be used in 10Mbps or 100Mbps Ethernet environments.

Argument:

Early adopters of Gigabit technology will deploy small, dedicated, Gigabit only LAN segments. The expense associated with Gigabit capable clients, servers, and networking gear (i.e. switches) will forgo deployment of gigabit capable workstations in heterogeneous (10/100/1000Mbps) environments. In addition, a multi speed Gigabit NIC will most likely need to support a common medium (i.e. CAT5 UTP) and the 10BASE-T, 100BASE-TX, and 1000BASE-TX standards. Since the 1000BASE-T standard is still in progress, the likelihood of a NIC with a 10/100/1000Mbps requirement is small.

3.2 GMII and SERDES PHY support

Assumption:

Rio must support two interfaces to PHY devices. Both the standard GMII and a non-standard, defacto interface popular among fiber PHY vendors (referred to as the TBI interface).

Argument:

The IEEE 802.3z Gigabit Ethernet standard specifies an extension to the MII introduced in 802.3u Fast Ethernet. The new standard interface, known as GMII is to be used for PHY and MAC device interconnection. Prior to definition of the GMII, several vendors (Amcc, Vitesse) have created PHY SERDES devices to support fiber optic (1000BASE-X) Gigabit Ethernet. These devices do not utilize the GMII, but instead have adopted a generic interface. The industry trend appears to be that GMII will be used for 1000BASE-T (copper) and TBI will be used for 1000BASE-X (fiber optic). In order to support these legacy devices, Rio must also provide a generic TBI. The TBI should support as many PHY vendor SERDES devices as possible.

3.3 PCS included (includes 8B10B ENDEC, and Fiber auto-negotiation)

Assumption:

Rio must include the PCS portion of the Gigabit Ethernet protocol stack to support 1000BASE-X PHY devices.

Argument:

The IEEE 802.3z specification defines the GMII as the interface between the MAC and PCS layers within the Gigabit Ethernet protocol stack. None of the 1000BASE-X PHY devices support the GMII (so called SERDES devices) do not incorporate the PCS layer. Therefore, to support these PHY devices Rio must include a Gigabit Ethernet PCS.

3.4 Jumbo frame support

Assumption:

Rio must support Alteon's jumbo frame technology.

Argument:

Jumbo frames or extended frames, as defined by Alteon Networks, are frames that are larger than the 1518 octet (or 1522 octet for VLAN tagged) frames which are specified in the IEEE 802.3 standard. Alteon Networks supports a "Jumbo Frame" option on adapters and switches that allows for the extension of Ethernet frame length up to 9018 (or 9022 for VLAN tagged) octets. Microsoft and 3Com (among others) are showing support for this technology, and tests have shown a Microsoft NT server using jumbo frames has a significant performance improvement over non-jumbo frame environments when operating in a Gigabit Ethernet environment.

3.5 64 bit, 33/66MHz PCI bus support

Assumption:

Rio will support both a 32 bit and 64 bit, 33/66MHz PCI bus.

Argument:

Gigabit Ethernet will most likely be deployed in servers. To utilize the bandwidth provided by Gigabit Ethernet high performance machines will be required. Typically, high end servers utilize 64 bit PCI to maximize their bus bandwidth. Therefore, a 32 bit only MAC would have limited application in Gigabit enabled servers. While support for 66MHz PCI is not expected to be popular among high end servers immediately, 66MHz support is expected to be a competitive feature providing customers with an upgrade path to improve system performance without changing NICs.

4.0 Competitive Products

Sun

Alteon

Packet Engines

Via Technologies

XaQti

Intel (acquired LevelOne who acquired Jato)

5.0 Interoperability & Compatibility

Rubicon must be compatible with and/or interoperate with the following devices and systems.

TABLE 1: Target Rubicon Fiber Optic Transceiver Compatibility

MANUFACTURER	TRANSCEIVER
HP	HFBR-53D5 Family, 850 nm VCSEL HFCT-53D5 Family, 1300 nm FP Laser

TABLE 2: Target Rubicon 1000BASE-X PHY Device Interoperability

MANUFACTURER	PHY DEVICE
Amcc	S2066/2061/2060/2054/2053/2046/2047
AMD	Am79761
HP	HDMP-1636A/1646A
LSI Logic	GigaBlaze Transceiver Core
TI	TNETE2201

TABLE 2: Target Rubicon 1000BASE-X PHY Device Interoperability

MANUFACTURER	PHY DEVICE
Vitesse	VSC7123/7133/7135

TABLE 3: Target Rubicon 1000BASE-T PHY Device Interoperability

MANUFACTURER	PHY DEVICE
Broadcomm	BCM5400
Level1	?
National Semiconductor	?

TABLE 4: Target Rubicon NIC Interoperability

MANUFACTURER	NIC
3Com	Gigabit EtherLink® Server Network Interface Card
Alteon	ACEnic
HP	1000BaseSX Gigabit Ethernet adapter cards
Intel	PRO/1000 Gigabit Server Adapter
Packet Engines	G-NIC™ II Network Interface Card
SGI	Gigabit Ethernet adapter (Alteon OEM)
Solidum Systems	PCI NIC (technology evaluation and OEM)
Sun	GigabitEthernet™ 2.0 adapter
XaQti	Gigabit Development Kit II (GDK II)

TABLE 5: Target Rubicon Hub/Switch/Router Interoperability

MANUFACTURER	HUB/SWITCH/ROUTER
3Com	SuperStack II Hub SuperStack II Switch 9300 SuperStack II Switch 3900 SuperStack II Switch 9000 SuperStack Gigabit Ethernet Module CoreBuilder 3500 Gigabit Ethernet Module CoreBuilder 7000 Gigabit Ethernet Interface Card CoreBuilder 5000 Gigabit Ethernet Module CoreBuilder 9000 Enterprise Switch
Alteon	ACEswitch 180

TABLE 5: Target Rubicon Hub/Switch/Router Interoperability

MANUFACTURER	HUB/SWITCH/ROUTER
Anritsu	MultiFlowTM 1000 Multilayer Switch
Cabletron	SmartSwitch 9000 with Gigabit Ethernet module SmartSTACK Switch with Gigabit Ethernet module GIGAswitch/Ethernet (a DEC product)
Cisco	Catalyst 5509 Switching System Catalyst 5000 Gigabit Ethernet/EtherChannel modules
Compaq	GIGAswitch/Ethernet (a DEC product)
Extreme Networks	Black Diamond 6800 Switch Summit1 Switch Summit4 Switch Summit24 Switch Summit28 Switch Summit Virtual Chassis SummitGbX Switch
Foundry	FastIron Workgroup Switch FastIron Backbone Switches TurboIron Backbone Switches NetIron Switching Routers TurboIron Switching Routers ServerIron Switch BigIron Switching Router FastIron II Wiring Closet Switch
HP	ProCurve Routing Switch 9308M ProCurve Routing Switch 9304M ProCurve Switch 8000M ProCurve Switch 16000M ProCurve Switch 4000M
IBM	8274 Nways LAN RouteSwitch 8271 Nways Ethernet LAN Switch 8277 Nways Ethernet RouteSwitch 8275 300 Series Switch 8275 200 Series Switch
Intel	Gigabit Express Switch
LANNET (acquired by Lucent)	Cajun P550 Gigabit Switch Cajun P220 Series Gigabit Switches Cajun M770 Switch Cajun P100 Series Switches Cajun M400 Series Switches
NetVantage	NV9200 Series Switches

TABLE 5: Target Rubicon Hub/Switch/Router Interoperability

MANUFACTURER	HUB/SWITCH/ROUTER
Nortel (acquired Bay Networks)	BayStack 350 Switch BayStack 450 Switch Accelar 1000 Series Switches Versalar IP Access Switch 15000
ODS	LANBlazer 7000 Gigabit Switch
Packet Engines	FDR™ Gigabit Ethernet Hub PowerRail™ 5200 Enterprise Routing Switch PowerRail™ 2200 Enterprise Routing Switch PowerRail™ 1000 Routing Switch
Performance Technologies	Nebula 4000 Workgroup Switch Nebula 6000 Departmental Switch Nebula 8000 Fault Tolerant Backbone Switch
Plaintree	WaveSwitch 9200 WaveSwitch 9202
XLNT	Millennium 4000 Gigabit Ethernet LAN Switch

TABLE 6: Primary Target Rubicon Operating Systems

CLASS	OPERATING SYSTEM
Microsoft	WINDOWS NT SERVER 4.0
	Windows NT Server Enterprise Edition
	Windows 2000 Server
Novell	Netware 4.2
	Netware 5
Linux	Kernel version 2.2.x Popular distributions: Caldera OpenLinux Debian Eonova Mandrake PHT TurboLinux (popular in Japan) Red Hat Slackware Stampede S.u.S.E. (popular in Germany and Europe)

TABLE 7: Secondary Target Rubicon Operating Environments

CLASS	OPERATING SYSTEM
Microsoft	Windows NT Workstation 4.0
	Windows NT 4.0 Server Terminal Server Edition
Sun	Solaris 2.6, 2.7

TABLE 8: Target Rubicon Systems

MANUFACTURER	SYSTEM
Dell	PowerEdge 6300/6350
Compaq	Proliant 6000/7000
DEC	AlphaServer 800/1200/DS20, Server 3300/3300R/5300/ 7300/7300R
HP	LXr 8000, LH 4/4r,
IBM	Netfinity 7000, RS/6000 260
Unisys	Aquanta 5045R/5045/5043/ES 2045R/ES 2045
SGI	320/540/O2/Origin200/200 Gigachannel/2000

5.1 Standards Compliance

Rubicon must be compliant with the following standards:

IEEE 802.3 1998 Edition

IEEE 802.3z,x,p,Q

PCI Bus Rev 2.2

ACPI Rev 1.0

PC 2001

6.0 Features

PCI Features

- 32/64 bit bus operation
- 33/66MHz bus operation
- Bus master capability
- PCI Specification Revision 2.2 compliant

DMA Features

- Efficient DMA operation maximizes PCI bandwidth utilization
- 1 Terabyte (40 bit) address space
- Scatter, gather transmit/receive DMA
- Dynamic transmit/receive resource allocation
- Transmit and receive descriptor polling

- Transmit "interrupt-less" mode of operation
- IEEE 802.1Q Datalink Layer prioritization
- Transmit frame priority queuing
- Receive frame priority interrupts
- Receive interrupt coalescing
- Timer based interrupt mode

FIFO Features

- Large transmit & receive FIFO
- No external memory required
- Receive FIFO flow control thresholds
- Transmit and receive FIFO occupancy priority DMA requests

MAC Features

- IEEE 802.3 1998 Edition compliant
- 1000Mbps, 100Mbps, 10Mbps triple speed operation
- Full wire speed support
- Half/full duplex operation
- Half duplex carrier extension and packet bursting
- Asymmetric/symmetric flow control
- Transmit and receive back to back frame support
- VLAN support
- IP, TCP, UDP checksum calculation/verification
- Rich statistic register set
- 64bit hash table for receive multicast frame filtering
- Jumbo frame support, transmit and receive
- GMII support
- MII support

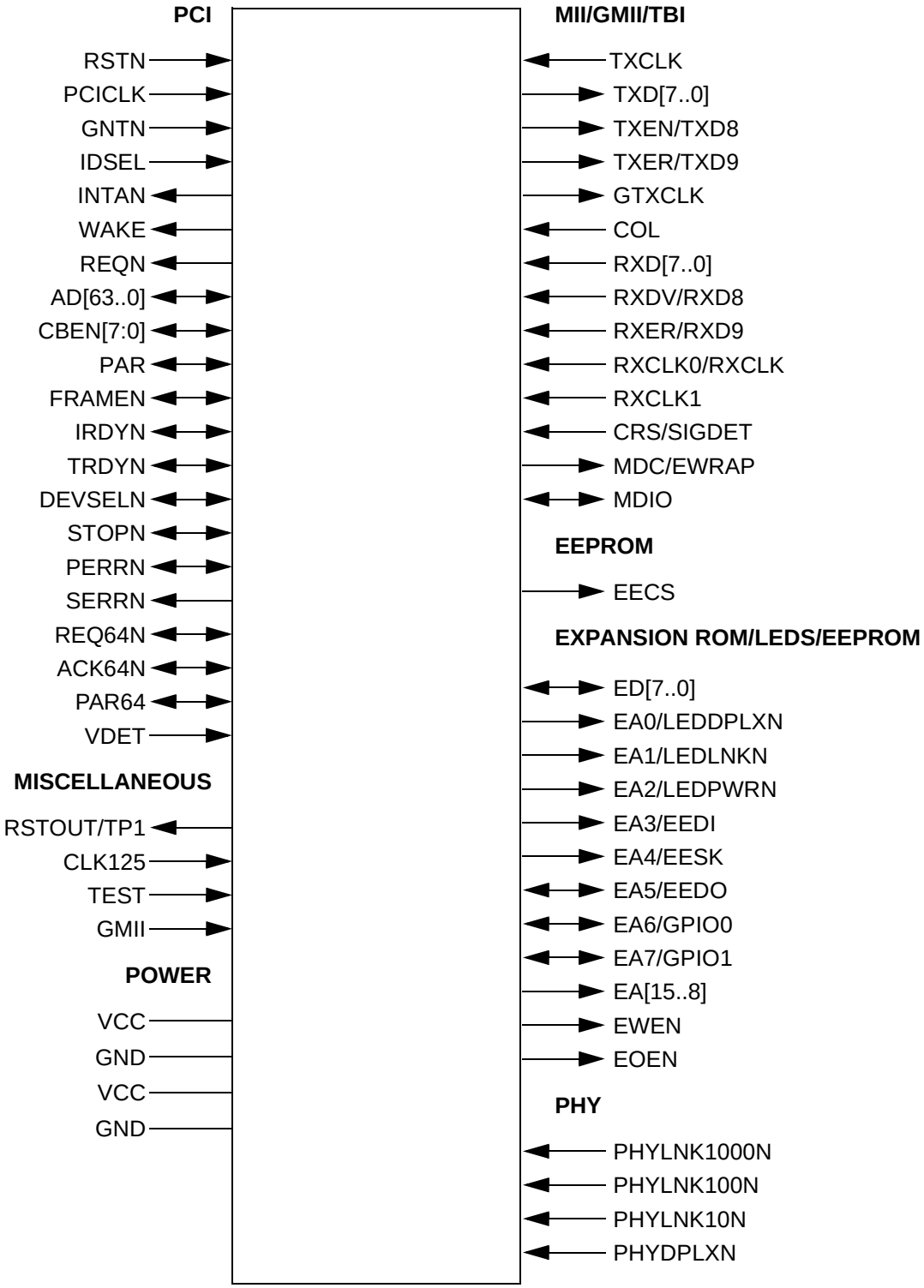
Gigabit PCS Features

- For use with non-GMII enabled Gigabit PHY devices
- Ten Bit Interface (TBI)
- Bypass mode
- 1000BASE-X compliant
- 8B10B encoder/decoder
- Auto negotiation
- PCS related PHY management registers

Power Management

- WakeOnLAN support
- ACPI support

7.0 Signals


PCI
RSTN
PCICLK
GNTN
IDSEL
INTAN
WAKE
REQN
AD[63..0]
CBEN[7:0]
PAR
FRAMEN
IRDYN
TRDYN
DEVSELN
STOPN
PERRN
SERRN
REQ64N
ACK64N
PAR64
VDET
MISCELLANEOUS
RSTOUT/TP1
CLK125
TEST
GMII
POWER
VCC
GND
VCC
GND
MII/GMII/TBI
TXCLK
TXD[7..0]
TXEN/TXD8
TXER/TXD9
GTXCLK
COL
RXD[7..0]
RXDV/RXD8
RXER/RXD9
RXCLK0/RXCLK
RXCLK1
CRS/SIGDET
MDC/EWRAP
MDIO
EEPROM
EECS
EXPANSION ROM/LEDS/EEPROM
ED[7..0]
EA0/LEDDPLXN
EA1/LEDLNKN
EA2/LEDPWRN
EA3/EEDI
EA4/EESK
EA5/EEDO
EA6/GPIO0
EA7/GPIO1
EA[15..8]
EWEN
EOEN
PHY
PHYLNK1000N
PHYLNK100N
PHYLNK10N
PHYDPLXN

8.0 Signal Descriptions

SIGNAL NAME	SIGNAL TYPE	SIGNAL DESCRIPTION
PCI		
RSTN	INPUT	Reset, asserted LOW. RSTN will cause the Rio to reset all of its functional blocks. RSTN must be asserted for a minimum duration of 10 PCICLK cycles.
PCICLK	INPUT	PCI Bus Clock. This clock is used to drive the PCI bus interfaces and the internal DMA logic. All bus signals are sampled on the rising edge of PCICLK. PCICLK can operate from 0MHz to 66MHz.
GNTN	INPUT	PCI Bus Grant, asserted LOW. GNTN signals access to the PCI bus has been granted to Rio.
IDSEL	INPUT	Initialization Device Select. The IDSEL is used to select the Rio during configuration read and write transactions.
INTAN	OUTPUT	Interrupt Request, asserted LOW. The Rio asserts INTAN to request an interrupt, when any one of the programmed interrupt event occurs.
WAKE	OUTPUT	Wake Event, assertion level is programmable. The Rio asserts WAKE to signal the detection of a wake event. The WAKE signal eventually drives the PCI bus PME# signal, but is not intended to be directly connected to PME#. See the PCI Bus Power Management Interface Specification for details on generating PME# from WAKE.
REQN	OUTPUT	Request, asserted LOW. The Rio asserts REQN to request PCI bus master operation.
AD [63..0]	IN/OUT	PCI Bus Address/Data. Address and data are multiplexed on the AD pins. Bits 0 through 31 carry the lower 32 bits of the physical address during the first clock cycle of a transaction, and carry data during the subsequent clock cycles. Bits 32 through 63 carry the upper 32 bits of the physical address during the first clock cycle of a transaction (if the Dual Address Cycle, or DAC command is used along with the assertion of REQ64N), and carry an additional 32 bits of data during subsequent clock cycles (if a 64 bit transaction has been negotiated through the assertion of both REQ64N and ACK64N).
CBEN [7..0]	IN/OUT	PCI Bus Command/Byte Enable, asserted LOW. Bus command and byte enables are multiplexed on the CBEN signals. CBEN specify the bus command during the address phase transaction, and carry byte enables during the data phase. CBEN bits 4 through 0 are reserved unless a 64 bit transaction has been negotiated through the assertion of both REQ64N and ACK64N.

TABLE 9: Rio Signal Descriptions

SIGNAL NAME	SIGNAL TYPE	SIGNAL DESCRIPTION
PAR	IN/OUT	Parity. PCI Bus parity is even across bits 0 through 31 of AD and bits 0 through 3 of CBEN. The Rio generates PAR during address and write data phases as a bus master, and during read data phase as a target. It checks for correct parity during read data phase as bus master, during every address phase as a bus slave, and during write data phases as a target.
FRAMEN	IN/OUT	PCI Bus Cycle Frame, asserted LOW. FRAMEN is asserted at the beginning of the address phase of the bus transaction and deasserted before the final transfer of the data phase of the transaction.
IRDYN	IN/OUT	Initiator Ready, asserted LOW. A bus master asserts IRDYN to indicate valid data phases on AD during write data phases, and to indicate it is ready to accept data during read data phases. A target will monitor IRDYN.
TRDYN	IN/OUT	Target Ready, asserted LOW. A bus target asserts TRDYN to indicate valid read data phases, and to indicate it is ready to accept data during write data phases. A bus master will monitor TRDYN.
DEVSELN	IN/OUT	Device Select, asserted LOW. The Rio asserts DEVSELN when it is selected as a target during a bus transaction. It monitors DEVSELN for any target to acknowledge a bus transaction initiated by the Rio.
STOPN	IN/OUT	Stop, asserted LOW. STOPN is driven by the slave target to inform the bus master to terminate the current transaction.
PERRN	IN/OUT	Parity Error, asserted LOW. The Rio asserts PERRN when it checks and detects a bus parity errors. When it is generating PAR output, the Rio monitors for any reported parity error on PERRN.
SERRN	OUTPUT	System Error, asserted LOW.
REQ64N	IN/OUT	64 Bit Transaction Request, asserted LOW. The REQ64N signal is asserted by a bus master to indicate a preference for 64 bit transactions.
ACK64N	IN/OUT	64 Bit Transaction Acknowledgment, asserted LOW. The ACK64N signal is asserted by a bus target which supports 64 bit transactions in response to REQ64N assertion.
PAR64	IN/OUT	Parity. PCI Bus parity is even across bits 32 through 63 of AD and bits 4 through 7 of CBEN. The Rio generates PAR64 during address and write data phases as a bus master, and during read data phase as a target. It checks for correct parity during read data phase as bus master, during every address phase as a bus slave, and during write data phases as a target.

TABLE 9: Rio Signal Descriptions

SIGNAL NAME	SIGNAL TYPE	SIGNAL DESCRIPTION
VDET	INPUT	Power Detect. The Rio detects PCI bus power supply loss when VDET is LOW.
MII/GMII/TBI		
TXCLK	INPUT	Transmit Clock. TXCLK is a continuous clock supplied by the PHY to synchronize transmit data transfers when the Rubicon is configured in 10Mbps or 100Mbps Mode. The nominal rate of TXCLK is 2.5MHz for 10Mbps operation, and 25MHz for 100Mbps operation.
TXD [7..0]	OUTPUT	Transmit Data. Eight bits of transmit data from the transmit MAC to the PHY device. When the Rio is configured in GMII Mode, these signals represent the entire transmit data bus. TXD are synchronized to the GTXCLK. When the Rio is configured in TBI Mode, these pins represent the lower 8 bits of the 10 bit transmit data bus. TXD are synchronized with GTXCLK.
TXEN/TXD8	OUTPUT	Transmit Enable/Transmit Data Bit 9. When the Rubicon is configured in GMII mode, this signal is TXEN. When asserted, TXEN/TXD8 indicates to the PHY that TXD carry valid transmit data. TXEN/TXD8 is asserted with the first byte of the preamble until the last byte of the frame data. When the Rubicon is configured in TBI Mode, this signal is TXD[8], the ninth transmit data bit. TXEN/TXD8 is synchronous with GTXCLK.
TXER/TXD9	OUTPUT	Transmit Error/Transmit Data Bit 10. When the Rubicon is configured in GMII mode, this signal is TXER. When asserted, TXER/TXD9 indicates to the PHY that an error symbol should be transmitted in place of the TXD transmit data. When the Rubicon is configured in TBI Mode, this signal is TXD[9], the tenth transmit data bit. TXER/TXD9 is synchronous with GTXCLK.
GTXCLK	OUTPUT	Gigabit Transmit Clock. GTXCLK is a continuous clock supplied to the PHY to synchronize transmit data transfers when the Rubicon is configured in GMII Mode or TBI Mode. The nominal rate of GTXCLK is 125MHz.
COL	INPUT	Collision. When the Rubicon is configured in TBI Mode, COL is ignored and should be tied LOW. When the Rubicon is configured in GMII Mode, COL is asserted by the PHY to signal a collision condition is detected on the physical medium. COL is asynchronous.

TABLE 9: Rio Signal Descriptions

SIGNAL NAME	SIGNAL TYPE	SIGNAL DESCRIPTION
RXD [7..0]	INPUT	Receive Data. 8 bits of receive data from the PHY device to the receive MAC. When the Rio is configured in GMII Mode, these signals represent the entire receive data bus. In GMII Mode, RXD are synchronized to the RXCLK0/RXCLK. When the Rio is configured in TBI Mode, these pins represent the lower 8 bits of the 10 bit receive data bus. In TBI Mode, RXD are synchronized with both RXCLK0/RXCLK and RXCLK1.
RXDV/RXD8	INPUT	Receive Data Valid/Receive Data Bit 9. When the Rubicon is configured in GMII Mode, this signal is RXDV, indicating valid frame data is present on the RXD pins. The PHY asserts RXDV/RXD8 before the SFD, and de-asserts it after the last data nibble of the frame. In GMII mode, RXDV/RXD8 is synchronous with RXCLK0/RXCLK. When the Rubicon is configured in TBI Mode, this signal is RXD[8], the ninth receive data bit. In TBI mode, RXDV/RXD8 is synchronous with both RXCLK0/RXCLK and RXCLK1.
RXER/RXD9	INPUT	Receive Error/Receive Data Bit 10. When the Rubicon is configured in GMII Mode, this signal is RXER, an indication from the PHY that an error has been detected during frame data reception. In GMII mode, RXER/RXD9 is synchronous with RXCLK0/RXCLK. When the Rubicon is configured in TBI Mode, this signal is RXD[9], the tenth receive data bit. In TBI mode, RXER/RXD9 is synchronous with both RXCLK0/RXCLK and RXCLK1.
RXCLK0/ RXCLK	INPUT	Receive Clock Zero. When the Rubicon is configured in GMII Mode, RXCLK0/RXCLK provide the timing reference for RXD, RXDV/RXD8, and RXER/RXD9 signals. The nominal rate for RXCLK0/RXCLK is 125MHz. When the Rubicon is configured in TBI Mode, RXCLK0/RXCLKand RXCLK1 together provide a timing reference for the 10 bit data bus defined by RXD, RXDV/RXD8, and RXER/RXD9 signals. RXCLK0/ RXCLK rising edges correspond to odd-numbered code-groups appearing on the 10 bit data bus defined by RXD, RXDV/RXD8, and RXER/RXD9 signals. RXCLK0/RXCLK and RXCLK1 are supplied by the PHY based on the receive clock recovery circuit. RXCLK0/RXCLK and RXCLK1 are 180 degrees out of phase. Nominal rate for RXCLK0/ RXCLK is 62.5MHz. RXCLK0/RXCLK may be stretched, but is never shortened.

TABLE 9: Rio Signal Descriptions

SIGNAL NAME	SIGNAL TYPE	SIGNAL DESCRIPTION
RXCLK1	INPUT	Receive Clock One. When the Rubicon is configured in GMII Mode, RXCLK1 is ignored and should be tied LOW. When the Rubicon is configured in TBI Mode, RXCLK0/RXCLKand RXCLK1 together provide a timing reference for the 10 bit data bus defined by RXD, RXDV/RXD8, and RXER/RXD9 signals. RXCLK0/ RXCLK rising edges correspond to even-numbered code-groups appearing on the 10 bit data bus defined by RXD, RXDV/RXD8, and RXER/RXD9 signals. RXCLK0/RXCLK and RXCLK1 are supplied by the PHY based on the receive clock recovery circuit. RXCLK0/RXCLK and RXCLK1 are 180 degrees out of phase. Nominal rate for RXCLK1 is 62.5MHz. RXCLK1 may be stretched, but is never shortened.
CRS/ SIGDET	INPUT	Carrier Sense/Signal Detect. When the Rio is configured in GMII Mode, this signal is CRS, asserted by the PHY to signal a non-idle medium with either transmit or receive activity detected. CRS is asynchronous. When the Rubicon is configured in TBI Mode, this signal is SIGDET. An indication from the media transceiver that a signal is present.
MDC/ EWRAP	OUTPUT	Management Data Clock/Electrically Wrap. When the Rubicon is configured in GMII Mode, this signal is MDC which is used to synchronize the read and write operations of MDIO. When the Rio is configured in TBI Mode, this signal is EWRAP which is asserted HIGH to indicate the PHY device should internally loop transmit data to the receiver, and should keep the transmitter serial outputs in a static state. MDC/EWRAP asserted LOW indicates the PHY device should operate normally.
MDIO	IN/OUT	Management Data Input/Output. When the Rubicon is configured in TBI Mode, MDIO is ignored and should be tied HIGH through a resistor (10kΩ or higher). When the Rio is configured in GMII Mode, MDIO carries management data for the management port read and write operations.
PHY INTERFACE		
PHYLNK1000N	INPUT	PHY Link 100Mbps Status, asserted LOW. PHYLNK1000N is driven by the physical layer device. It is asserted to signal a functional 1000Mbps link (link up).
PHYLNK100N	INPUT	PHY Link 100Mbps Status, asserted LOW. PHYLNK100N is driven by the physical layer device. It is asserted to signal a functional 100Mbps link (link up).
PHYLNK10N	INPUT	PHY Link 100Mbps Status, asserted LOW. PHYLNK10N is driven by the physical layer device. It is asserted to signal a functional 10Mbps link (link up).

TABLE 9: Rio Signal Descriptions

SIGNAL NAME	SIGNAL TYPE	SIGNAL DESCRIPTION
PHYDPLXN	INPUT	PHY Duplex Status, assertion level is programmable. PHYDPLXN is driven by the physical layer device. It is asserted to indicate a full duplex link, and de-asserted to indicate a half duplex link. PHYDPLXN is undefined when PHYLNK10N is not asserted.
EEPROM INTERFACE (SEE ALSO EXPANSION ROM INTERFACE/LED DRIVERS)		
EECS	OUTPUT	EEPROM Chip Select. EECS is asserted by the Rio to access the EEPROM. EECS is connected directly to the chip select input of the EEPROM device.
EXPANSION ROM INTERFACE/LED DRIVERS		
ED [7..0]	IN/OUT	Expansion ROM Data. ED provide data access to the expansion ROM.
EA15/ LEDLNK10N	OUTPUT	Expansion ROM Address Bit 15/10Mbps Link Status LED. EA4/EESK is a shared pin assuming the value of bit 15 of the expansion ROM address during accesses to the expansion ROM (see ExpRomAddr and ExpRomData registers). If the expansion ROM is not being accessed, EA4/EESK is the 10Mbps Link Status LED driver signal.
EA14/ LEDLNK100 N	OUTPUT	Expansion ROM Address Bit 14/100Mbps Link Status LED. EA3/EEDI is a shared pin assuming the value of bit 14 of the expansion ROM address during accesses to the expansion ROM (see ExpRomAddr and ExpRomData registers). If the expansion ROM is not being accessed, EA3/EEDI is the 100Mbps Link Status LED driver signal.
EA [13..8]	OUTPUT	Expansion ROM Address. The EA along with EA7/GPIO1, EA6/ GPIO0, EA5/EEDO, EA4/EESK, EA3/EEDI, EA2/LEDPWRN, EA1/ LEDLNK1000N, and EA0/LEDDPLXN carry the address to the expansion ROM.
EA7/GPIO1	IN/OUT	Expansion ROM Address Bit 7/General Purpose Input/Output 1. EA7/ GPIO1 is a shared pin assuming the value of bit 7 of the expansion ROM address during accesses to the expansion ROM (see ExpRomAddr and ExpRomData registers). If the expansion ROM is not being accessed, EA7/GPIO1 is the general purpose input/output bit 1.
EA6/GPIO0	IN/OUT	Expansion ROM Address Bit 7/General Purpose Input/Output 0. EA6/ GPIO0 is a shared pin assuming the value of bit 6 of the expansion ROM address during accesses to the expansion ROM (see ExpRomAddr and ExpRomData registers). If the expansion ROM is not being accessed, EA6/GPIO0 is the general purpose input/output bit 0.

TABLE 9: Rio Signal Descriptions

SIGNAL NAME	SIGNAL TYPE	SIGNAL DESCRIPTION
EA5/EEDO	IN/OUT	Expansion ROM Address Bit 5/EEPROM Data Output. EA5/EEDO is a shared pin assuming the value of bit 5 of the expansion ROM address during accesses to the expansion ROM (see ExpRomAddr and ExpRomData registers). If the expansion ROM is not being accessed, EA5/EEDO is an input connected directly to the data output of the EEPROM device.
EA4/EESK	OUTPUT	Expansion ROM Address Bit 4/EEPROM Serial Clock. EA4/EESK is a shared pin assuming the value of bit 4 of the expansion ROM address during accesses to the expansion ROM (see ExpRomAddr and ExpRomData registers). If the expansion ROM is not being accessed, EA4/EESK is the clock (frequency 150kHz) used to synchronize the EEPROM data access with EA3/EEDI and EA5/EEDO. EA4/EESK is connected directly to the clock input of the EEPROM device.
EA3/EEDI	OUTPUT	Expansion ROM Address Bit 3/EEPROM Data Input. EA3/EEDI is a shared pin assuming the value of bit 3 of the expansion ROM address during accesses to the expansion ROM (see ExpRomAddr and ExpRomData registers). If the expansion ROM is not being accessed, EA3/EEDI is the data input to an EEPROM. EA3/EEDI is connected directly to the data input of the EEPROM device.
EA2/ LEDPWRN	OUTPUT	Expansion ROM Address Bit 2/Power Status LED. EA2/LEDPWRN is a shared pin assuming the value of bit 2 of the expansion ROM address during accesses to the expansion ROM (see ExpRomAddr and ExpRomData registers). If the expansion ROM is not being accesses, EA2/LEDPWRN is the Power Status LED driver signal.
EA1/ LEDLNK1000N	OUTPUT	Expansion ROM Address Bit 2/1000Mbps Link Status LED. EA1/ LEDLNK1000N is a shared pin assuming the value of bit 2 of the expansion ROM address during accesses to the expansion ROM (see ExpRomAddr and ExpRomData registers). If the expansion ROM is not being accesses, EA1/LEDLNK1000N is the 1000Mbps Link Status LED driver signal.
EA0/ LEDDPLXN	OUTPUT	Expansion ROM Address Bit 2/Power Status LED. EA0/LEDDPLXN is a shared pin assuming the value of bit 2 of the expansion ROM address during accesses to the expansion ROM (see ExpRomAddr and ExpRomData registers). If the expansion ROM is not being accesses, EA0/LEDDPLXN is the Duplex Status LED driver signal.
EWEN	OUTPUT	Expansion ROM Write Enable.
EOEN	OUTPUT	Expansion ROM Output Enable.

TABLE 9: Rio Signal Descriptions

SIGNAL NAME	SIGNAL TYPE	SIGNAL DESCRIPTION
MISCELLANEOUS		
RSTOUT	OUTPUT	Reset Output. RSTOUT is the reset output from the Rio. The Rio will assert RSTOUT when it is being reset. RSTOUT is intended to be used to reset other circuitry on the adapter.
TEST	INPUT	Test Mode. TEST asserted HIGH during reset places the Rubicon in TEST mode. For normal operation, TEST is LOW.
GMII	INPUT	Physical Layer Mode. GMII asserted HIGH places the Rubicon in GMII Mode. GMII asserted LOW places the Rubicon in TBI Mode.
CLK125	INPUT	125MHz Clock. CLK125 is the reference clock used for the Rubicon's internal logic.
POWER AND GROUND		
VCCIO	POWER	+3.3 volt power supply for input/output circuits.
GNDIO	GROUND	+3.3 volt return.
VCCCORE	POWER	+2.5 volt power supply for internal core logic.
GNDCORE	GROUND	+2.5 volt return.

TABLE 9: Rio Signal Descriptions

9.0 Pin Designations

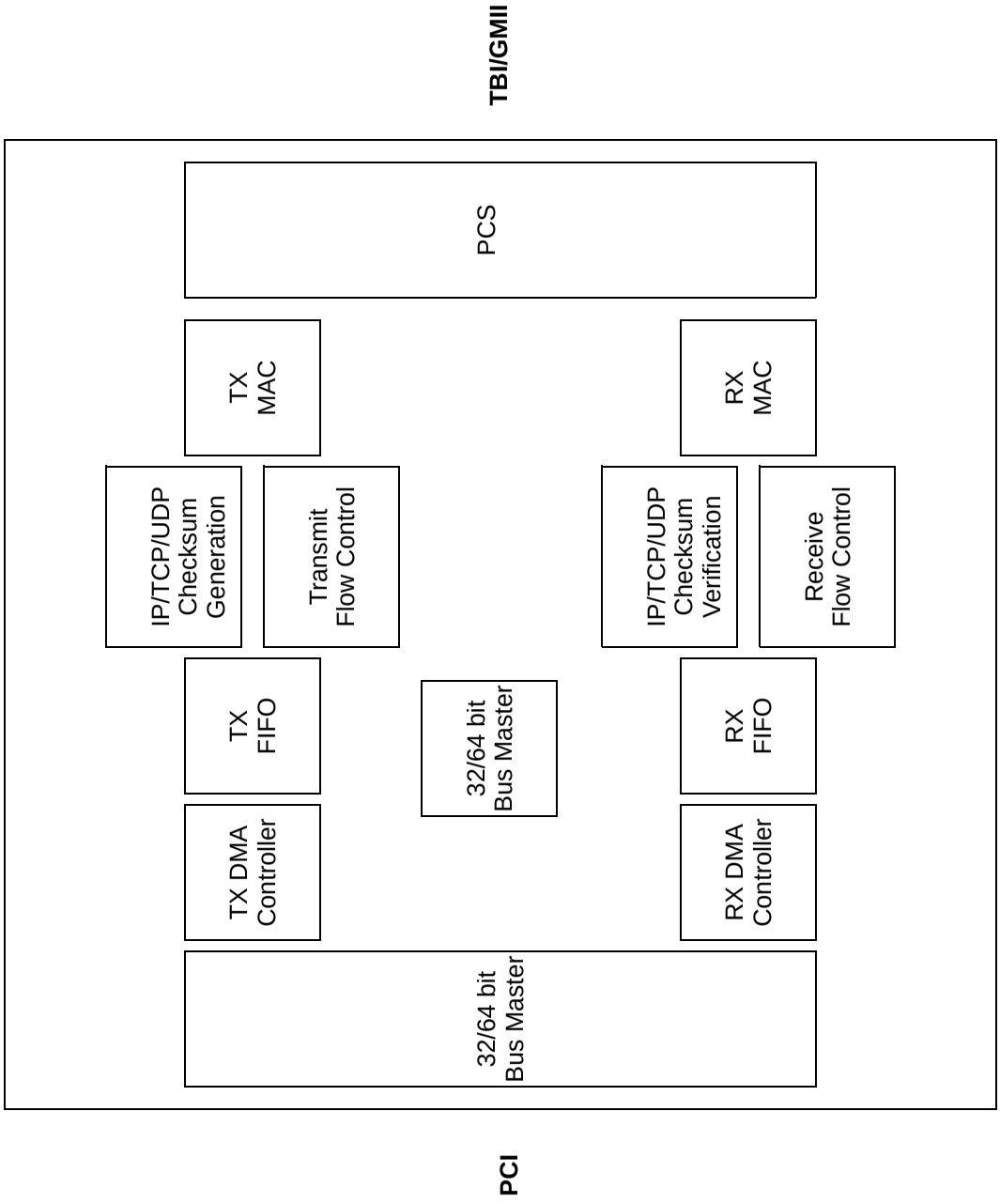
#	PIN NAME	#	PIN NAME	#	PIN NAME	#	PIN NAME
1	MDIO	53	AD17	105	VCCIO	157	ED5
2	MDC/EWRAP	54	AD16	106	AD62	158	ED4
3	CRS/SIGDET	55	VCCIO	107	AD61	159	ED3
4	COL	56	CBEN2	108	AD60	160	VCCIO
5	GNDIO	57	FRAMEN	109	AD59	161	ED2
6	GTXCLK	58	IRDYN	110	GNDIO	162	ED1
7	VCCIO	59	GNDIO	111	AD58	163	ED0
8	TXER/TXD9	60	TRDYN	112	AD57	164	EA0/LEDDPLXN
9	TXEN/TXD8	61	DEVSELN	113	AD56	165	EA1/ LEDLNK1000N
10	TXD7	62	STOPN	114	VCCIO	166	GNDIO
11	TXD6	63	PERRN	115	AD55	167	EA2/LEDPWRN
12	TXD5	64	VCCIO	116	AD54	168	EA3/EEDI
13	TXD4	65	SERRN	117	AD53	169	EA4/EESK
14	GNDIO	66	PAR	118	GNDIO	170	EA5/EEDO
15	TXD3	67	CBEN1	119	AD52	171	EA6/GPIO0
16	TXD2	68	GNDIO	120	GNDCORE	172	VCCIO
17	TXD1	69	GNDCORE	121	AD51	173	EA7/GPIO1
18	GNDCORE	70	AD15	122	VCCCORE	174	GNDCORE
19	TXD0	71	VCCCORE	123	AD50	175	EA8
20	VCCCORE	72	AD14	124	AD49	176	EA9
21	TXCLK	73	AD13	125	VCCIO	177	EA10
22	WAKE	74	AD12	126	AD48	178	EA11
23	INTAN	75	VCCIO	127	AD47	179	EA12
24	RSTN	76	AD11	128	GNDIO	180	EA13
25	VCCIO	77	AD10	129	AD46	181	VCCCORE
26	PCICLK	78	GNDIO	130	AD45	182	EA14/ LEDLNK100N
27	GNDIO	79	AD9	131	AD44	183	EA15/ LEDLNK10N
28	GNTN	80	AD8	132	AD43	184	GNDIO
29	REQN	81	CBEN0	133	AD42	185	CLK125
30	AD31	82	AD7	134	VCCIO	186	VCCIO
31	AD30	83	AD6	135	AD41	187	TEST
32	AD29	84	VCCIO	136	GNDIO	188	GMII
33	GNDIO	85	VCCCORE	137	GNDCORE	189	RSTOUT
34	GNDCORE	86	AD5	138	AD40	190	GNDCORE
35	AD28	87	GNDCORE	139	VCCCORE	191	PHYDPLXN
36	VCCCORE	88	GNDIO	140	AD39	192	PHYLNK10N
37	VCCIO	89	AD4	141	AD38	193	PHYLNK100N

TABLE 10: Rio Pin Designations

#	PIN NAME	#	PIN NAME	#	PIN NAME	#	PIN NAME
38	AD27	90	AD3	142	AD37	194	PHYLNK1000N
39	AD26	91	AD2	143	AD36	195	RXD0
40	AD25	92	AD1	144	VCCIO	196	RXD1
41	AD24	93	AD0	145	AD35	197	RXD2
42	CBEN3	94	GNDIO	146	GNDIO	198	RXD3
43	GNDIO	95	ACK64N	147	AD34	199	RXD4
44	IDSEL	96	VCCIO	148	AD33	200	RXD5
45	AD23	97	REQ64N	149	AD32	201	RXD6
46	VCCIO	98	CBEN7	150	VDET	202	RXD7
47	AD22	99	CBEN6	151	EECS	203	RXDV/RXD8
48	AD21	100	CBEN5	152	EWEN	204	RXER/RXD9
49	AD20	101	CBEN4	153	EOEN	205	GNDIO
50	AD19	102	GNDIO	154	GNDIO	206	RXCLK0/RXCLK
51	GNDIO	103	PAR64	155	ED7	207	VCCIO
52	AD18	104	AD63	156	ED6	208	RXCLK1

TABLE 10: Rio Pin Designations

10.0 Block Diagram



11.0 Functional Description

11.1 PCI

The PCI Bus Interface (PBI) implements the procedures and algorithms needed to link the Rio to a PCI bus. The Rio can be either a PCI bus master or slave. The PBI is also responsible for managing the DMA interfaces and the host processor access to the Rio registers. Arbitration logic within the PBI block accepts bus requests from the TxDMA Logic and RxDMA Logic. The arbiter services the four requests in the fixed priority order of:

RxDMA Urgent Request

TxDMA Urgent Request

RxDMA Request

TxDMA Request

The PBI also manages interrupt generation for a host processor.

The Rio supports all of the PCI memory commands and decides on a burst-by-burst basis which command to use in order to maximize bus efficiency. The list of PCI memory commands is shown below. For all commands, “read” and “write” are with respect to the Rio (i.e. read implies the Rio obtains information from an off-chip location, write implies the Rio sends information to an off-chip location).

Memory Read (MR)

Memory Read Line (MRL)

Memory Read Multiple (MRM)

Memory Write (MW)

Memory Write Invalidate (MWI)

MR is used for all fetches of descriptor information. For reads of transmit frame data, MR, MRL, or MRM is used, depending upon the remaining number of bytes in the fragment, the amount of free space in the TxFIFO, and whether the RxDMA Logic is requesting a bus master operation.

MW is used for all descriptor writes. Writes of receive frame data use either MW or MWI, depending upon the remaining number of bytes in the fragment, the amount of frame data in the RxFIFO, and whether the TxDMA Logic is requesting a bus master operation.

The Rio provides two configuration bits to control the use of advanced memory commands. The MWIEnable bit in the ConfigCommand configuration register allows the host to enable or disable the use of MWI. The MWIDisable bit in the DMACtrl register allows the host system the ability to disable the use of MWI PCI command. MWIDisable is cleared by default, enabling MWI commands.

The Rio provides a set of registers that control the PCI burst behavior. These registers allow a trade-off to be made between PCI bus efficiency and underrun/overrun frequency. Arbitration logic within the PCI Bus Interface block accepts bus requests from the TxDMA Logic and RxDMA Logic. The TxDMA Logic uses the TxDMABurstThresh register, as described in section 11.2.1, to delay the bus request until there is enough free space in the transmit FIFO for a long, efficient burst. The TxDMA Logic can also make an urgent bus request as described in section 11.2.1, where burst efficiency is sacrificed in favor of avoiding a transmit FIFO underrun condition.

The RxDMA process is described in section 11.2.2. Typically, RxDMA requests will be forwarded to the Arbiter, however RxDMA Urgent Requests are also possible in order to prevent receive overruns. The Arbiter services the four requests in the fixed priority order as described in section 11.1.

In support of bus isolation requirements for system states in which the Rio is powered down, all Rio PCI outputs will enter the tri-state condition when the RSTN is active.

11.1.1 Reset

When the host system issues a reset to the Rio via the AsicCtrl register, a delay of at least 5ms is required before any register access should be attempted.

11.1.2 Operating Speed Changes

The Rio is capable of operating in 1000Mbps, 100Mbps, or 10Mbps mode of operation. Swithcing between these three modes must be controlled by the host system using the DebugCtrl register. Upon host system intialization of Rio, the dbSpeedSel bit within the DebugCtrl register must be set to allow for manual speed selection. When a link event interrupt is detected via the LinkEvent bit of the IntStatus register, the PhyCtrl register LinkSpeed field must be read to determine which speed at which the external Physical Layer device is operating. The dbSpeed field within the DebugCtrl register must then be set to the matching value.

11.2 DMA

The Rio implements scatter gather Direct Memory Access (DMA) for moving data from the Rio to/from the host's system memory. Two independent DMA processes are used to transfer transmit data from host system memory to the Rio (transmit DMA), and to transfer receive data from the Rio to host system memory (receive DMA).

11.2.1 Transmit DMA

To utilize the Rio to transmit data onto a Gigabit Ethernet network, the data to be transmitted must be transferred from the host's system memory to the Rio. The data bus utilized by the Rubicon for this data transfer is the PCI bus, and the method for transferring the data is DMA. The locations within system memory which contain the data to be transmitted are indicated to the Rubicon using Transmit Frame Descriptors.

The Transmit Frame Descriptor (TFD) is a data structure containing fields specifying a pointer to another TFD (the TFDNextPtr field), control information (the TFC field), and from one to 15 pointers to locations within system memory containing the Ethernet frame data (the FragInfo fields). The format of a TFD is described in section 11.10. The TFD is used to indicate to the Rio which blocks of system memory comprise the Ethernet frame data to be transmitted. Each Ethernet frame is described by one and only one TFD.

TFDs are typically grouped into linked lists (called TFD lists) within system memory by the host system. The TFDNextPtr field is used to link one TFD to the next in the list. The location of the first TFD in a TFD list is indicated to the Rio by writing the memory location of the first TFD to the TFDListPtr register. Upon reset, the TFDListPtr register contains a null value (0x0000000000) indicating to the Rio that there is no data to transfer via the transmit DMA process. Once a TFD list has been created in system memory, and the location of the first TFD in the list has been written to the TFDListPtr register, the transmit DMA process begins. Once started, the Rio transmit DMA process will proceed as follows:

1. The Rio reads the TFC field to determine how to process the TFD. If the TFDDone bit is a logic 1, the transmit DMA process pauses and the Rio begins polling the TFDDone bit until it is a logic 0.
2. If any fragments are contained in the TFD, up to five FragInfo fields (based on the value in the TxBurstLimit field of the DMACtrl register) are transferred to the Rio via a single DMA transaction.
3. The data fragment referenced by the FragInfo fields is transferred to the Rio via separate DMA transactions (one transaction for each fragment).
4. The process of transferring FragInfo fields, followed by the corresponding fragment data continues until the entire TFD has been processed by the Rio.
5. When the Rio finishes transferring all of the Ethernet frame fragments from host system memory, the Rio checks for a transmit FIFO underrun condition. If an underrun has occurred, the transmit DMA process is halted requiring a transmit path reset to restart.
6. If no transmit FIFO underrun condition occurs and if the DMACtrl register TxWriteBackDisable bit is a logic 0, the Rio sets the TFDDone bit in the TFD's TFC field to a logic 1 (indicating the Rio is finished, and the host system is now the owner of the TFD).

7. If the TFDNextPtr field in the TFD is a null value (0x0000000000), the transmit DMA process pauses and the Rio begins polling the TFDNextPtr field. If the TFDNextPtr field is non zero, the Rio will proceed on to the next TFD in the list (as specified by the TFDNextPtr field) and return to step 1

There are two types of transmit DMA polling. If the TFDNextPtr field of the last TFD is 0x0000000000, the Rubicon will periodically examine (or poll) the memory location of the last TFD's TFDNextPtr field for a value other than 0x0000000000. If the TFDNextPtr field of the last TFD is not 0x0000000000, then the Rio will proceed on to the next TFD and read the new TFC. If the TFDDone bit within the TFC field is a logic 1, the Rio will periodically examine (or poll) the memory location of the new TFD's TFC field until the TFDDone bit is a logic 0. Note, polling of the TFDDone bit within the TFC field is disabled if the DMACtrl register TxWriteBackDisable bit is a logic 1. The rate at which the Rubicon checks for a new TFDNextPtr value, or the TFDDone bit is defined by the TxDMAPollPeriod register. The host system may also force an examination (or poll) of the TFDNextPtr field or the TFC field's TFDDone bit (as applicable) using the "poll now" function of the Rio (see section 11.2.1.1.2).

There are several cases during transmit DMA operation in which host system interrupts are generated. See section 11.3.1 for details on transmit DMA interrupts.

11.2.1.1 Host System Transmit Procedure

The general procedure to accomplish a DMA transfer from host system memory to the Rubicon is shown below. The starting point for this procedure is after system start up, with the Rubicon reset and configured with appropriate register values to facilitate communication with the host system using the PCI bus.

1. Configure the Rubicon for transmit DMA operation (this step is typically performed only once):

- Reset the transmit data path within the Rubicon by setting the following bits in the AsicCtrl register: TxReset, DMA, FIFO, Network.
- Set the TxStartThresh register to configure when data is transmitted by the Rubicon onto a Gigabit Ethernet network. Use of the TxStartThresh register is not recommended due to the high frame rates associated with Gigabit Ethernet LAN segments (see the TxStartThresh register definition for more details).
- Set the TxDMABurstThresh register to configure when the Rubicon will make transmit DMA requests.
- Set the TxDMAUrgentThresh register to configure when the Rubicon will make urgent (i.e. high bus priority) transmit DMA requests.
- Set the TxDMAPollPeriod register.

2. The host system creates a TFD list specifying the Ethernet frame data to be transmitted (see section 11.2.1.2 for details on TFD list creation). The host system writes the value of the memory location of the first TFD in the list to the Rubicon TFDListPtr register. After the first TFD list has been created and indicated to the Rio (by writing to the TFDListPtr register), the host system does not write to the Rubicon TFDListPtr register again, unless the Rio transmit DMA process is reset.
3. The Rio processes the TFD list, transferring Ethernet frame data from host system memory to the transmit FIFO via one or more DMA transactions. The Rio will begin transmitting the frame transferred via DMA (the point at which the Rio begins frame transmission depends on the configuration of the TxStartThresh register and the TFC field of frame's TFD). If the Rio issues an interrupt, the host system processes the interrupt (see section 11.3.1).
4. To transmit additional Ethernet frames, the host system must create a new TFD list (specifying the new data to transmit), and write the value of the memory location of the first TFD in the new TFD list to the TFDNextPtr of the last TFD in the previous list. In addition, in the new TFD list the host system must set the TFDDone bit within each TFD's TFC field to a logic 0. The host system returns to step 3.

Occasionally, while the Rio is processing a TFD list, the host system may wish to halt the processing of the current TFD list to insert one or more TFDs (to transmit high priority Ethernet frames for example). See section 11.2.1.1.1 for the TFD insertion procedure.

Finally, during idle periods between processing of TFD lists, an Ethernet frame may become ready for transmission, with a corresponding TFD list present in system memory, before the Rio poll timer expires. If the host system wishes to avoid delaying the new Ethernet frames by the time it takes for the poll period timer to expire, follow the “poll now” procedure indicated in section 11.2.1.1.2.

11.2.1.1.1 TFD Insertion

Given an existing TFD list located somewhere within system memory, currently being processed by the Rubicon but not yet completed, to insert a TFD list within the existing list (to insert high priority Ethernet frames for example):

1. The host system must traverse the TFD list (starting from the beginning of the current list) checking the TFDDone bit of each TFD’s TFC field until a 0 value for TFDDone bit is found.
2. For the first TFD with TFC field TFDDone bit set to 0 (referred to as the “break point TFD“), the host system stores the value of the TFDNextPtr field (holding the memory location of the next TFD, referred to as the “continuation TFD“), and modifies the TFDNextPtr field of the break point TFD to 0x0000000000.
3. The host system verifies the TFDDone bit of the break point TFD’s TFC field is still 0. If the TFDDone bit of the break point TFD’s TFC has changed to 1, the host system should restore the break point TFD’s TFDNextPtr field to the value of the continuation TFD, and must return to step 1.
4. If the host system verified the TFDDone bit of the break point TFD’s TFC field is still 0, then the Rio has not yet processed the break point TFD, and since the TFDNextPtr is 0x0000000000 the Rio will pause when it reaches this TFD.
5. The host system sets the TFDNextPtr of the last TFD in the insertion TFD list to the memory location of the continuation TFD.
6. The host system now modifies the TFDNextPtr of the break point TFD, changing it from 0x0000000000 to the memory location holding the first TFD in the insertion TFD list.
7. The host system may issue a “poll now” command (see section 11.2.1.1.2), or wait for the Rio to automatically poll the TFDNextPtr of the break point TFD to resume transmit DMA operation.

11.2.1.1.2 TFD Poll Now

To force an examination, or poll, of the current TFD’s TFDNextPtr field or TFC field, TFDDone bit:

1. Write to the DMACtrl register, TxDMAPollNow bit.

11.2.1.2 TFD List Considerations

TFD lists can be built within system memory using two main linked list geometries: chain and ring. TFD lists in a chain geometry are built with only the last TFD in the list containing a value of 0x0000000000 for it’s TFDNextPtr field. TFD lists in a ring geometry are built such that no TFD in the list contains a value of 0x0000000000 for it’s TFDNextPtr field. Ring geometry TFD lists are similar to chain geometry TFD lists, except that the TFDNextPtr field for the “last” TFD in the list contains the memory location of the “first” TFD in the list.

The geometry of the TFD list is irrelevant to the Rio, and is chosen based solely on system related issues. To process a TFD list, the Rio is only concerned with two issues:

- Is the current TFD’s TFC field, TFDDone bit a logic 0?
  If not, the Rio will not process the TFD, but instead will poll the TFC field, TFDDone bit until it becomes a logic 0 before processing the TFD.
- Is the current TFD’s TFDNextPtr non-zero?
  After the Rio finishes processing the data within the TFD, it will proceed onto another TFD only if the memory location of the new TFD is found in the TFDNextPtr field of the current TFD. If a value of 0x0000000000 is found in the in the TFDNextPtr field of the current TFD, the Rio transmit DMA operation will begin polling the TFDNextPtr field.

The geometry of the TFD list is relevant to the host system. A chain geometry is processed differently from a ring geometry from the host system’s perspective. When new transmit data becomes available, first a

new TFD list must be created. The location of this TFD list, and the specifics of it's creation, as well as the indication of the new list to the Rio differs depending on the TFD list geometry being used by the host system.

- If the host system is utilizing a chain geometry for it's TFD lists, the host system may create the new TFD list anywhere within host system memory (with the only restriction that the location can be addressed by the Rio's transmit DMA process). The new TFD list is created with the TFC field, TFDDone bit set to a logic 0, and the TFDNextPtr fields updated appropriately (to point from one TFD to the next in the list, with the last TFDNextPtr field set to 0x0000000000). The host system then writes the memory location of the first TFD in the new TFD list to the TFDNextPtr field of the last TFD in the previous list (overwriting the 0x0000000000 value which paused the Rio transmit DMA operation).
- If the host system is utilizing a ring geometry for it's TFD list, the host system must modify the existing TFD list ring. The modifications begin with the TFD following the last TFD processed by the Rio. The host modifies each TFD's TFC (leaving the TFDDone bit at a logic 1), and the FragInfo fields, but not the TFDNextPtr fields. When all of the new Ethernet frame data to be transferred has been described within the modified TFD list, the host system writes a logic 0 to the TFC field TFDDone bit of each modified TFD (which indicates to the Rio to begin processing the TFDs).

11.2.2 Receive DMA

To utilize the Rio to receive data from a Gigabit Ethernet network, the received data must be transferred from the Rio to the host's system memory. The data bus utilized by Rubicon for this data transfer is the PCI bus, and the method for transferring the data is DMA. The locations within system memory reserved for the received data are indicated to Rubicon using Receive Frame Descriptors.

The Receive Frame Descriptor (RFD) is a data structure containing fields specifying a pointer to another RFD (the RFDNextPtr field), status information (the RFS field), and one pointer (the FragInfo field) to a unique, contiguous block of system memory which is reserved for holding the received data. Typically, one RFD will completely specify a single received Ethernet frame. While it is possible to use multiple RFDs to describe a single Ethernet frame, it is not possible to describe multiple Ethernet frames with a single RFD. The RFD format is described in section 11.10.2.

RFDs are typically grouped into linked lists (called RFD lists) within system memory by the host system. The RFDNextPtr field is used to link RFDs within the list. The location of the first RFD in a RFD list is indicated to the Rio by writing the memory location of the first RFD to the RFDListPtr register. Upon reset, the RFDListPtr register contains a value of 0x0000000000 indicating to the Rio that there is no room in host system memory reserved for received data. Once a RFD list has been created in system memory, and the location of the first RFD in the list has been written to the RFDListPtr register, the receive DMA process begins. Once started, the Rio receive DMA process will proceed as follows:

1. The Rio clears the Reserved(RxDMAStatus) register.
2. The Rio reads the RFS field for the top RFD in the RFD list.
3. If the RFDDone bit within the RFS field of the RFD is set, the Rio will enter polling mode and issue a RFDListEnd interrupt (if enabled via the IntEnable register RFDListEnd bit). After expiration of the poll timer, the Rio receive DMA process returns to step 2.
4. If the RFDDone bit within the RFS field of the RFD is a logic 0, the Rio proceeds with receive DMA operation, waiting for received data from the Gigabit Ethernet network.
5. When received data is ready for transfer to the host system, the Rio transfers the received data to the memory locations indicated by the RFD(s) FragInfo field.
6. The Rio will update the RFD's RFS field, writing a logic 1 to the RFDDone bit. If this is the first RFD for a given Ethernet frame, the Rio will write a logic 1 to the FrameStart bit within the RFD's RFS field. If the amount of received data, up to the end of the Ethernet frame, is less than or equal to the RFD fragment length, the Rio writes a logic 1 to the FrameEnd bit. Other than the RFDDone, and FrameStart bits, the remainder of the RFS field frame status bits are only written when the FrameEnd bit is written as a logic 1.

7. The Rio reads the RFD's RFDNextPtr field.
8. If the RFD's RFDNextPtr field is 0x0000000000, the Rubicon will enter polling mode and issue a RFDListEnd interrupt (if enabled via the IntEnable register RFDListEnd bit). After expiration of the poll timer, the Rio receive DMA process returns to step 7.
9. If the RFD's RFDNextPtr field is a value other than 0x0000000000 the Rio receive DMA process returns to step 2.

The rate at which Rubicon checks for a new RFDNextPtr value, or the RFDDone bit within the RFS field is defined by the RxDMAPollPeriod register. The host system may also force a examination (or poll) of the RFDNextPtr field or the RFS field's RFDDone bit (as applicable) using the "poll now" function of the Rio.

There are several cases during receive DMA operation in which host system interrupts are generated. See section 11.3.3 for details on receive DMA interrupts.

11.2.2.1 Host System Receive Procedure

The general procedure to accomplish a DMA transfer from Rio to host system memory is shown below. The start point for this procedure is after system start up, with Rio reset and configured with appropriate register values to facilitate communication with the host system using the PCI bus.

1. Configure the Rio for receive DMA operation (this step is typically performed only once):

- Reset the receive data path within the Rio by setting the following bits in the AsicCtrl register: RxReset, DMA, FIFO, Network.
- Set the RxEarlyThresh register to configure when data received by the Rio from a Gigabit Ethernet network is transferred to host system memory (if desired, not recommended).
- Set the RxDMABurstThresh register to configure when the Rio will make receive DMA requests.
- Set the RxDMAUrgentThresh register to configure when the Rio will make urgent (i.e. high bus priority) receive DMA requests.
- Set the RxDMAPollPeriod register to configure the Rio polling period.

2. Create a RFD list within system memory (see section 11.2.2.1.3 for details on RFD list creation). Write the value of the memory location holding first RFD in the list to the Rio RFDListPtr register. After the first RFD list has been created and indicated to the Rio (by writing to the RFDListPtr register), the host system does not write to the Rubicon RFDListPtr register again, unless the Rio receive DMA process is reset.
3. When new Ethernet frames are received the Rio processes the RFD list, transferring the received Ethernet frame data from the receive FIFO to host system memory via one or more DMA transactions. The Rio may issue a RxEarly interrupt, if configured, which the host system would process. Typically interrupts related to received Ethernet frames are coalesced (see section 11.3.4). When the host system detects (via an interrupt) that new Ethernet frame data has been transferred to system memory, the RFD list must be processed by the host system. Any errors during frame reception are indicated in the RFS fields of the RFDs corresponding to frames which encountered errors.
4. When additional Ethernet frames are received, the Rio will transfer this data to host system memory only if a RFD is available to receive the data. For the last RFD processed by the Rio, the host system must assure the RFD's RFDNextPtr field holds the location in system memory of the next available RFD. In addition, the host system must assure the next RFD's RFS field, RFDDone bit is a logic 0. The host system returns to step 3.

Occasionally, while the Rio is processing a RFD list, the host system may insert one or more RFDs. See section 11.2.2.1.1 for the RFD insertion procedure.

Finally, if receive DMA processing pauses (due to the Rio encountering an RFD with RFDNextPtr field set to 0x0000000000, or RFS field RFDDone bit a logic 1), the host system must eventually modify the RFD list to enable further receive DMA processing. The host system may wish for the Rio to begin processing the modified RFD list before the Rio poll timer expires. If the host system wishes to avoid delaying any

newly received Ethernet frames by the time it takes for the poll period timer to expire, follow the “poll now” procedure indicated in section 11.2.2.1.2.

11.2.2.1.1 RFD Insertion

If the host system desires to modify the RFD list (a rare requirement), follow this procedure. Given an existing RFD list located somewhere within system memory, to insert a RFD within the list the host system must:

1. Traverse the RFD list checking the RFDNextPtr field and the RFS field RFDDone bit of each RFD. The first RFD found whose RFDNextPtr field is 0x0000000000, or whose RFS field RFDDone bit is a logic 1 is the “break point” RFD.
2. Store the break point RFD’s RFDNextPtr value (call this the “continuation RFD“), and modify the break point RFD’s RFDNextPtr to the value of the insertion RFD list.
3. For the last RFD in the insertion RFD list, set the RFDNextPtr field to the memory location of the continuation RFD.

11.2.2.1.2 RFD Poll Now

To force an examination, or poll, of the current RFD’s RFDNextPtr field or RFS field, RFDDone bit:

1. Write to the DMACtrl register, RxDMAPollNow bit.

11.2.2.1.3 RFD List Considerations

RFD lists can be built within system memory using two main geometries: chain and ring. RFD lists in a chain geometry are built with the last RFD in the list containing a value of 0x0000000000 for it’s RFDNextPtr field. RFD lists in a chain geometry are built such that no RFD in the list contains a value of 0x0000000000 for it’s RFDNextPtr field. Ring geometry RFD lists are similar to chain geometry RFD lists, except that the RFDNextPtr field for the “last” RFD in the list contains the memory location of the “first” RFD in the list.

The geometry of the RFD list is irrelevant to the Rio, and is chosen based solely on system related issues. To process a RFD list, the Rio is only concerned with two issues:

- Is the current RFD’s RFS field, RFDDone bit a logic 0?
  If not, the Rio will not process the RFD (transfer received Ethernet frame data to the host system memory locations indicated by the RFD’s FragInfo field), but instead will wait for the RFS field, RFDDone bit to become a logic 0 before processing the RFD.
- Is the current RFD’s RFDNextPtr non-zero?
  The Rio will proceed onto another RFD only if the memory location of the new RFD is found in the RFDNextPtr field of the current RFD. If a value of 0x0000000000 is found in the in the RFDNextPtr field of the current RFD, the Rio will receive DMA operation will be paused.

The geometry of the RFD list is relevant to host system. A chain geometry is processed differently from a ring geometry from the host system’s perspective. Before receive data can be transferred to host system memory, a RFD list must be created. The location of this RFD list, and the specifics of it’s creation, as well as the indication of the new list to the Rio differs depending on the RFD list geometry being used by the host system.

- If the host system is utilizing a chain geometry for it’s RFD lists, the host system may create RFD lists anywhere within host system memory (with the only restriction that the location can be addressed by the Rio’s receive DMA process). The new RFD list is created with the RFS field, RFDDone bit set to a logic 0, and the RFDNextPtr fields updated appropriately (to point from one RFD to the next in the list, with the last RFDNextPtr field set to 0x0000000000). The host system then writes the memory location of the first RFD in the new RFD list to the RFDNextPtr field of the last RFD in the previous list (overwriting the 0x0000000000 value which paused the Rio receive DMA operation).
- If the host system is utilizing a ring geometry for it’s RFD list, the host system must modify the existing RFD list ring. After processing the received data indicated by RFD’s with their RFS field RFDDone bit

set to a logic 1, the host system must write a logic 0 to the RFS field RFDDone bit to allow the Rio to re-use the RFD.

11.3 Interrupts

The Rio generates host system processor interrupts via the PCI bus based on events related to transmit and receive DMA operation. It is the responsibility of the host system to detect these interrupts, identify the corresponding condition which caused the interrupt, and take the appropriate action.

At gigabit per second data rates, interrupts related to Gigabit Ethernet frame transmission and reception can quickly overwhelm a host system processor. The Rio incorporates several features for minimizing the number of interrupts generated. These features should be carefully understood and utilized to achieve maximum system performance in Gigabit Ethernet networks.

11.3.1 Transmit DMA Interrupts

Interrupts can be generated by the Rio based on a number of events related to transmit DMA operation:

- TxDMAComplete interrupt is issued after successful transfer of an Ethernet frame to the Rio via transmit DMA with the TxDMAIndicate bit in the TFD's TFC field is a logic 1. Use of this interrupt is not recommended due to the frequency of transmit DMA operations in a Gigabit Ethernet network.
- TxComplete interrupt (frame transmission complete without error) is issued after successful transmission of an Ethernet frame which has already been transferred to the Rio with the TxIndicate bit in the TFD's TFC field is a logic 1. A recommended use of this feature is to avoid setting the TxIndicate bit in every TFD, but instead only set the TxIndicate bit in the last TFD of a TFDList, or in every Nth frame (where N>1).
- TxComplete interrupt (frame transmission encountered an error) is issued if an error occurs during transmission of an Ethernet frame which has already been transferred to the Rio independent of the TxIndicate bit setting in the TFD's TFC field. When an error occurs, the transmit MAC of the Rio is disabled (and must be re-enabled to resume operation). Transmit DMA operation continues in spite of transmit errors except for the case of a transmit underrun error (indicated by the TxUnderrun bit in the TxStatus register). To resume transmit DMA operation after a transmit underrun error, the transmit DMA, transmit FIFO, and transmit MAC functions within Rio must be reset.
- IntRequested interrupt is issued after expiration of the Rio Countdown timer register. The Countdown timer register can be programmed to generate an interrupt at fixed intervals.

A common use of interrupts during transmit DMA operation is to determine which TFDs have been successfully transmitted so the host system can free the memory occupied by old TFDs. Interrupts however usually incur a significant cost in terms of host system performance, requiring a large percentage of processor time to service. While interrupts are expensive, memory is usually abundant, therefore a trade off which minimizes interrupts in exchange for more memory usage is desirable.

11.3.2 "Interrupt-Less" Transmit DMA

Rio's transmit DMA can operate without generating host system processor interrupts. In this mode of operation, the host system does not set the TxIndicate or TxDMAIndicate bits in the TFC field of any TFDs used to transfer Ethernet frames from system memory. Thus, an interrupt is not issued by the Rubicon to indicate successful DMA transfer or successful transmission of each Ethernet frame. An interrupt will only be issued by the Rubicon in the event of a transmit error, but this case should be rare.

Without the use of interrupts, the Rio provides another mechanism for the host system to determine which Ethernet frames have been successfully transmitted. This mechanism allows the host system to free memory locations holding old TFD lists. This "interrupt-less" mechanism involves using the TxFrameId field of the TxStatus register. The TxFrameId field of the TxStatus register indicates the last Ethernet frame which was successfully transmitted. Using this information, the host system can infer successful transmission of all Ethernet frames up to the frame indicated by the TxFrameId field of the TxStatus register. Thus, the host system decides when to poll the TxFrameId field of the TxStatus register (for example, when the amount of

memory occupied by old TFD lists becomes excessive) and avoid generation of processor intensive interrupts by the Rio.

11.3.3 Receive DMA Interrupts

Interrupts can be generated by the Rio based on a number of events related to receive DMA operation:

- RxEarly interrupt is issued after receipt of Ethernet frame data from the Gigabit Ethernet network, where the number of bytes received is greater than the value of RxEarlyThresh register. Use of this interrupt is not recommended due to the frequency of Ethernet frame receipts in a Gigabit Ethernet network.
- RxDMAComplete interrupt is issued after successful transfer of one or more Ethernet frames (based on the interrupt coalescing configuration) from the Rio to the host system memory. Interrupt coalescing should be used in conjunction with the RxDMAComplete interrupt given the frequency of frame receipts in a Gigabit Ethernet network.
- RxDMAPriority interrupt is issued if a received Ethernet frame contains a Tag Control Information field with priority greater than or equal to the priority set in the RxDMAIntCtrl register.
- RFDListEnd interrupt is issued if the end of the RFD list is reached (indicated by an RFDNextPtr field with a value of 0x0000000000), or a RFD with the RFDDone bit of the RFS field with a value of logic 1 is encountered.

11.3.4 Receive DMA Interrupt Coalescing

A common use of interrupts during receive DMA operation is to indicate when new Ethernet frames have been transferred to host system memory. Interrupts however usually incur a significant cost in terms of host system performance, requiring a large percentage of processor time to service. One way to minimize the number of interrupts issued by the Rio related to receive DMA operation is to issue a single interrupt to indicate multiple Ethernet frames have been received. While minimizing interrupts can improve host system performance, it can also require more host system memory usage, and increase network latency. Therefore, a balance between interrupt frequency and network latency must be reached by the host system to optimize performance.

RxDMAComplete interrupt coalescing is configured via the RxDMAIntCtrl register. With interrupt coalescing, the trigger for issuing a RxDMAComplete interrupt is determined by:

- the maximum number of frames transferred via receive DMA (beginning with the previous assertion of RxDMAComplete) or,
- a maximum delay from receive DMA transfer completion of the first frame (beginning with the previous assertion of RxDMAComplete) to assertion of the interrupt;

whichever occurs first.

11.4 ACPI

The Rio supports operating system directed power management according to the ACPI specification. Power management registers in the PCI configuration space, as defined by the PCI Bus Power Management Interface specification, Revision 1.0 are described in section 11.10.

11.4.1 Power Management States

The Rio supports several power management states. The PowerState field in the PowerMgmtCtrl register determines Rio's current power state. The power states are defined as follows:

- D0 Uninitialized (power state 0) is entered as a result of hardware reset, or after a transition from D3 Hot to D0. This state is the same as D0 Active except that the PCI configuration registers are uninitialized. In this state, the Rio responds to PCI configuration cycles only.
- D0 Active (power state 0) is the normal operational power state for the Rio. In this state, the PCI configuration registers have been initialized by the system, including the IoSpace, MemorySpace, and

BusMaster bits in the ConfigCommand register, so the Rio is able to respond to PCI I/O, memory and configuration cycles and can operate as a PCI master. The Rio cannot signal wake (PMEN on the PCI bus) from the D0 state.

- D1 (power state 1) is a “light-sleep” state. The Rio optionally supports this state determined by the D1Support bit in the ConfigParm word in the EEPROM. The D1 state allows transition back to D0 with no delay. In this state, the Rio responds to PCI configuration accesses, to allow the system to change the power state. In D1 the Rio does not respond to any PCI I/O or memory accesses. The Rio’s function in the D1 state is to recognize wake events and link state events and pass them on to the system by asserting the PMEN signal on the PCI bus.
- D2 (power state 2) is a partial power-down state. The Rio optionally supports this state determined by the D2Support bit in the ConfigParm word in the EEPROM. D2 allows a faster transition back to D0 than is possible from the D3 state. In this state, the Rio responds to PCI configuration accesses, to allow the system to change the power state. In D2 the Rio does not respond to any PCI I/O or memory accesses. The Rio’s function in the D2 state is to recognize wake events and link state events and pass them on to the system by asserting the PMEN signal on the PCI bus.
- D3 Hot (power state 3) is the full power-down state for the Rio. In D3 Hot, the Rio loses all PCI configuration information except for the value in PowerState. In this state, the Rio responds to PCI configuration accesses, to allow the system to change the power state back to D0 Uninitialized. In D3 hot, the Rio does not respond to any PCI I/O or memory accesses. The Rio’s main responsibility in the D3 Hot state is to recognize wake events and link state events and signal those to the system by asserting the PMEN signal on the PCI bus.
- D3 Cold (power state undefined) is the power-off state for the Rio. The Rio does not function in this state. When power is restored, the system guarantees the assertion of hardware reset, which puts the Rio into the D0 Uninitialized state.

11.5 Wake On LAN

Wake on LAN is a key component of the IBM/Intel® Advanced Manageability Alliance (AMA) initiative. The Rio implements a portion of the Wake On LAN functionality defined by the AMA intiative. Specifically, the Rio can be configured to respond to wake up frames sent by a Wake On LAN management station.

11.5.1 Wake Events

The Rio can generate wake events to the system as a result of Wake Packet reception, Magic Packet reception, or due to a change in the link status. The WakeEvent register gives the host system control over which of these events are passed to the system. Wake events are signaled over the PCI bus using the PMEN signal.

A Wake Packet event is controlled by the WakePktEnable bit in WakeEvent register. The WakePktEnable bit has no effect when Rio is in the D0 power state, as the wake process can only take place in states D1, D2, or D3. When the Rio detects a Wake Packet, it signals a wake event on WAKE (if WAKE assertion is enabled), and sets the WakePktEvent bit in the WakeEvent register. The Rio can signal that a wake event has occurred when it receives a pre-defined frame from another station. The host system transfers a set of frame data patterns into the transmit FIFO using the TxDMA function before placing the Rio in a power-down state. Once powered down, the Rio compares receive frames with the frame patterns in the transmit FIFO. When a matching frame is received (and also passes the filtering mode set in the ReceiveMode register), a wake event is signaled.

Frame patterns are written to the transmit FIFO in a single “pseudo-packet”. Prior to transferring this pseudo-packet, the host system should first set the TxReset in the AsicCtrl (to reset the transmit FIFO pointers and prevent transmission) then prepare a TFD that points to a single data buffer. The buffer should contain one or more frame patterns placed contiguously. The number of frame patterns is limited by the transmit FIFO size. The FragLen field in the TFD must exactly equal the sum of the frame pattern bytes. Also, the host system must set the WordAlign field to ‘x1’ in the TFC field of the TFD to prevent

frame word-alignment. Finally, the host system must write the TFD's address to the TFDListPtr register to transfer the frame into the transmit FIFO.

The frame patterns in the transmit FIFO specify which bytes in the incoming frames are to be examined. A CRC is calculated over these bytes and compared with a CRC value supplied in the frame pattern. This matching technique may result in false wake events being reported to the host system. Each wake packet pattern contains one or more byte-offset/byte-count pairs, an end-of-pattern symbol, and a 4-byte CRC value. The byte-offset indicates the number of frame bytes to be skipped in order to reach the next group of bytes to be included in the CRC calculation. The byte-count indicates the number of bytes in the next group to be included in the CRC calculation. End-of pattern, which is a byte value of 0x00, indicates the end of the pattern for that wake frame. Immediately following the end-of-pattern is a 4-byte CRC. The CRC calculation uses the same polynomial as the Ethernet MAC FCS. The pseudo packet frame patterns are described in section 11.10.

An example pseudo-packet (based on the ARP packet example from Appendix A of the "OnNow Network Device Class Power Management Specification") loaded into the transmit FIFO of the Rio is shown in Figure 1.

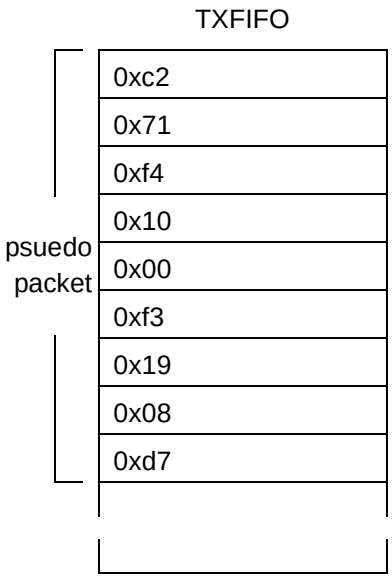


FIGURE 1: Example Psuedo Packet

Using the pseudo packet in Figure 1, the Rio will assert a wake event if a packet of the form shown in Figure 2 is received whereby a 32-bit CRC over the indicated bytes of the received packet yields the value 0xF31908D7.

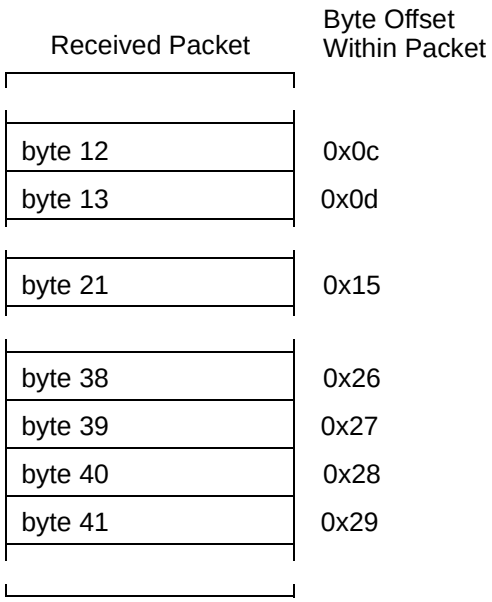


FIGURE 2: Example Wake Packet

The Rio also supports Magic Packet™ technology developed by Advanced Micro Devices to allow remote wake-up of a sleeping station on a network via transmission of a special frame. Once the Rio has been placed in Magic Packet mode and put to sleep, it scans all incoming frames addressed to it for a data sequence consisting of 16 consecutive repetitions of its own 48-bit Ethernet MAC StationAddress. This sequence can be located anywhere within the frame, but must be preceded by a synchronization stream. The synchronization stream is defined as 6 bytes of 0xFF. For example, if the MAC address programmed

into the StationAddress register is 0x11:22:33:44:55:66, then the Rio would be scanning for the frame data shown in Figure 3.

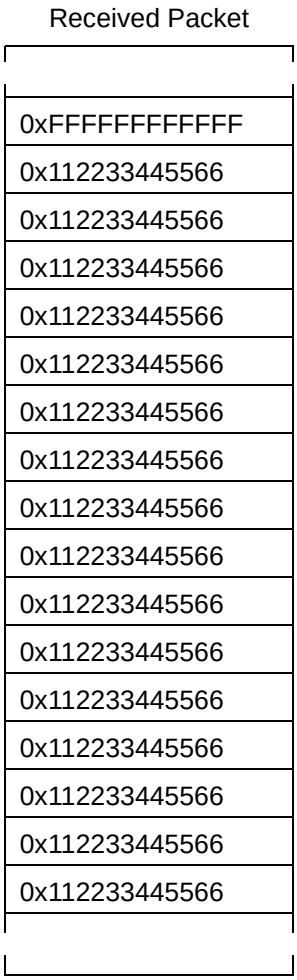


FIGURE 3: Example Magic Packet

Magic Packet wake up is controlled by the MagicPktEnable bit in the WakeEvent register. A wake event can only take place in the D1, D2, or D3 states, and the MagicPktEnable bit has no effect when the Rio is in the D0 power state. The Magic Packet must also pass the address matching criteria set in ReceiveMode register. A Magic Packet may also be a broadcast frame. When the Rio detects a Magic Packet, it signals a wake event on WAKE (if WAKE assertion is enabled), and sets the MagicPktEvent bit in the WakeEvent register.

The Rio can also signal a wake event when it senses a change in the network link state, from “link up” to “link fail”, or vice versa. Link state wake is controlled by the LinkEventEnable bit in the WakeEvent register. At the time LinkEventEnable bit is set by the host system, the Rio samples the current link state. It then waits for the link state to change. If the link state changes before the Rio returns to state D0 or the LinkEventEnable bit is cleared, the LinkEvent bit is set in the WakeEvent register, and (if it is enabled) the WAKE signal is asserted.

11.6 FIFO

The Rio incorporates on chip transmit and receive FIFOs. The transmit FIFO is 16kbytes (16,384 bytes) in length while the receive FIFO is 32kbytes (32,768 bytes) in length.

11.7 MAC

The MAC block implements the IEEE Ethernet 802.3 Media Access Control functions with Full Duplex and Flow Control enhancements. In half duplex mode, the MAC implements the Carrier Sense Multiple Access with Collision Detect (CSMA/CD) protocol. Full duplex mode by definition does not utilize CSMA/CD, allowing data to be transmitted on demand. An optional flow control mechanism in full duplex mode is provided via the MAC Control PAUSE function. Additionally, the MAC also performs these functions in either half or full duplex mode:

- Optional transmit frame check sequence (FCS) generation
- Padding to the minimum legal frame size
- Preamble and SFD generation
- Preamble and SFD removal
- Receive frame FCS checking and optional FCS stripping
- Receive frame destination address matching
- Support for multicast and broadcast frame reception or rejection (via filtering)

In addition, the MAC is responsible for generation of hardware signals to update the internal statistics counters.

11.7.1 VLAN

Virtual Local Area Network (VLAN) technology is used to regulate broadcast and multicast traffic in switched Ethernet networks. VLAN technology utilizes Ethernet frame tagging, providing Ethernet switches a mechanism to correlate a specific Ethernet frame with a specific group of end stations. Using this correlation, Ethernet switches in a network are able to regulate broadcast and multicast VLAN tagged frames, forwarding such frames only to those nodes which are members of the same VLAN (instead of to all nodes). In this way, broadcast and multicast network utilization is minimized.

The IEEE defines VLANs as follows:

- VLANs facilitate easy administration of logical groups of stations that can communicate as if they were on the same LAN. They also facilitate easier administration of moves, adds and changes in members of these groups.
- Traffic between VLANs is restricted. Bridges forward unicast, multicast and broadcast traffic only on LAN segments that serve the VLAN to which the traffic belongs.
- As far as possible, VLANs maintain compatibility with existing bridges and end-stations.

Detailed information on VLAN implementation is located in the following standards:

- IEEE 802.1p Traffic Class Expediting and Dynamic Multicast Filtering (now part of ISO/IEC 15802-3: 1998). Introduces the concept of priority processing in Ethernet bridges and specifies the registration protocols used to disseminate switching rules between nodes and switches in a network. Also defines GMRP and GARP.
- IEEE 802.1Q Virtual Bridged Local Area Networks (also now part of ISO/IEC 15802-3: 1998). Specifies the operation of VLAN enabled Ethernet bridges, and defines the tagged frame format.
- IEEE 802.3ac Frame Extensions for Virtual Bridged Local Area Networks (VLAN) Tagging on 802.3 Networks. Modifies the IEEE 802.3 specification to accommodate tagging for VLANs as specified in IEEE 802.1Q.

The Rio supports VLANs with the following functions:

- Transmission and reception of VLAN tagged frames, increasing the maximum frame size by four octets.

- VLAN tags for transmit frames may be applied either by the host system prior to transfer of the frame to the Rio via the transmit DMA process, or by the Rio via the VLAN tag information specified in the TFC or the VLANTag register. The the TFC VLANTagInsert field, and MACCtrl register AutoVLANtagging bit determines the source for VLAN frame tagging with the TFC VLANTagInsert having priority over the MACCtrl register AutoVLANtagging bit.
- Any VLAN tagged frames received by the Rio may be transferred to the host system unmodified, or stripped of all VLAN tags as determined by the MACCtrl register AutoVLANuntagging bit. For any received frame which contains a VLAN tag, regardless of the state of the MACCtrl register AutoVLANuntagging bit, the VLAN tag is copied to the RFS TCI field.
- The priority of VLAN tagged frames received by the Rio may be detected and based on the programmable PriorityThresh field of the RxDMAIntCtrl register, an interrupt asserted via the RxDMAPriority field of the IntStatus or IntStatusAck register.
- Using the ReceiveVLANMatch or ReceiveVLANHash fields of the ReceiveMode register, only VLAN tagged frames with specified VLAN ID values are passed to the host system. All other VLAN tagged frames and all un-tagged frames are dropped.

11.7.2 Layer 3/4 Checksums

The Ethernet Carrier Sense Multiple Access with Collision Detect (CSMA/CD) protocol comprises a portion of Layer 2 within the Open Systems Interconnect (OSI) Seven Layer Model of network systems. Ethernet incorporates a CRC capability (via the FCS field) in an attempt to check for errors during transmission. Higher layer protocols which utilize Ethernet may also utilize checksums in addition to the Ethernet FCS. These higher layer protocol checksum are typically calculated by the host system, and inserted within the Ethernet frame (for transmit data) prior to frame transfer to the Rio via the transmit DMA process. Similarly, higher layer protocol checksums within received Ethernet frames are verified by the host system after the frames have been transferred from the Rio via receive DMA process.

The Rio can perform checksum calculations, and verifications for three popular higher layer protocols.

- Internet Protocol version 4 (Layer 3 within the OSI model) defined in RFC 791
- Transmission Control Protocol (Layer 4 within the OSI model) defined in RFC 793
- User Datagram Protocol (Layer 4 within the OSI model) defined in RFC 768

By configuring the Rio to perform the checksum calculations for the supported protocols, the host system work load is lightened resulting in higher performance.

Utilization of the Rio layer 3/4 checksum calculation capability for transmit data depends on the make up of the Ethernet frame.

- If the frame contains an IP datagram, write a logic 1 to the IPChecksumEnable bit within the TFC field of the frame's TFD.
- If the frame contains a TCP segment within an IP datagram, write a logic 1 to the TCPChecksumEnable bit and a logic 1 to the IPChecksumEnable bit within the TFC field of the frame's TFD.
- If the frame contains a UDP segment within an IP datagram, write a logic 1 to the UDPChecksumEnable bit bit and a logic 1 to the IPChecksumEnable bit within the TFC field of the frame's TFD.
- If any combination of the IPChecksumEnable, TCPChecksumEnable, and/or UDPChecksumEnable bits within the TFC field of the frame's TFD are a logic 1, the FCSAppendDisable bit within the TFC field must be a logic 0.
- Use of the layer 3/4 checksum functions affects operation of the transmit process with respect to the TxStartThresh register (see the TxStartThresh definition for details).

It is the host system's responsibility to assure that the respective checksum functions are only enabled for Ethernet frames which contain the respective layer 3/4 data. That is, if an Ethernet frame does not contain an IP datagram, the IPChecksumEnable bit within the TFC field of the frame's TFD must be a logic 0.

Utilization of the Rio layer 3/4 checksum verification capability for receive data does not require any register configuration. The checksum verification capabilities are always active. The host system decides on a frame by frame basis how to utilize the layer 3/4 checksum verification functions.

- For the last RFD of a received Ethernet frame, the RFS field IPDetected, TCPDetected, and UDPDetected bits will be a logic 1 to indicate if the received frame contains an IP datagram, TCP segment within an IP datagram, and/or UDP segment within an IP datagram.
- If the IPDetected, TCPDetected, and/or UDPDetected bits within the RFS field of the received frame are a logic 1, then the host system may examine the IPError, TCPError, and UDPError bits within the RFS field to verify the respective layer 3/4 checksums for the received frame. Separate host system algorithms to verify the checksums are not necessary, but may still be performed (the layer 3/4 checksums are part of the frame data transferred by the Rio via the receive DMA process).
- The IPCheckSumErrors, TCPCheckSumErrors, and UDPCheckSumErrors statistic registers will count the number of received frame's which contain an IP datagram, TCP segment within an IP datagram, and/or UDP segment within an IP datagram where the respective layer 3/4 checksum calculation within the Rio failed.

11.7.3 Flow Control

The Rio supports both asymmetric and symmetric IEEE 802.3 flow control via the MAC Control PAUSE function. Any IEEE 802.3 flow control compliant node receiving a PAUSE control frame must inhibit frame transmission for the amount of time specified in the PAUSE control frame. The pause time is specified in pause quanta (in Gigabit Ethernet, a pause quanta is 512 bit times and a bit time is 1ns). The maximum pause time is 65,535 pause quanta, or 33.6ms.

Asymmetric operation corresponds to the Rio acting on PAUSE frames received from a Gigabit Ethernet network. Symmetric operation corresponds to the Rio both acting on received PAUSE frames, and transmitting PAUSE frames onto a Gigabit Ethernet network. Use of asymmetric and symmetric flow control is typically determined during auto negotiation (see section 11.9.1).

When participating in symmetric flow control operation, transmit PAUSE control frames can be generated by the host system, or automatically by the Rio. The host system may use any mechanism to determine when to transfer a PAUSE control frame to the Rio. Automatic generation of PAUSE control frames by the Rio is related to the state of the receive FIFO. If the receive FIFO fills beyond a host system configurable point (the flow control on threshold, as defined by the FlowOnThresh register), the Rio will automatically transmit a PAUSE control frame in an attempt to halt the transmitting node. The flow control on threshold, above which the Rio sends a PAUSE control frame, must be chosen carefully to account for receiving frames already in transit. A general rule is to set the flow control on threshold offset (the difference between the maximum size of the FIFO, and the flow control on threshold) equal to or greater than twice the size (in bytes) of the maximum expected receive frame size.

For the Rio automatic flow control mechanism to function properly, the receive FIFO must empty beyond a second host system configurable point (the flow control off threshold, as defined by the FlowOffThresh register) within 33.6ms (the amount of time the opposite node will remain paused when receiving the Rio generated PAUSE frame). When the Rio receive FIFO empties beyond the flow control off threshold a second PAUSE frame (with pause time equal to zero) is transmitted. This zero time PAUSE frame indicates the Rio is ready to receive more data. The Rio will not transmit a maximum time PAUSE frame (even if the flow control on threshold is crossed again) until the receive FIFO flow control off threshold is crossed. If the Rio receive FIFO flow control off threshold is not crossed within 33.6ms after the maximum time PAUSE frame is transmitted, the Rio receive FIFO will likely overrun (see Figure 4). This condition indicates a problem with the host system, in that data is not being transferred from the Rio receive FIFO at a sufficient rate to keep up with the Gigabit Ethernet LAN segment.

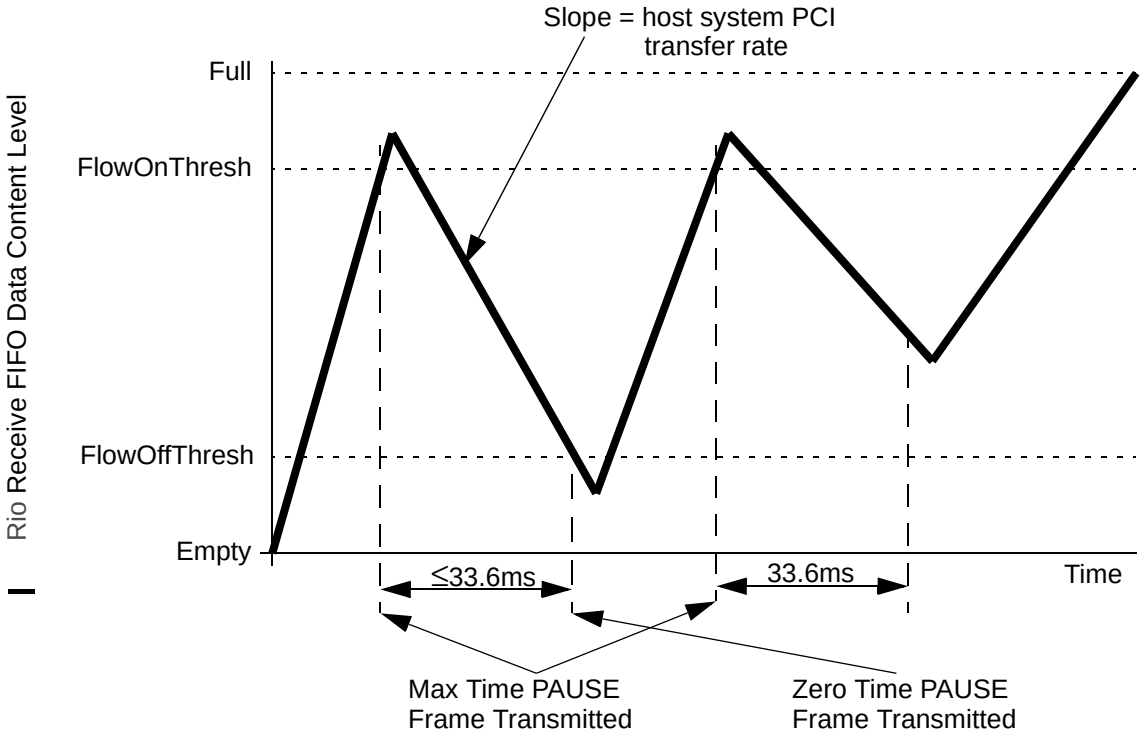


FIGURE 4: Flow On and Flow Off Threshold Considerations

The flow control thresholds must be chosen carefully to account for the time required by the host system to empty the receive FIFO. A general rule is to set the flow control thresholds using the minimum expected transfer rate of the host system PCI bus (based on bus width, speed, and availability) according to the following equation:

$$\frac{FlowOnThresh_{bytes} - FlowOffThresh_{bytes}}{33.6ms} \leq MinHostSystemPCITransferRate_{Bps}$$

Receive flow control operation of the Rio is controlled via the MACCtrl register RxFlowControlEnable bit. Transmit flow control operation of the Rio (automatic PAUSE frame generation) is controlled via the MACCtrl register TxFlowControlEnable bit and the FlowOnThresh and FlowOffThresh registers.

11.8 GMII

The Rio can support a variety of physical signaling schemes via the IEEE 802.3 defined Gigabit Media Independent Interface (GMII). Through the GMII, the Rio supports various physical layer implementations of the Gigabit Ethernet standard. The GMII provides a general-purpose interface between an 802.3 MAC and various physical layer devices, and is comprised of two independent components. The data interface provides separate, 8-bit wide paths for receive and transmit data, as well as independent clock and control signals. The management interface is a bidirectional, serial link that provides the Rio access to registers residing within the physical layer device. The host system controls the GMII management interface through the PhyCtrl register.

Since the GMII is independent of the signaling method (1000BASE-X, 1000BASE-T), it is possible to use it to support numerous Ethernet LAN types depending upon the availability of GMII-compliant PHY devices.

It is most likely that a physical layer device connected to Rio's GMII will include implementation of the 802.3 Auto-Negotiation function. For instance, a PHY device may be able to auto-negotiate various parameters of Gigabit Ethernet operation. A host system attempting to determine link status should check the Auto-Negotiation function contained in the GMII-based PHY device through the GMII management interface of the Rio.

11.9 PCS

The Rio implements an IEEE 802.3 compliant Physical Coding Sublayer (PCS) for use with PHY devices using the TBI. The Rio PCS implements the following functions:

- 8B10B Encoder
- Transmit State Machines
- Auto-negotiation State Machine
- 10B8B Decoder
- Receive State Machine
- Synchronization State Machine
- Management Registers

11.9.1 Auto-Negotiation

The Rio PCS layer implements IEEE 802.3 1000BASE-X auto-negotiation for Gigabit Ethernet only. The auto-negotiation function is controlled via the PCS registers internal to the Rio PCS layer implementation. The host system must utilize these registers in order to properly configure the Rio for compatible operation with the Rio's link partner. The host system responsibilities are as follows:

- Enable or disable the Rio for auto-negotiation via the Auto-Negotiation Enable bit of the PCS Control register.
- Configure the PCS Advertisement register to indicate the capabilities of the Gigabit Ethernet implementation.
- If there is any Next Page data to send, and/or if the host system is willing to accept Next Page data, configure the Next Page bit of the Advertisement register.
- Re-start auto-negotiation using the Control register.
- Read the Page Received bit of the Expansion register to determine when the link partner's base page data has been received. This is only necessary if Next Page transmissions are to follow. If there are to be no Next Page transmissions, the Control register Auto-Negotiation Complete bit can be used to determine when to read the link partner's base page data.
- Read the LinkPartnerBasePage register to determine the link partner's capabilities.
- If the LinkPartnerBasePage register Next Page bit is 1 (indicating the link partner is able to accept and/or would like to transmit Next Page information) transmit the first page of the Next Page information to the link partner via a write to the NextPage register. If there is no Next Page data to transfer, write 0x2001 to the NextPage register.
- After writing to the NextPage reigster, read the Expansion register, Page Received bit to determine when new Next Page data from the link partner has arrived.
- When new Next Page data has arrived from the link parnter, read the LinkPartnerNextPage register.
- For multiple Next Page data transfers, the host system must repeat the process of writing to the NextPage register, reading the Expansion register, and reading the LinkPartnerNextPage register until a logic 0 is read in the LinkPartnerNextPage register Next Page bit is read or a value of 0x2001 for the Next Page data is read and there are no more Next Page transfers required.

- Optionally, the NextPage register Acknowledge 2 bit may be used to indicate to the link partner that received Next Page data will be acted upon. A logic 1 in the Acknowledge 2 bit of the LinkPartnerNextPage register indicates the link partner is able to perform the task defined in the Next Page message.
- The host system must implement the Priority Resolution function (specified in IEEE 802.3 1998 Edition Clause 37.2.4.2) to resolve the mode of operation for the Rio. Priority Resolution applies to the duplex, and pause modes of operation. The rules for resolution are contained in the IEEE standard, and duplicated here for reference.

Table 11: Duplex Mode Priority Resolution

LOCAL DEVICE ADVERTISED DUPLEX MODE	LINK PARTNER ADVERTISED DUPLEX MODE	LOCAL DEVICE DUPLEX MODE RESOLUTION
Half	Half	Half
Half	Full	Half
Full	Half	Half
Full	Full	Full

Table 12: PAUSE Mode Priority Resolution

LOCAL DEVICE ADVERTISED PAUSE MODE		LINK PARTNER ADVERTISED PAUSE MODE		LOCAL DEVICE PAUSE MODE RESOLUTION
PAUSE	ASM_DIR	PAUSE	ASM_DIR	
0	0	-	-	Disable PAUSE Transmit and Receive
0	1	0	-	Disable PAUSE Transmit and Receive
0	1	1	0	Disable PAUSE Transmit and Receive
0	1	1	1	Enable PAUSE Transmit, Disable PAUSE Receive
1	0	0	-	Disable PAUSE Transmit and Receive
1	0	1	-	Enable PAUSE Transmit and Receive
1	1	0	0	Disable PAUSE Transmit and Receive
1	1	0	1	Enable PAUSE Receive, Disable PAUSE Transmit

Table 12: PAUSE Mode Priority Resolution

LOCAL DEVICE ADVERTISED PAUSE MODE		LINK PARTNER ADVERTISED PAUSE MODE		LOCAL DEVICE PAUSE MODE RESOLUTION
PAUSE	ASM_DIR	PAUSE	ASM_DIR	
1	1	1	-	Enable PAUSE Transmit and Receive

- After resolving the modes of operation, the host system must configure the Rio MACCtrl register DuplexSelect bit, RxFlowControlEnable bit (to configure PAUSE MAC Control Frame receive functionality), and the Duplex Mode bit of the PCS Control register (used only for properly setting the EA0/LEDDPLXN LED signal). PAUSE Mac Control Frame transmit functionality is controlled wholly by the host system.
- Finally, the host system must configure functionality negotiated via the Next Page data transfers.

11.10 Registers and Data Structures

The Rio utilizes both on chip registers, as well as data structures located within the host system memory, and external non-volatile memory devices.

11.10.1 Transmit DMA Data Structure

The host system uses the Transmit Frame Descriptor (TFD) to transfer data from host system memory to Rubicon via the transmit DMA process. TFDs must be located within host system memory on quad word (64 bit) boundaries. The TFD format is shown in Table 13.

TABLE 13: Rubicon TFD Format

64 BITS	ADDR OFFSET
TFDNextPtr	0x00
TFC	0x08
FragInfo0	0x10
FragInfo1	0x18
FragInfo2	0x20
FragInfo3	0x28
FragInfo4	0x30
FragInfo5	0x38
FragInfo6	0x40
FragInfo7	0x48
FragInfo8	0x50

TABLE 13: Rubicon TFD Format

64 BITS	ADDR OFFSET
FragInfo9	0x58
FragInfo10	0x60
FragInfo11	0x68
FragInfo12	0x70
FragInfo13	0x78
FragInfo14	0x80

11.10.1.1 TFDNextPtr

Class..... Transmit DMA Data Structure
Base Address ..... TFD starting address in system memory
Address Offset..... 0x00
Width ..... 64 bits
Transmit Frame Descriptor Next Pointer field.

BIT	BIT NAME	BIT DESCRIPTION
39..0	TFDNextPtr	TFD Next Pointer indicates the location in system memory of the next TFD. Only the lower 40 bits of the TFDNextPtr field are valid, the upper 24 bits are reserved. A value of 0x0000000000 in this field indicates there are no more TFDs ready for transfer.
63..40	Reserved	Reserved for future use. Write as zero, ignore on read.

11.10.1.2 TFC

Class..... Transmit DMA Data Structure
Base Address ..... TFD starting address in system memory
Address Offset..... 0x08
Width ..... 64 bits
Transmit Frame Control field.

<table>
<tr><th>BIT</th><th>BIT NAME</th><th>BIT DESCRIPTION</th></tr>
<tr><td>15..0</td><td>FrameId</td><td>Frame Identification associates the data referenced by this TFD to a specific Ethernet frame.</td></tr>
<tr><td>17..16</td><td>WordAlign</td><td>Word Alignment bits determine the boundary to which transmit frame lengths are rounded up in the transmit FIFO, and transmitted onto the network medium.
<table>
<tr><th>BIT 17</th><th>BIT 16</th><th>ALIGNMENT</th></tr>
<tr><td>0</td><td>0</td><td>Align to Double Word</td></tr>
<tr><td>1</td><td>0</td><td>Align to Word</td></tr>
<tr><td>x</td><td>1</td><td>Alignment Disabled</td></tr>
</table>
When using word alignment, it is the responsibility of the host system to recognize that any added bytes necessary to achieve the desired alignment may affect byte oriented functions and fields (i.e. if the Ethernet Length/Type field holds a frame length, this value is not updated to reflect any bytes added via word alignment).</td></tr>
<tr><td>18</td><td>TCPChecksumEnable</td><td>Transmission Control Protocol Checksum Enable indicates the host system, or the Rio will calculate and insert the TCP checksum as defined in RFC 793. A logic 1 indicates the Rio should calculate the TCP checksum. TCPChecksumEnable should only be set if the Ethernet frame referenced by this TFD contains TCP data within an IP datagram. For TFD’s which do not contain TCP data, TCPChecksumEnable MUST be 0. In addition, if TCPChecksumEnable is set, IPChecksumEnable MUST be a logic 1 and FCSAppendDisable MUST be 0. The Rio must calculate and append the FCS when performing TCP checksum calculation and insertion.</td></tr>
<tr><td>19</td><td>UDPChecksumEnable</td><td>User Datagram Protocol Checksum Enable indicates the host system, or the Rio will calculate and insert the UDP checksum as defined in RFC 768. A logic 1 indicates the Rio should calculate the UDP checksum. UDPChecksumEnable should only be set if the Ethernet frame referenced by this TFD contains UDP data within an IP datagram. For TFD’s which do not contain UDP data, UDPChecksumEnable MUST be 0. In addition, if UDPChecksumEnable is set, IPChecksumEnable MUST be a logic 1 and FCSAppendDisable MUST be 0. The Rio must calculate and append the FCS when performing UDP checksum calculation and insertion.</td></tr>
</table>

BIT	BIT NAME	BIT DESCRIPTION
20	IPChecksumEnable	Internet Protocol Checksum Enable indicates the host system, or the Rio will calculate and insert the IP checksum as defined in RFC 791. A logic 1 indicates the Rio should calculate the IP checksum. IPChecksumEnable should only be set if the Ethernet frame referenced by this TFD contains IP data. For TFD's which do not contain IP data, IPChecksumEnable MUST be 0. In addition, if IPChecksumEnable is set, FCSAppendDisable MUST be 0. The Rio must calculate and append the FCS when performing IP checksum calculation and insertion.
21	FCSAppendDisable	Frame Check Sequence Disable is set by the host system to prevent the Rio from appending the 4-byte Frame Check Sequence (FCS) to the end of the current frame. In this case, the host system must supply the frame's FCS as part of the data transferred by the transmit DMA process to the transmit FIFO. An exception exists when a transmit underrun occurs. In this case a guaranteed invalid FCS will be appended to the frame by the Rio. When FCSAppendDisable is cleared, the Rio will compute and append the FCS to the transmit frame referenced by this TFD. When using TCPChecksumEnable, UDPChecksumEnable, and/or IPChecksumEnable, the FCSAppendDisable bit MUST be set to logic 0 and the host system MUST NOT append an FCS to the Ethernet frame.
22	TxIndicate	Transmit Indicate is set by the host system to request a TxComplete interrupt upon completion of frame transmission by the MAC. If TxIndicate is cleared, no interrupt of transmission completion will be issued by the Rio, unless a transmit error occurs.
23	TxDMAIndicate	Transmit DMA Indicate is set if the host system desires a TxDMAComplete interrupt upon completion of the transmit DMA process for this frame.
27..24	FragCount	Fragment Count indicates the number of fragments (from 1 to 15 fragments) used by this TFD. A value of 0 indicates this TFD contains no data to be transferred to the Rio. The Rio will write a logic 0 to the FragCount bits upon completion of the transmit DMA process for this TFD (as part of the TFDDone bit update process).
28	VLANTagInsert	VLAN Tag Insert indicates that the VLAN information specified in VID, CFI, and UserPriority fields of the TFC should be inserted into the Ethernet frame specified by this TFD. In addition to the VID, CFI, and UserPriority fields, the Length/Type field of the Ethernet frame specified by this TFD will be set to the value 0x8100 to indicate the frame is VLAN tagged. The Rio will write a logic 0 to the VLANTagInsert bit upon completion of the transmit DMA process for this TFD (as part of the TFDDone bit update process).

BIT	BIT NAME	BIT DESCRIPTION
30..29	Reserved	Reserved for future use. Write as zero, ignore on read. The Rio will write a logic 0 to the Reserved bits upon completion of the transmit DMA process for this TFD (as part of the TFDDone bit update process).
31	TFDDone	Transmit Frame Descriptor Done is used as an ownership bit to indicate whether the host system, or the Rio “owns” (or is currently processing and/or modifying) the TFD. The Rio will not transfer the TFD from host system memory until a logic 0 is written to the TFDDone bit by the host system. The Rio will write a logic 1 to the TFDDone bit upon completion of the transmit DMA process for this TFD.
43..32	VID	VLAN Identifier specifies the VID portion of the Tag Control Information (TCI) field within an Ethernet frame VLAN tag (see section 11.7.1). The VID is only valid if the VLANTagInsert bit is a logic 1.
44	CFI	Canonical Format Indicator specifies the CFI portion of the Tag Control Information (TCI) field within an Ethernet frame VLAN tag (see section 11.7.1). The CFI is only valid if the VLANTagInsert bit is a logic 1.
47..45	UserPriority	User Priority specifies the UserPriority portion of the Tag Control Information (TCI) field within an Ethernet frame VLAN tag (see section 11.7.1). The UserPriority is only valid if the VLANTagInsert bit is a logic 1.
63..48	Reserved	Reserved for future use. Write as zero, ignore on read.

11.10.1.3 FragInfo

Class..... Transmit DMA Data Structure

Base Address ..... TFD starting address in system memory

Address Offset..... 0x10 + n * 0x8 (n = fragment number = 0x0, 0x1, 0x2... 0xE)

Width ..... 64 bits

Fragment Information fields (up 15 in a single TFD) indicate a block of contiguous system memory containing a portion of the frame data to be transmitted. The fragments may be located anywhere in system memory, and need not be contiguous.

BIT	BIT NAME	BIT DESCRIPTION
39..0	FragAddr	Fragment Address indicates the location within host system memory of the first byte of the data fragment. Fragments can be located anywhere within a 40 bit (1,099,511,627,776 or 1 Terabyte) memory space.
47..40	Reserved	Reserved for future use. Write as zero, ignore on read.
63..48	FragLen	Fragment Length indicates the length of the fragment in bytes. Fragments can range from zero to 64k (65,536) bytes in length.

11.10.2 Receive DMA Data Structure

The host system uses the Receive Frame Descriptor (RFD) to transfer data from the Rio to host system memory via the receive DMA process. RFDs must be located within host system memory on quad word (64 bit) boundaries. The RFD format is shown in Table 14.

TABLE 14: Rubicon RFD Format

64 BITS	ADDR OFFSET
RFDNextPtr	0x00
RFS	0x08
FragInfo0	0x10

11.10.2.1 RFDNextPtr

Class..... Receive DMA Data Structure
Base Address ..... RFD starting address in system memory
Address Offset..... 0x00
Width ..... 64 bits
Receive Frame Descriptor Next Pointer field.

BIT	BIT NAME	BIT DESCRIPTION
39..0	RFDNextPtr	RFD Next Pointer indicates the location in system memory of the next RFD. A value of 0x0000000000 in this field indicates there are no more RFDs ready for data.
63..40	Reserved	Reserved for future use. Write as zero, ignore on read.

11.10.2.2 RFS

Class..... Receive DMA Data Structure
Base Address ..... RFD starting address in system memory
Address Offset..... 0x08
Access Mode ..... Read/Write
Width ..... 64 bits
Receive Frame Status field.

BIT	BIT NAME	BIT DESCRIPTION
15..0	RxFrameLen	Receive Frame Length indicates the length in bytes of the received Ethernet frame. If the RFD, FragInfo field's FragLen field is less than the Ethernet frame length, multiple RFDs are required to store the frame. In this case, the RxFrameLen field of the RFS in the last RFD will indicate the actual Ethernet frame length, while all other RxFrameLen fields in the series of RFDs used to store the frame will be 0x0000.
16	RxFIFOOverrun	Receive FIFO Over Run indicates the received frame incurred a FIFO over run error. RxFIFOOverrun is undefined (and should be ignored) unless the RFS FrameEnd field is a logic 1.
17	RxRuntFrame	Receive Runt Frame indicates the received frame was a runt (a frame which is less than 60 octets in length, measured from the DA field to the end of the Data field). RxRuntFrame is undefined (and should be ignored) unless the RFS FrameEnd field is a logic 1.
18	RxAlignmentError	Receive Alignment Error indicates the received frame incurred an alignment error (alignment errors are not encountered in Gigabit Ethernet). RxAlignmentError is undefined (and should be ignored) unless the RFS FrameEnd field is a logic 1.
19	RxFCSError	Receive Frame Check Sequence Error indicates the received frame incurred a FCS error (the frames FCS field did not match the FCS calculation performed over the frame DA through Data fields). RxFCSError is undefined (and should be ignored) unless the RFS FrameEnd field is a logic 1.
20	RxOversizedFrame	Receive Oversized Frame indicates the received frame was larger than the value set in the MaxFrameSize register. The entire frame is written into the RFD regardless of this error. RxOversizedFrame is undefined (and should be ignored) unless the RFS FrameEnd field is a logic 1.
21	RxLengthError	Receive Length Error indicates the received frame length differed from the length field in the Ethernet frame header. If the received frame is a runt, RxLengthError is a logic 0. RxLengthError is undefined (and should be ignored) unless the RFS FrameEnd field is a logic 1.
22	VLANDetected	VLAN Detected indicates the received Ethernet frame contains a VLAN tag. VLANDetected is undefined (and should be ignored) unless the RFS FrameEnd field is a logic 1.

BIT	BIT NAME	BIT DESCRIPTION
23	TCPDetected	Transmission Control Protocol Detected indicates the received Ethernet frame contains TCP data within an Internet Protocol datagram. TCPDetected is undefined (and should be ignored) unless the RFS FrameEnd field is a logic 1.
24	TCPError	Transmission Control Protocol Error indicates the received Ethernet frame contains TCP data within an Internet Protocol datagram, and the TCP checksum calculation by Rio did not match the checksum value in the TCP header. TCPError is undefined (and should be ignored) unless the RFS FrameEnd field is a logic 1.
25	UDPDetected	User Datagram Protocol Detected indicates the received Ethernet frame contains UDP data within an Internet Protocol datagram. UDPDetected is undefined (and should be ignored) unless the RFS FrameEnd field is a logic 1.
26	UDPError	User Datagram Protocol Error indicates the received Ethernet frame contains UDP data within an Internet Protocol datagram, and the UDP checksum calculation by Rio did not match the checksum value in the UDP header. UDPError is undefined (and should be ignored) unless the RFS FrameEnd field is a logic 1.
27	IPDetected	Internet Protocol Detected indicates the received Ethernet frame contains an IP datagram. IPDetected is undefined (and should be ignored) unless the RFS FrameEnd field is a logic 1.
28	IPError	Internet Protocol Error indicates the received Ethernet frame contains an IP datagram, and the IP checksum calculation by Rio did not match the checksum value in the IP header. IPError is undefined (and should be ignored) unless the RFS FrameEnd field is a logic 1.
29	FrameStart	Frame Start indicates the first RFD used to describe a received Ethernet frame. In conjunction with the FrameEnd bit, these bits indicate the beginning and end of a single Ethernet frame within the RFD list. If both the FrameStart and FrameEnd bits in the RFS field are a logic 1, the frame required only a single RFD.
30	FrameEnd	Frame End indicates the last RFD used to describe a received Ethernet frame. In conjunction with the FrameStart bit, these bits indicate the beginning and end of a single Ethernet frame within the RFD list. If both the FrameStart and FrameEnd bits in the RFS field are a logic 1, the frame required only a single RFD.

BIT	BIT NAME	BIT DESCRIPTION
31	RFDDone	RFD Done is written as a logic 1 by the Rio after transferring received data to host system memory. If the Rio is directed to use (typically re-use) an RFD whose RFDDone bit is a logic 1, the Rio will not transfer data to the location within host system memory indicated by the RFD (and thus will pause receive DMA operation) until the RFDDone bit is a logic 0. The RFDDone bit is analogous to an ownership bit, when a logic 1 indicates the host system owns the RFD and has not yet processed the corresponding data. When a logic 0, RFDDone indicates the Rio owns the RFD and may overwrite any data referenced by the RFD.
47..32	TCI	Tag Control Information contains the TCI portion of the VLAN tag found in the received frame described by this RFD (if a VLAN tag was found) regardless of the state of the MACCtrl register AutoVLANuntagging bit. TCI is only valid if the VLANDetected bit is a logic 1.
63..48	Reserved	Reserved for future use. Write as zero, ignore on read.

11.10.2.3 FragInfo

Class..... Receive DMA Data Structure

Base Address ..... RFD starting address in system memory

Address Offset..... 0x10

Access Mode ..... Read/Write

Width ..... 64 bits

Fragment Information indicates a block of contiguous system memory reserved for a portion, or for an entire received Ethernet frame. Fragments may be located anywhere in system memory, and need not be contiguous.

BIT	BIT NAME	BIT DESCRIPTION
39..0	FragAddr	Fragment Address indicates the location within host system memory of the first byte of the data fragment. Fragments can be located anywhere within a 40 bit (1,099,511,627,776 or 1 Terabyte) memory space.
47..40	Reserved	Reserved for future use. Write as zero, ignore on read.
63..48	FragLen	Fragment Length indicates the length of the fragment in bytes. Fragments can range from zero to 64k (65,536) bytes in length.

11.10.3 RMON Statistics

The Rio implements a portion of the Remote Network Monitoring Management Information Base (RMON MIB) defined in RFC 1757. The Ethernet Statistics Group as defined in RFC 1757 is implemented via a set of Rio registers. Many Ethernet Statistics Group objects are implemented as dedicated registers which must be accessed using memory (not I/O) operations. A memory map of these dedicated RMON registers in shown in Table 15.

TABLE 15: Rubicon RMON MIB Register Map

CLK	BYTE 4	BYTE 3	BYTE 2	BYTE 1	ADDR OFFSET
Tx	EtherStatsPkts1024to1518Octets				0x150
Tx	EtherStatsPkts512to1023Octets				0x14C
Tx	EtherStatsPkts256to511Octets				0x148
Tx	EtherStatsPkts128to255Octets				0x144
Tx	EtherStatsPkts65to127Octets				0x140
Tx	EtherStatsPkts64Octets				0x13C
Tx	EtherStatsPkts				0x138
Tx	EtherStatsOctets				0x134
Tx	EtherStatsJabbers				0x130
Tx	EtherStatsFragments				0x12C
Tx	EtherStatsUndersizePkts				0x128
Tx	EtherStatsCRCAlignErrors				0x124
Tx	EtherStatsPkts1024to1518OctetsTransmit				0x120
Tx	EtherStatsPkts512to1023OctetsTransmit				0x11C
Tx	EtherStatsPkts256to511OctetsTransmit				0x118
Tx	EtherStatsPkts128to255OctetsTransmit				0x114
Tx	EtherStatsPkts65to127OctetsTransmit				0x110
Tx	EtherStatsPkts64OctetsTransmit				0x10C
Tx	EtherStatsPktsTransmit				0x108
Tx	EtherStatsOctetsTransmit				0x104
Tx	EtherStatsCollisions				0x100

The remainder of the Ethernet Statistics Group objects are accessed via dual purpose statistic registers. A list of these dual purpose registers, and their RMON Ethernet Statistics equivalents is shown in Table 16.

Table 16: RMON MIB to Rio Register Equivalent Mapping

RMON MIB ETHERNET STATISTICS GROUP OBJECT	EQUIVALENT RIO STATISTIC REGISTER
EtherStatsDropEvents	FramesLostRxErrors
EtherStatsOversizePkts	FrameTooLongErrors
EtherStatsBroadcastPkts	BcstFramesRcvdOk
EtherStatsMulticastPkts	McstFramesRcvdOk
EtherStatsBroadcastPktsTransmit	BcstFramesXmtdOk
EtherStatsMulticastPktsTransmit	McstFramesXmtdOk

Note, RMON statistic objects with the “Transmit” suffix are not described specifically in RFC 1757, but are extensions of similar objects defined in RFC 1757 applied to transmit data.

11.10.3.1 EtherStatsCollisions

Class..... RMON Statistics
I/O Base Address ..... Not accessible via I/O
Memory Base Address . MemBaseAddress register value
Address Offset..... 0x100
Default Value 0x00000000
Access Rule..... Double Word
Width ..... 32 bits

BIT	BIT NAME	R/W	BIT DESCRIPTION
31..0	EtherStatsCollisions	R/W	From RFC 1757: "The best estimate of the total number of collisions on this Ethernet segment." An UpdateStats interrupt (UpdateStats bit within the IntStatus register) will occur when EtherStatsCollisions reaches a value of 0xC000. EtherStatsCollisions is enabled by writing a logic 1 to the StatisticsEnable bit in the MACCtrl register, and a logic 0 to the EtherStatsCollisions bit within the RMONStatisticsMask register.

11.10.3.2 EtherStatsCRCAlignErrors

Class..... RMON Statistics
I/O Base Address ..... Not accessible via I/O
Memory Base Address . MemBaseAddress register value
Address Offset..... 0x124
Default Value 0x00000000
Access Rule..... Double Word
Width ..... 32 bits

BIT	BIT NAME	R/W	BIT DESCRIPTION
31..0	EtherStatsCR-CAlignErrors	R/W	From RFC 1757: "The total number of packets received that had a length (excluding framing bits, but including FCS octets) of between 64 and 1518 octets, inclusive, but had either a bad Frame Check Sequence (FCS) with an integral number of octets (FCS Error) or a bad FCS with a non-integral number of octets (Alignment Error)." An UpdateStats interrupt (UpdateStats bit within the IntStatus register) will occur when EtherStatsCRCAlignErrors reaches a value of 0xC000. EtherStatsCRCAlignErrors is enabled by writing a logic 1 to the StatisticsEnable bit in the MACCtrl register, and a logic 0 to the EtherStatsCRCAlignErrors bit within the RMONStatisticsMask register.

11.10.3.3 EtherStatsFragments

Class..... RMON Statistics
I/O Base Address ..... Not accessible via I/O
Memory Base Address . MemBaseAddress register value
Address Offset..... 0x12C
Default Value 0x00000000
Access Rule..... Double Word
Width ..... 32 bits

BIT	BIT NAME	R/W	BIT DESCRIPTION
31..0	EtherStatsFrag-ments	R/W	From RFC 1757: "The total number of packets received that were less than 64 octets in length (excluding framing bits but including FCS octets) and had either a bad Frame Check Sequence (FCS) with an integral number of octets (FCS Error) or a bad FCS with a non-integral number of octets (Alignment Error). Note that it is entirely normal for etherStatsFragments to increment. This is because it counts both runts (which are normal occur-rences due to collisions) and noise hits." An UpdateStats interrupt (UpdateStats bit within the IntStatus register) will occur when EtherStatsFragments reaches a value of 0xC000. EtherStatsFragments is enabled by writing a logic 1 to the StatisticsEnable bit in the MACCtrl register, and a logic 0 to the EtherStatsFragments bit within the RMONStatisticsMask register.

11.10.3.4 EtherStatsJabbers

Class..... RMON Statistics
I/O Base Address ..... Not accessible via I/O
Memory Base Address . MemBaseAddress register value
Address Offset..... 0x130
Default Value 0x00000000
Access Rule..... Double Word
Width ..... 32 bits

BIT	BIT NAME	R/W	BIT DESCRIPTION
31..0	EtherStatsJabbers	R/W	From RFC 1757: "The total number of packets received that were longer than 1518 octets (excluding framing bits, but including FCS octets), and had either a bad Frame Check Sequence (FCS) with an integral number of octets (FCS Error) or a bad FCS with a non-integral number of octets (Alignment Error). Note that this definition of jabber is different than the definition in IEEE-802.3." An UpdateStats interrupt (UpdateStats bit within the IntStatus register) will occur when EtherStatsJabbers reaches a value of 0xC000. EtherStatsJabbers is enabled by writing a logic 1 to the StatisticsEnable bit in the MACCtrl register, and a logic 0 to the EtherStatsJabbers bit within the RMONStatisticsMask register.

11.10.3.5 EtherStatsOctets

Class..... RMON Statistics
I/O Base Address ..... Not accessible via I/O
Memory Base Address . MemBaseAddress register value
Address Offset..... 0x134
Default Value 0x00000000
Access Rule..... Double Word
Width ..... 32 bits

BIT	BIT NAME	R/W	BIT DESCRIPTION

31..0	EtherStatsOctets	R/W	From RFC 1757:
			"The total number of octets of data (including those in bad packets) received on the network (excluding framing bits but including FCS octets).
			This object can be used as a reasonable estimate of ethernet utilization. If greater precision is desired, the etherStatsPkts and etherStatsOctets objects should be sampled before and after a common interval. The differences in the sampled values are Pkts and Octets, respectively, and the number of seconds in the interval is Interval. These values are used to calculate the Utilization as follows:
			`Pkts * (96 + 64) + (Octets * 8)`
			`Utilization = ----------------------------------`
			`Interval * 10,000,000`
			The result of this equation is the value Utilization which is the percent utilization of the ethernet segment on a scale of 0 to 100 percent."
			All Rio byte and octet count based statistic registers include the VLAN tag (4 octets) regardless of whether or not the Rio is adding (on frame transmission) or removing (on frame reception) the VLAN tag automatically.
			An UpdateStats interrupt (UpdateStats bit within the IntStatus register) will occur when EtherStatsOctets reaches a value of 0xC000. EtherStatsOctets is enabled by writing a logic 1 to the StatisticsEnable bit in the MACCtrl register, and a logic 0 to the EtherStatsOctets/EtherStatsPkts bit within the RMONStatistics-Mask register.

11.10.3.6 EtherStatsOctetsTransmit

Class..... RMON Statistics
I/O Base Address ..... Not accessible via I/O
Memory Base Address . MemBaseAddress register value
Address Offset..... 0x104
Default Value 0x00000000
Access Rule..... Double Word
Width ..... 32 bits

BIT	BIT NAME	R/W	BIT DESCRIPTION
31..0	EtherStatsOctet-sTransmit	R/W	An extension of the similar named object in RFC 1757 for transmit data as apposed to traditional RMON objects which record receive data. All Rio byte and octet count based statistic registers include the VLAN tag (4 octets) regardless of whether or not the Rio is adding (on frame transmission) or removing (on frame reception) the VLAN tag automatically. An UpdateStats interrupt (UpdateStats bit within the IntStatus register) will occur when EtherStatsOctetsTransmit reaches a value of 0xC000. EtherStatsOctetsTransmit is enabled by writing a logic 1 to the StatisticsEnable bit in the MACCtrl register, and a logic 0 to the EtherStatsOctetsTransmit/EtherStatsPktsTransmit bit within the RMONStatisticsMask register.

11.10.3.7 EtherStatsPkts

Class..... RMON Statistics
I/O Base Address ..... Not accessible via I/O
Memory Base Address . MemBaseAddress register value
Address Offset..... 0x138
Default Value 0x00000000
Access Rule..... Double Word
Width ..... 32 bits

BIT	BIT NAME	R/W	BIT DESCRIPTION
31..0	EtherStatsPkts	R/W	From RFC 1757: “The total number of packets (including bad packets, broadcast packets, and multicast packets) received.” Note, the ReceiveMode register settings affect the results of EtherStatsPkts. An UpdateStats interrupt (UpdateStats bit within the IntStatus register) will occur when EtherStatsPkts reaches a value of 0xC000. EtherStatsPkts is enabled by writing a logic 1 to the StatisticsEnable bit in the MACCtrl register, and a logic 0 to the EtherStatsOctets/EtherStatsPkts bit within the RMONStatistics-Mask register.

11.10.3.8 EtherStatsPkts64Octets

Class..... RMON Statistics
I/O Base Address ..... Not accessible via I/O
Memory Base Address . MemBaseAddress register value
Address Offset..... 0x13C
Default Value 0x00000000
Access Rule..... Double Word
Width ..... 32 bits

BIT	BIT NAME	R/W	BIT DESCRIPTION
31..0	EtherStatsPkts64Octets	R/W	From RFC 1757: "The total number of packets (including bad packets) received that were 64 octets in length (excluding framing bits but including FCS octets)." All Rio byte and octet count based statistic registers include the VLAN tag (4 octets) regardless of whether or not the Rio is adding (on frame transmission) or removing (on frame reception) the VLAN tag automatically. An UpdateStats interrupt (UpdateStats bit within the IntStatus register) will occur when EtherStatsPkts64Octets reaches a value of 0xC000. EtherStatsPkts64Octets is enabled by writing a logic 1 to the StatisticsEnable bit in the MACCtrl register, and a logic 0 to the EtherStatsPkts64Octets bit within the RMONStatisticsMask register.

11.10.3.9 EtherStatsPkts65to127Octets

Class..... RMON Statistics
I/O Base Address ..... Not accessible via I/O
Memory Base Address . MemBaseAddress register value
Address Offset..... 0x140
Default Value 0x00000000
Access Rule..... Double Word
Width ..... 32 bits

BIT	BIT NAME	R/W	BIT DESCRIPTION
31..0	EtherStatsPkts65to127Octets	R/W	From RFC 1757: "The total number of packets (including bad packets) received that were between 65 and 127 octets in length inclusive (excluding framing bits but including FCS octets)." All Rio byte and octet count based statistic registers include the VLAN tag (4 octets) regardless of whether or not the Rio is adding (on frame transmission) or removing (on frame reception) the VLAN tag automatically. An UpdateStats interrupt (UpdateStats bit within the IntStatus register) will occur when EtherStatsPkts65to127Octets reaches a value of 0xC000. EtherStatsPkts65to127Octets is enabled by writing a logic 1 to the StatisticsEnable bit in the MACCtrl register, and a logic 0 to the EtherStatsPkts65to127Octets bit within the RMONStatistics-Mask register.

11.10.3.10 EtherStatsPkts128to255Octets

Class..... RMON Statistics
I/O Base Address ..... Not accessible via I/O
Memory Base Address . MemBaseAddress register value
Address Offset..... 0x144
Default Value 0x00000000
Access Rule..... Double Word
Width ..... 32 bits

BIT	BIT NAME	R/W	BIT DESCRIPTION
31..0	EtherStatsPkts128to255Octets	R/W	From RFC 1757: "The total number of packets (including bad packets) received that were between 128 and 255 octets in length inclusive (excluding framing bits but including FCS octets)." All Rio byte and octet count based statistic registers include the VLAN tag (4 octets) regardless of whether or not the Rio is adding (on frame transmission) or removing (on frame reception) the VLAN tag automatically. An UpdateStats interrupt (UpdateStats bit within the IntStatus register) will occur when EtherStatsPkts128to255Octets reaches a value of 0xC000. EtherStatsPkts128to255Octets is enabled by writing a logic 1 to the StatisticsEnable bit in the MACCtrl register, and a logic 0 to the EtherStatsPkts128to255Octets bit within the RMONStatistics-Mask register.

11.10.3.11 EtherStatsPkts256to511Octets

Class..... RMON Statistics
I/O Base Address ..... Not accessible via I/O
Memory Base Address . MemBaseAddress register value
Address Offset..... 0x148
Default Value 0x00000000
Access Rule..... Double Word
Width ..... 32 bits

BIT	BIT NAME	R/W	BIT DESCRIPTION
31..0	EtherStatsPkts256to511Octets	R/W	From RFC 1757: "The total number of packets (including bad packets) received that were between 256 and 511 octets in length inclusive (excluding framing bits but including FCS octets)." All Rio byte and octet count based statistic registers include the VLAN tag (4 octets) regardless of whether or not the Rio is adding (on frame transmission) or removing (on frame reception) the VLAN tag automatically. An UpdateStats interrupt (UpdateStats bit within the IntStatus register) will occur when EtherStatsPkts256to511Octets reaches a value of 0xC000. EtherStatsPkts256to511Octets is enabled by writing a logic 1 to the StatisticsEnable bit in the MACCtrl register, and a logic 0 to the EtherStatsPkts256to511Octets bit within the RMONStatistics-Mask register.

11.10.3.12 EtherStatsPkts512to1023Octets

Class..... RMON Statistics
I/O Base Address ..... Not accessible via I/O
Memory Base Address . MemBaseAddress register value
Address Offset..... 0x14C
Default Value 0x00000000
Access Rule..... Double Word
Width ..... 32 bits

BIT	BIT NAME	R/W	BIT DESCRIPTION
31..0	EtherStatsPkts512to1023Octets	R/W	From RFC 1757: "The total number of packets (including bad packets) received that were between 512 and 1023 octets in length inclusive (excluding framing bits but including FCS octets)." All Rio byte and octet count based statistic registers include the VLAN tag (4 octets) regardless of whether or not the Rio is adding (on frame transmission) or removing (on frame reception) the VLAN tag automatically. An UpdateStats interrupt (UpdateStats bit within the IntStatus register) will occur when EtherStatsPkts512to1023Octets reaches a value of 0xC000. EtherStatsPkts512to1023Octets is enabled by writing a logic 1 to the StatisticsEnable bit in the MACCtrl register, and a logic 0 to the EtherStatsPkts512to1023Octets bit within the RMONStatisticsMask register.

11.10.3.13 EtherStatsPkts1024to1518Octets

Class..... RMON Statistics
I/O Base Address ..... Not accessible via I/O
Memory Base Address . MemBaseAddress register value
Address Offset..... 0x150
Default Value 0x00000000
Access Rule..... Double Word
Width ..... 32 bits

BIT	BIT NAME	R/W	BIT DESCRIPTION
31..0	EtherStatsPkts1024to1518Octets	R/W	From RFC 1757: "The total number of packets (including bad packets) received that were between 1024 and 1518 octets in length inclusive (excluding framing bits but including FCS octets)." All Rio byte and octet count based statistic registers include the VLAN tag (4 octets) regardless of whether or not the Rio is adding (on frame transmission) or removing (on frame reception) the VLAN tag automatically. An UpdateStats interrupt (UpdateStats bit within the IntStatus register) will occur when EtherStatsPkts1024to1518Octets reaches a value of 0xC000. EtherStatsPkts1024to1518Octets is enabled by writing a logic 1 to the StatisticsEnable bit in the MACCtrl register, and a logic 0 to the EtherStatsPkts1024to1518Octets bit within the RMONStatisticsMask register.

11.10.3.14 EtherStatsPktsTransmit

Class..... RMON Statistics
I/O Base Address ..... Not accessible via I/O
Memory Base Address . MemBaseAddress register value
Address Offset..... 0x108
Default Value 0x00000000
Access Rule..... Double Word
Width ..... 32 bits

BIT	BIT NAME	R/W	BIT DESCRIPTION
31..0	EtherStatsPkt-sTransmit	R/W	An extension of the similar named object in RFC 1757 for transmit data as apposed to traditional RMON objects which record receive data. An UpdateStats interrupt (UpdateStats bit within the IntStatus register) will occur when EtherStatsPktsTransmit reaches a value of 0xC000. EtherStatsPktsTransmit is enabled by writing a logic 1 to the StatisticsEnable bit in the MACCtrl register, and a logic 0 to the EtherStatsOctetsTransmit/EtherStatsPkt-sTransmit bit within the RMONStatisticsMask register.

11.10.3.15 EtherStatsPkts64OctetsTransmit

Class..... RMON Statistics
I/O Base Address ..... Not accessible via I/O
Memory Base Address . MemBaseAddress register value
Address Offset..... 0x10C
Default Value 0x00000000
Access Rule..... Double Word
Width ..... 32 bits

BIT	BIT NAME	R/W	BIT DESCRIPTION
31..0	EtherStatsPkts64OctetsTransmit	R/W	An extension of the similar named object in RFC 1757 for transmit data as apposed to traditional RMON objects which record receive data. All Rio byte and octet count based statistic registers include the VLAN tag (4 octets) regardless of whether or not the Rio is adding (on frame transmission) or removing (on frame reception) the VLAN tag automatically. An UpdateStats interrupt (UpdateStats bit within the IntStatus register) will occur when EtherStatsPkts64OctetsTransmit reaches a value of 0xC000. EtherStatsPkts64OctetsTransmit is enabled by writing a logic 1 to the StatisticsEnable bit in the MACCtrl register, and a logic 0 to the EtherStatsPkts64OctetsTransmit bit within the RMONStatisticsMask register.

11.10.3.16 EtherStatsPkts65to127OctetsTransmit

Class..... RMON Statistics
I/O Base Address ..... Not accessible via I/O
Memory Base Address . MemBaseAddress register value
Address Offset..... 0x110
Default Value 0x00000000
Access Rule..... Double Word
Width ..... 32 bits

BIT	BIT NAME	R/W	BIT DESCRIPTION
31..0	EtherStatsPkts65to127OctetsTransmit	R/W	An extension of the similar named object in RFC 1757 for transmit data as apposed to traditional RMON objects which record receive data. All Rio byte and octet count based statistic registers include the VLAN tag (4 octets) regardless of whether or not the Rio is adding (on frame transmission) or removing (on frame reception) the VLAN tag automatically. An UpdateStats interrupt (UpdateStats bit within the IntStatus register) will occur when EtherStatsPkts65to127OctetsTransmit reaches a value of 0xC000. EtherStatsPkts65to127OctetsTransmit is enabled by writing a logic 1 to the StatisticsEnable bit in the MACCtrl register, and a logic 0 to the EtherStatsPkts65to127OctetsTransmit bit within the RMONStatisticsMask register.

11.10.3.17 EtherStatsPkts128to255OctetsTransmit

Class..... RMON Statistics
I/O Base Address ..... Not accessible via I/O
Memory Base Address . MemBaseAddress register value
Address Offset..... 0x114
Default Value 0x00000000
Access Rule..... Double Word
Width ..... 32 bits

BIT	BIT NAME	R/W	BIT DESCRIPTION
31..0	EtherStatsPkts128to255OctetsTransmit	R/W	An extension of the similar named object in RFC 1757 for transmit data as apposed to traditional RMON objects which record receive data. All Rio byte and octet count based statistic registers include the VLAN tag (4 octets) regardless of whether or not the Rio is adding (on frame transmission) or removing (on frame reception) the VLAN tag automatically. An UpdateStats interrupt (UpdateStats bit within the IntStatus register) will occur when EtherStatsPkts128to255OctetsTransmit reaches a value of 0xC000. EtherStatsPkts128to255OctetsTransmit is enabled by writing a logic 1 to the StatisticsEnable bit in the MACCtrl register, and a logic 0 to the EtherStatsPkts128to255OctetsTransmit bit within the RMONStatisticsMask register.

11.10.3.18 EtherStatsPkts256to511OctetsTransmit

Class..... RMON Statistics
I/O Base Address ..... Not accessible via I/O
Memory Base Address . MemBaseAddress register value
Address Offset..... 0x118
Default Value 0x00000000
Access Rule..... Double Word
Width ..... 32 bits

BIT	BIT NAME	R/W	BIT DESCRIPTION
31..0	EtherStatsPkts256to511OctetsTransmit	R/W	An extension of the similar named object in RFC 1757 for transmit data as apposed to traditional RMON objects which record receive data. All Rio byte and octet count based statistic registers include the VLAN tag (4 octets) regardless of whether or not the Rio is adding (on frame transmission) or removing (on frame reception) the VLAN tag automatically. An UpdateStats interrupt (UpdateStats bit within the IntStatus register) will occur when EtherStatsPkts256to511OctetsTransmit reaches a value of 0xC000. EtherStatsPkts256to511OctetsTransmit is enabled by writing a logic 1 to the StatisticsEnable bit in the MACCtrl register, and a logic 0 to the EtherStatsPkts256to511OctetsTransmit bit within the RMONStatisticsMask register.

11.10.3.19 EtherStatsPkts512to1023OctetsTransmit

Class..... RMON Statistics
I/O Base Address ..... Not accessible via I/O
Memory Base Address . MemBaseAddress register value
Address Offset..... 0x11C
Default Value 0x00000000
Access Rule..... Double Word
Width ..... 32 bits

BIT	BIT NAME	R/W	BIT DESCRIPTION
31..0	EtherStatsPkts512to1023OctetsTransmit	R/W	An extension of the similar named object in RFC 1757 for transmit data as apposed to traditional RMON objects which record receive data. All Rio byte and octet count based statistic registers include the VLAN tag (4 octets) regardless of whether or not the Rio is adding (on frame transmission) or removing (on frame reception) the VLAN tag automatically. An UpdateStats interrupt (UpdateStats bit within the IntStatus register) will occur when EtherStatsPkts512to1023OctetsTransmit reaches a value of 0xC000. EtherStatsPkts512to1023OctetsTransmit is enabled by writing a logic 1 to the StatisticsEnable bit in the MACCtrl register, and a logic 0 to the EtherStatsPkts512to1023OctetsTransmit bit within the RMONStatisticsMask register.

11.10.3.20 EtherStatsPkts1024to1518OctetsTransmit

Class..... RMON Statistics
I/O Base Address ..... Not accessible via I/O
Memory Base Address . MemBaseAddress register value
Address Offset..... 0x120
Default Value 0x00000000
Access Rule..... Double Word
Width ..... 32 bits

BIT	BIT NAME	R/W	BIT DESCRIPTION
31..0	EtherStatsPkts1024to1518OctetsTransmit	R/W	An extension of the similar named object in RFC 1757 for transmit data as apposed to traditional RMON objects which record receive data. All Rio byte and octet count based statistic registers include the VLAN tag (4 octets) regardless of whether or not the Rio is adding (on frame transmission) or removing (on frame reception) the VLAN tag automatically. An UpdateStats interrupt (UpdateStats bit within the IntStatus register) will occur when EtherStatsPkts1024to1518OctetsTransmit reaches a value of 0xC000. EtherStatsPkts1024to1518OctetsTransmit is enabled by writing a logic 1 to the StatisticsEnable bit in the MACCtrl register, and a logic 0 to the EtherStatsPkts1024to1518OctetsTransmit bit within the RMONStatisticsMask register.

11.10.3.21 EtherStatsUndersizePkts

Class..... RMON Statistics
I/O Base Address ..... Not accessible via I/O
Memory Base Address . MemBaseAddress register value
Address Offset..... 0x128
Default Value 0x00000000
Access Rule..... Double Word
Width ..... 32 bits

BIT	BIT NAME	R/W	BIT DESCRIPTION
31..0	EtherStatsUnder-sizePkts	R/W	From RFC 1757: “The total number of packets received that were less than 64 octets long (excluding framing bits, but including FCS octets) and were otherwise well formed.” An UpdateStats interrupt (UpdateStats bit within the IntStatus register) will occur when EtherStatsUndersizePkts reaches a value of 0xC000. EtherStatsUndersizePkts is enabled by writing a logic 1 to the StatisticsEnable bit in the MACCtrl register, and a logic 0 to the EtherStatsUndersizePkts bit within the RMONStatisticsMask register.

11.10.4 Ethernet MIB Statistics

The host interacts with the Rio mainly through slave registers, which occupy 256 bytes in the host system's I/O space, 512 bytes in memory space, or both. Generally, registers are referred to as "I/O registers", implying that the registers may in fact be mapped and accessed by the host system in memory space. These registers must be accessed with instructions that are no larger than the bit-width of the register being accessed. There are several classes of I/O registers, with Ethernet Management Information Base (MIB) Statistics comprising a portion of the total I/O register space. The Ethernet MIB Statistic registers implement several counters defined in the IEEE 802.3 standard.

TABLE 17: Rubicon Ethernet MIB Statistics Register Map

CLK	BYTE 4	BYTE 3	BYTE 2	BYTE 1	ADDR OFFSET
Tx	FramesWEXDeferal		FramesAbortXSColls		FC
Tx	MacControlFramesXmtd		CarrierSenseErrors		F8
Tx	BcstFramesXmtdOk				F4
Tx	SingleColFrames				F0
Tx	MultiColFrames				EC
Tx	LateCollisions				E8
Tx	FramesWDeferredXmt				E4
Tx	McstFramesXmtdOk				E0
Tx	FramesXmtdOk				DC
Tx	BcstOctetXmtOk				D8
Tx	McstOctetXmtOk				D4
Tx	OctetXmtOk				D0
Rx	FramesLostRxErrors		FramesCheckSeqErrors		CC
Rx	InRangeLengthErrors		FrameTooLongErrors		C8
Rx	MacControlFramesRcvd				C4
Rx					C0
Rx	BcstFramesRcvdOk				BC
Rx	McstFramesRcvdOk				B8
Rx	FramesRcvdOk				B4
Rx	BcstOctetRcvOk				B0
Rx	McstOctetRcvdOk				AC
Rx	OctetRcvOk				A8

11.10.4.1 BcstFramesRcvdOk

Class..... Ethernet MIB Statistics
I/O Base Address ..... IoBaseAddress register value
Memory Base Address . MemBaseAddress register value
Address Offset..... 0xBC
Default Value 0x0000
Access Rule..... Word
Width ..... 16 bits

BIT	BIT NAME	R/W	BIT DESCRIPTION
15..0	BcstFramesRcv-dOk	R/W	Broadcast Frames Received OK is the count of the number of frames that are successfully received with destination address equal to the broadcast address (0xFFFFFFFFFFFF). BcstFramesRcvdOk does not include frames received with frames too long, FCS, length or alignment errors, or frames lost due to internal MAC sublayer error (i.e. overrun). BcstFramesRcvdOk will wrap around to zero after reaching 0xFFFF. See IEEE 802.3 Clause 30.3.1.1.22. An UpdateStats interrupt (UpdateStats bit within the IntStatus register) will occur when BcstFramesRcvdOk reaches a value of 0xC000. BcstFramesRcvdOk is enabled by writing a logic 1 to the StatisticsEnable bit in the MACCtrl register, and a logic 0 to the BcstFramesRcvdOk bit within the StatisticsMask register.

11.10.4.2 BcstFramesXmtdOk

Class..... Ethernet MIB Statistics
I/O Base Address ..... IoBaseAddress register value
Memory Base Address . MemBaseAddress register value
Address Offset..... 0xF6
Default Value 0x0000
Access Rule..... Word
Width ..... 16 bits

BIT	BIT NAME	R/W	BIT DESCRIPTION
15..0	BcstFramesXmt-dOk	R/W	Broadcast Frames Transmitted is the count of the number of frames that are successfully transmitted to the broadcast address (0xFFFFFFFFFFFF). Frames transmitted to other multicast addresses are excluded from this statistic. BcstFramesXmtdOk will wrap around to zero after reaching 0xFFFF. See IEEE 802.3 Clause 30.3.1.1.19. An UpdateStats interrupt (UpdateStats bit within the IntStatus register) will occur when BcstFramesXmtdOk reaches a value of 0xC000. BcstFramesXmtdOk is enabled by writing a logic 1 to the StatisticsEnable bit in the MACCtrl register, and a logic 0 to the BcstFramesXmtdOk bit within the StatisticsMask register.

11.10.4.3 BcstOctetRcvOk

Class..... Ethernet MIB Statistics
I/O Base Address ..... IoBaseAddress register value
Memory Base Address . MemBaseAddress register value
Address Offset..... 0xB0
Default Value 0x00000000
Access Rule..... Double Word
Width ..... 32 bits

BIT	BIT NAME	R/W	BIT DESCRIPTION
31..0	BcstOctetRcvOk	R/W	Broadcast Octets Received OK is the count of the number of data and padding octets in frames, with destination address equal to the broadcast address (0xFFFFFFFFFFFF), that are successfully received. BcstOctetRcvOk does not include frames received with frames too long, FCS, length or alignment errors, or frames lost due to internal MAC sublayer error (i.e. overrun). BcstOctetRcvOk will wrap around to zero after reaching 0xFFFFFFFF. All Rio byte and octet count based statistic registers include the VLAN tag (4 octets) regardless of whether or not the Rio is adding (on frame transmission) or removing (on frame reception) the VLAN tag automatically. An UpdateStats interrupt (UpdateStats bit within the IntStatus register) will occur when BcstOctetRcvOk reaches a value of 0xC0000000. BcstOctetRcvOk is enabled by writing a logic 1 to the StatisticsEnable bit in the MACCtrl register, and a logic 0 to the BcstOctetRcvOk bit within the StatisticsMask register.

11.10.4.4 BcstOctetXmtOk

Class..... Ethernet MIB Statistics
I/O Base Address ..... IoBaseAddress register value
Memory Base Address . MemBaseAddress register value
Address Offset..... 0xD8
Default Value 0x00000000
Access Rule..... Double Word
Width ..... 32 bits

BIT	BIT NAME	R/W	BIT DESCRIPTION
31..0	BcstOctetXmtOk	R/W	Broadcast Octets Transmitted OK is a count of data and padding octets of frames successfully transmitted to a the broadcast address (0xFFFFFFFFFFFF). BcstOctetXmtOk will wrap around to zero after reaching 0xFFFFFFFF. All Rio byte and octet count based statistic registers include the VLAN tag (4 octets) regardless of whether or not the Rio is adding (on frame transmission) or removing (on frame reception) the VLAN tag automatically. An UpdateStats interrupt (UpdateStats bit within the IntStatus register) will occur when BcstOctetXmtOk reaches a value of 0xC0000000. BcstOctetXmtOk is enabled by writing a logic 1 to the StatisticsEnable bit in the MACCtrl register, and a logic 0 to the BcstOctetXmtOk bit within the StatisticsMask register.

11.10.4.5 CarrierSenseErrors

Class..... Ethernet MIB Statistics
I/O Base Address ..... IoBaseAddress register value
Memory Base Address . MemBaseAddress register value
Address Offset..... 0xF8
Default Value 0x0000
Access Rule..... Word
Width ..... 16 bits

BIT	BIT NAME	R/W	BIT DESCRIPTION
15..0	CarrierSenseErrors	R/W	Carrier Sense Errors counts the number of times that the carrier sense signal (CRS) was de-asserted (a logic 0) during the transmission of a frame without collision. The carrier sense signal is not monitored for the purpose of this statistic until after the preamble and start-of-frame delimiter fields of the Ethernet frame have been transmitted. CarrierSenseErrors will wrap around to zero after reaching 0xFFFF. See IEEE 802.3 Clause 30.3.1.1.13. An UpdateStats interrupt (UpdateStats bit within the IntStatus register) will occur when CarrierSenseErrors reaches a value of 0xC000. CarrierSenseErrors is enabled by writing a logic 1 to the StatisticsEnable bit in the MACCtrl register, and a logic 0 to the CarrierSenseErrors bit within the StatisticsMask register.

11.10.4.6 FramesAbortXSColls

Class..... Ethernet MIB Statistics
I/O Base Address ..... IoBaseAddress register value
Memory Base Address . MemBaseAddress register value
Address Offset..... 0xFC
Default Value 0x0000
Access Rule..... Word
Width ..... 16 bits

BIT	BIT NAME	R/W	BIT DESCRIPTION
15..0	FramesAbortX-SColls	R/W	Frames Aborted Due to Excess Collisions counts the number of frames which are not transmitted successfully due to excessive collisions. FramesAbortXSColls will wrap around to zero after reaching 0xFFFF. See IEEE 802.3 Clause 30.3.1.1.11. An UpdateStats interrupt (UpdateStats bit within the IntStatus register) will occur when FramesAbortXSColls reaches a value of 0xC000. FramesAbortXSColls is enabled by writing a logic 1 to the StatisticsEnable bit in the MACCtrl register, and a logic 0 to the FramesAbortXSColls bit within the StatisticsMask register.

11.10.4.7 FramesCheckSeqErrors

Class..... Ethernet MIB Statistics
I/O Base Address ..... IoBaseAddress register value
Memory Base Address . MemBaseAddress register value
Address Offset..... 0xCC
Default Value 0x0000
Access Rule..... Word
Width ..... 16 bits

BIT	BIT NAME	R/W	BIT DESCRIPTION
15..0	FramesCheckSeqErrors	R/W	Frame Check Sequence Errors is a count of received frames that are an integral number of octets in length and do not pass the FCS check. This does not include frames received with frame-too-long, or frame-too-short (runt) errors. FramesCheckSeqErrors will wrap around to zero after reaching 0xFFFF. See IEEE 802.3 Clause 30.3.1.1.6. An UpdateStats interrupt (UpdateStats bit within the IntStatus register) will occur when FramesCheckSeqErrors reaches a value of 0xC000. FramesCheckSeqErrors is enabled by writing a logic 1 to the StatisticsEnable bit in the MACCtrl register, and a logic 0 to the FramesCheckSeqErrors bit within the StatisticsMask register.

11.10.4.8 FramesLostRxErrors

Class..... Ethernet MIB Statistics
I/O Base Address ..... IoBaseAddress register value
Memory Base Address . MemBaseAddress register value
Address Offset..... 0xCE
Default Value 0x0000
Access Rule..... Word
Width ..... 16 bits

BIT	BIT NAME	R/W	BIT DESCRIPTION
15..0	FramesLostRxErrors	R/W	Frames Lost Due to Receive Errors is a count of the number of frames that should have been received (the destination address matched the filter criteria) but experienced a receive FIFO overrun error (the receive FIFO does not have enough free space to store the received data). FramesLostRxErrors only includes overruns that become apparent to the host system, and does not include frames that are completely ignored due to a completely full receive FIFO at the beginning of frame reception. FramesLostRxErrors will wrap around to zero after reaching 0xFFFF. See IEEE 802.3 Clause 30.3.1.1.15. An UpdateStats interrupt (UpdateStats bit within the IntStatus register) will occur when FramesLostRxErrors reaches a value of 0xC000. FramesLostRxErrors is enabled by writing a logic 1 to the StatisticsEnable bit in the MACCtrl register, and a logic 0 to the FramesLostRxErrors bit within the StatisticsMask register.

11.10.4.9 FramesRcvdOk

Class..... Ethernet MIB Statistics
I/O Base Address ..... IoBaseAddress register value
Memory Base Address . MemBaseAddress register value
Address Offset..... 0xB4
Default Value 0x00000000
Access Rule..... Double Word
Width ..... 32 bits

BIT	BIT NAME	R/W	BIT DESCRIPTION
31..0	FramesRcvdOk	R/W	Frames Received OK is the count of the number of frames that are successfully received. FramesRcvdOk does not include frames received with frames too long, FCS, length or alignment errors, or frames lost due to internal MAC sublayer error (i.e. overrun). FramesRcvdOk will wrap around to zero after reaching 0xFFFFFFFF. See IEEE 802.3 Clause 30.3.1.1.5. An UpdateStats interrupt (UpdateStats bit within the IntStatus register) will occur when FramesRcvdOk reaches a value of 0xC0000000. FramesRcvdOk is enabled by writing a logic 1 to the StatisticsEnable bit in the MACCtrl register, and a logic 0 to the FramesRcvdOk bit within the StatisticsMask register.

11.10.4.10 FramesWDeferredXmt

Class..... Ethernet MIB Statistics
I/O Base Address ..... IoBaseAddress register value
Memory Base Address . MemBaseAddress register value
Address Offset..... 0xE4
Default Value 0x00000000
Access Rule..... Double Word
Width ..... 32 bits

BIT	BIT NAME	R/W	BIT DESCRIPTION
31..0	FramesWDeferredXmt	R/W	Frames with Deferred Transmit is a count of the number of frames that must delay their first attempt of transmission because the medium was busy. Frames involved in any collisions are not counted by FramesWDeferredXmt. FramesWDeferredXmt wrap around to zero after reaching 0xFFFFFFFF. See IEEE 802.3 Clause 30.3.1.1.9. An UpdateStats interrupt (UpdateStats bit within the IntStatus register) will occur when FramesWDeferredXmt reaches a value of 0xC0000000. FramesWDeferredXmt is enabled by writing a logic 1 to the StatisticsEnable bit in the MACCtrl register, and a logic 0 to the FramesWDeferredXmt bit within the StatisticsMask register.

11.10.4.11 FramesWEXDeferal

Class..... Ethernet MIB Statistics
I/O Base Address ..... IoBaseAddress register value
Memory Base Address . MemBaseAddress register value
Address Offset..... 0xFE
Default Value 0x0000
Access Rule..... Word
Width ..... 16 bits

BIT	BIT NAME	R/W	BIT DESCRIPTION
15..0	FramesWEXDeferal	R/W	Frames with Excessive Deferals counts the number of frames that deferred for an excessive period of time (exceeding the defer limit). FramesWEXDeferal is only incremented once per LLC frame. FramesWEXDeferal will wrap around to zero after reaching 0xFFFF. See IEEE 802.3 Clause 30.3.1.1.20. An UpdateStats interrupt (UpdateStats bit within the IntStatus register) will occur when FramesWEXDeferal reaches a value of 0xC000. FramesWEXDeferal is enabled by writing a logic 1 to the StatisticsEnable bit in the MACCtrl register, and a logic 0 to the FramesWEXDeferal bit within the StatisticsMask register.

11.10.4.12 FramesXmtdOk

Class..... Ethernet MIB Statistics
I/O Base Address ..... IoBaseAddress register value
Memory Base Address . MemBaseAddress register value
Address Offset..... 0xDC
Default Value 0x00000000
Access Rule..... Double Word
Width ..... 32 bits

BIT	BIT NAME	R/W	BIT DESCRIPTION
31..0	FramesXmtdOk	R/W	Frames Transmitted OK is a count of the number of frames that are successfully transmitted. FramesXmtdOk will wrap around to zero after reaching 0xFFFFFFFF. See IEEE 802.3 Clause 30.3.1.1.2. An UpdateStats interrupt (UpdateStats bit within the IntStatus register) will occur when FramesXmtdOk reaches a value of 0xC0000000. FramesXmtdOk is enabled by writing a logic 1 to the StatisticsEnable bit in the MACCtrl register, and a logic 0 to the FramesXmtdOk bit within the StatisticsMask register.

11.10.4.13 FrameTooLongErrors

Class..... Ethernet MIB Statistics
I/O Base Address ..... IoBaseAddress register value
Memory Base Address . MemBaseAddress register value
Address Offset..... 0xC8
Default Value 0x0000
Access Rule..... Word
Width ..... 16 bits

BIT	BIT NAME	R/W	BIT DESCRIPTION
15..0	FrameTooLongErrors	R/W	Frame Too Long Errors is a count of frames received whose length exceed the value in the MaxFrameSize register. FrameTooLongErrors will wrap around to zero after reaching 0xFFFF. See IEEE 802.3 Clause 30.3.1.1.25. An UpdateStats interrupt (UpdateStats bit within the IntStatus register) will occur when FrameTooLongErrors reaches a value of 0xC000. FrameTooLongErrors is enabled by writing a logic 1 to the StatisticsEnable bit in the MACCtrl register, and a logic 0 to the FrameTooLongErrors bit within the StatisticsMask register.

11.10.4.14 InRangeLengthErrors

Class..... Ethernet MIB Statistics
I/O Base Address ..... IoBaseAddress register value
Memory Base Address . MemBaseAddress register value
Address Offset..... 0xCA
Default Value 0x0000
Access Rule..... Word
Width ..... 16 bits

BIT	BIT NAME	R/W	BIT DESCRIPTION
15..0	InRangeLengthErrors	R/W	In Range Length Errors is a count of the number of frames with the Length/Type field value between the minimum unpadded MAC client data size and the maximum allowed MAC client data size, inclusive, that does not match the number of MAC client data octets received. InRangeLengthErrors also increments for frames whose Length/Type field value is less than the minimum allowed unpadded MAC client data size, and the number of MAC client data octets received is greater than the minimum unpadded MAC client data size. InRangeLengthErrors will wrap around to zero after reaching 0xFFFF. See IEEE 802.3 Clause 30.3.1.1.23. An UpdateStats interrupt (UpdateStats bit within the IntStatus register) will occur when InRangeLengthErrors reaches a value of 0xC000. InRangeLengthErrors is enabled by writing a logic 1 to the StatisticsEnable bit in the MACCtrl register, and a logic 0 to the InRangeLengthErrors bit within the StatisticsMask register.

11.10.4.15 LateCollisions

Class..... Ethernet MIB Statistics
I/O Base Address ..... IoBaseAddress register value
Memory Base Address . MemBaseAddress register value
Address Offset..... 0xE8
Default Value 0x00000000
Access Rule..... Double Word
Width ..... 32 bits

BIT	BIT NAME	R/W	BIT DESCRIPTION
31..0	LateCollisions	R/W	Late Collisions is a count of the number of times that a collision has been detected later than 1 slot time into the transmitted frame. LateCollisions will wrap around to zero after reaching 0xFFFFFFFF. See IEEE 802.3 Clause 30.3.1.1.10. An UpdateStats interrupt (UpdateStats bit within the IntStatus register) will occur when LateCollisions reaches a value of 0xC0000000. LateCollisions is enabled by writing a logic 1 to the StatisticsEnable bit in the MACCtrl register, and a logic 0 to the LateCollisions bit within the StatisticsMask register.

11.10.4.16 MacControlFramesRcvd

Class..... Ethernet MIB Statistics
I/O Base Address ..... IoBaseAddress register value
Memory Base Address . MemBaseAddress register value
Address Offset..... 0xC6
Default Value 0x0000
Access Rule..... Word
Width ..... 16 bits

BIT	BIT NAME	R/W	BIT DESCRIPTION
15..0	MacControl-FramesRcvd	R/W	MAC Control Frames Received is a count of the number of MAC control PAUSE frames, and only PAUSE frames, received successfully. MacControlFramesRcvd will wrap around to zero after reaching 0xFFFF. An UpdateStats interrupt (UpdateStats bit within the IntStatus register) will occur when MacControlFramesRcvd reaches a value of 0xC000. MacControlFramesRcvd is enabled by writing a logic 1 to the StatisticsEnable bit in the MACCtrl register, and a logic 0 to the MacControlFramesRcvd bit within the StatisticsMask register.

11.10.4.17 MacControlFramesXmtd

Class..... Ethernet MIB Statistics
I/O Base Address ..... IoBaseAddress register value
Memory Base Address . MemBaseAddress register value
Address Offset..... 0xFA
Default Value 0x0000
Access Rule..... Word
Width ..... 16 bits

BIT	BIT NAME	R/W	BIT DESCRIPTION
15..0	MacControl-FramesXmtd	R/W	MAC Control Frames Transmitted is the count of MAC control frames transmitted by the Rio. Note, MacControlFramesXmtd does not include MAC control frames transferred to the Rio by the host system via the transmit DMA process. An UpdateStats interrupt (UpdateStats bit within the IntStatus register) will occur when MacControlFramesXmtd reaches a value of 0xC000. MacControlFramesXmtd is enabled by writing a logic 1 to the StatisticsEnable bit in the MACCtrl register, and a logic 0 to the MacControlFramesXmtd bit within the StatisticsMask register.

11.10.4.18 McstFramesRcvdOk

Class..... Ethernet MIB Statistics
I/O Base Address ..... IoBaseAddress register value
Memory Base Address . MemBaseAddress register value
Address Offset..... 0xB8
Default Value 0x00000000
Access Rule..... Double Word
Width ..... 32 bits

BIT	BIT NAME	R/W	BIT DESCRIPTION
31..0	McstFramesRcv-dOk	R/W	Broadcast Frames Received OK is the count of the number of frames that are successfully received to a group destination address other than the broadcast address (0xFFFFFFFFFFFF). McstFramesRcvdOk does not include frames received with frames too long, FCS, length or alignment errors, or frames lost due to internal MAC sublayer error (i.e. overrun). McstFramesRcvdOk will wrap around to zero after reaching 0xFFFFFFFF. See IEEE 802.3 Clause 30.3.1.1.21. An UpdateStats interrupt (UpdateStats bit within the IntStatus register) will occur when McstFramesRcvdOk reaches a value of 0xC0000000. McstFramesRcvdOk is enabled by writing a logic 1 to the StatisticsEnable bit in the MACCtrl register, and a logic 0 to the McstFramesRcvdOk bit within the StatisticsMask register.

11.10.4.19 McstFramesXmtdOk

Class..... Ethernet MIB Statistics
I/O Base Address ..... IoBaseAddress register value
Memory Base Address . MemBaseAddress register value
Address Offset..... 0xE0
Default Value 0x00000000
Access Rule..... Double Word
Width ..... 32 bits

BIT	BIT NAME	R/W	BIT DESCRIPTION
31..0	McstFramesXmt-dOk	R/W	Multicast Frames Transmitted OK is a count of the number of frames that are successfully transmitted to a group destination address other than the broadcast address (0xFFFFFFFFFFFF). McstFramesXmtdOk will wrap around to zero after reaching 0xFFFFFFFF. See IEEE 802.3 Clause 30.3.1.1.18. An UpdateStats interrupt (UpdateStats bit within the IntStatus register) will occur when McstFramesXmtdOk reaches a value of 0xC0000000. McstFramesXmtdOk is enabled by writing a logic 1 to the StatisticsEnable bit in the MACCtrl register, and a logic 0 to the McstFramesXmtdOk bit within the StatisticsMask register.

11.10.4.20 McstOctetRcvdOk

Class..... Ethernet MIB Statistics
I/O Base Address ..... IoBaseAddress register value
Memory Base Address . MemBaseAddress register value
Address Offset..... 0xAC
Default Value 0x00000000
Access Rule..... Double Word
Width ..... 32 bits

BIT	BIT NAME	R/W	BIT DESCRIPTION
31..0	McstOctetRcvdOk	R/W	Multicast Octets Received OK is the count of the number of data and padding octets in frames, to a group destination address other than the broadcast address (0xFFFFFFFFFFFF), that are successfully received. McstOctetRcvdOk does not include frames received with frames too long, FCS, length or alignment errors, or frames lost due to internal MAC sublayer error (i.e. overrun). McstOctetRcvdOk will wrap around to zero after reaching 0xFFFFFFFF. All Rio byte and octet count based statistic registers include the VLAN tag (4 octets) regardless of whether or not the Rio is adding (on frame transmission) or removing (on frame reception) the VLAN tag automatically. An UpdateStats interrupt (UpdateStats bit within the IntStatus register) will occur when McstOctetRcvdOk reaches a value of 0xC0000000. McstOctetRcvdOk is enabled by writing a logic 1 to the StatisticsEnable bit in the MACCtrl register, and a logic 0 to the McstOctetRcvdOk bit within the StatisticsMask register.

11.10.4.21 McstOctetXmtOk

Class..... Ethernet MIB Statistics
I/O Base Address ..... IoBaseAddress register value
Memory Base Address . MemBaseAddress register value
Address Offset..... 0xD4
Default Value 0x00000000
Access Rule..... Double Word
Width ..... 32 bits

BIT	BIT NAME	R/W	BIT DESCRIPTION
31..0	McstOctetXmtOk	R/W	Multicast Octets Transmitted OK is a count of data and padding octets of frames successfully transmitted to a group destination address other than the broadcast address (0xFFFFFFFFFFFF). McstOctetXmtOk will wrap around to zero after reaching 0xFFFFFFFF. All Rio byte and octet count based statistic registers include the VLAN tag (4 octets) regardless of whether or not the Rio is adding (on frame transmission) or removing (on frame reception) the VLAN tag automatically. An UpdateStats interrupt (UpdateStats bit within the IntStatus register) will occur when McstOctetXmtOk reaches a value of 0xC0000000. McstOctetXmtOk is enabled by writing a logic 1 to the StatisticsEnable bit in the MACCtrl register, and a logic 0 to the McstOctetXmtOk bit within the StatisticsMask register.

11.10.4.22 MultiColFrames

Class..... Ethernet MIB Statistics
I/O Base Address ..... IoBaseAddress register value
Memory Base Address . MemBaseAddress register value
Address Offset..... 0xEC
Default Value 0x00000000
Access Rule..... Double Word
Width ..... 32 bits

BIT	BIT NAME	R/W	BIT DESCRIPTION
31..0	MultiColFrames	R/W	Multiple Collision Frames is a count of the number of frames that are involved in more than one collision and are subsequently transmitted successfully. MultiColFrames will wrap around to zero after reaching 0xFFFFFFFF. See IEEE 802.3 Clause 30.3.1.1.4. An UpdateStats interrupt (UpdateStats bit within the IntStatus register) will occur when MultiColFrames reaches a value of 0xC0000000. MultiColFrames is enabled by writing a logic 1 to the StatisticsEnable bit in the MACCtrl register, and a logic 0 to the MultiColFrames bit within the StatisticsMask register.

11.10.4.23 OctetRcvOk

Class..... Ethernet MIB Statistics
I/O Base Address ..... IoBaseAddress register value
Memory Base Address . MemBaseAddress register value
Address Offset..... 0xA8
Default Value 0x00000000
Access Rule..... Double Word
Width ..... 32 bits

BIT	BIT NAME	R/W	BIT DESCRIPTION
31..0	OctetRcvOk	R/W	Octets Received OK is the count of the number of data and padding octets in frames that are successfully received. OctetRcvOk does not include frames received with frames too long, FCS, length or alignment errors, or frames lost due to internal MAC sublayer error (i.e. overrun). OctetRcvOk will wrap around to zero after reaching 0xFFFFFFFF. See IEEE 802.3 Clause 30.3.1.1.14. All Rio byte and octet count based statistic registers include the VLAN tag (4 octets) regardless of whether or not the Rio is adding (on frame transmission) or removing (on frame reception) the VLAN tag automatically. An UpdateStats interrupt (UpdateStats bit within the IntStatus register) will occur when OctetRcvOk reaches a value of 0xC0000000. OctetRcvOk is enabled by writing a logic 1 to the StatisticsEnable bit in the MACCtrl register, and a logic 0 to the OctetRcvOk bit within the StatisticsMask register.

11.10.4.24 OctetXmtOk

Class..... Ethernet MIB Statistics
I/O Base Address ..... IoBaseAddress register value
Memory Base Address . MemBaseAddress register value
Address Offset..... 0xD0
Default Value 0x00000000
Access Rule..... Double Word
Width ..... 32 bits

BIT	BIT NAME	R/W	BIT DESCRIPTION
31..0	OctetXmtOk	R/W	Octets Transmitted OK is a count of data and padding octets of frames successfully transmitted. OctetXmtOk will wrap around to zero after reaching 0xFFFFFFFF. See IEEE 802.3 Clause 30.3.1.1.8. All Rio byte and octet count based statistic registers include the VLAN tag (4 octets) regardless of whether or not the Rio is adding (on frame transmission) or removing (on frame reception) the VLAN tag automatically. An UpdateStats interrupt (UpdateStats bit within the IntStatus register) will occur when OctetXmtOk reaches a value of 0xC0000000. OctetXmtOk is enabled by writing a logic 1 to the StatisticsEnable bit in the MACCtrl register, and a logic 0 to the OctetXmtOk bit within the StatisticsMask register.

11.10.4.25 SingleColFrames

Class..... Ethernet MIB Statistics
I/O Base Address ..... IoBaseAddress register value
Memory Base Address . MemBaseAddress register value
Address Offset..... 0xF0
Default Value 0x00000000
Access Rule..... Double Word
Width ..... 32 bits

BIT	BIT NAME	R/W	BIT DESCRIPTION
31..0	SingleColFrames	R/W	Single Collision Frames is a count of the number of frames that are involved in a single collision, and are subsequently transmitted successfully. SingleColFrames will wrap around to zero after reaching 0xFFFFFFFF. See IEEE 802.3 Clause 30.3.1.1.3. An UpdateStats interrupt (UpdateStats bit within the IntStatus register) will occur when SingleColFrames reaches a value of 0xC0000000. SingleColFrames is enabled by writing a logic 1 to the StatisticsEnable bit in the MACCtrl register, and a logic 0 to the SingleColFrames bit within the StatisticsMask register.

11.10.5 I/O Registers

The host interacts with the Rio mainly through slave registers, which occupy 256 bytes in the host system's I/O space, 512 bytes in memory space, or both. Generally, registers are referred to as "I/O registers", implying that the registers may in fact be mapped and accessed by the host system in memory space. These registers must be accessed with instructions that are no larger than the bit-width of the register being accessed. There are several classes of I/O registers including Statistic, Control and Status, DMA, ASIC, and Interrupts.

TABLE 18: Rubicon I/O Register Map

CLK	BYTE 4	BYTE 3	BYTE 2	BYTE 1	ADDR OFFSET
Tx					FC
Tx					F8
Tx			TxJumboFrames		F4
Tx					F0
Tx					EC
Tx					E8
Tx					E4
Tx					E0
Tx					DC
Tx					D8
Tx					D4
Tx					D0
Rx					CC
Rx					C8
Rx			UDPCheckSumErrors		C4
Rx	IPCheckSumErrors		TCPCheckSumErrors		C0
Rx			RxJumboFrames		BC
Rx					B8
Rx					B4

TABLE 18: Rubicon I/O Register Map

CLK	BYTE 4	BYTE 3	BYTE 2	BYTE 1	ADDR OFFSET
Rx					B0
Rx					AC
Rx					A8
					A4
Tx					A0
Tx	StatisticsMask				9C
	RMONStatisticsMask				98
					94
Rx	HashTable[63:32]				90
Rx	HashTable[31:0]				8C
Rx	VLANHashTable		ReceiveMode		88
Rx	MaxFrameSize				84
		VLANId			80
Tx			StationAddress[47:32]		7C
Tx	StationAddress[31:0]				78
Tx		PhyCtrl			74
Tx	VLANTag				70
Tx	MACCtrl				6C
					68
					64
Tx	TxStatus				60
Tx	IntStatus		IntEnable		5C
Tx	IntStatusAck				58
Tx	Countdown				54
Tx			WakeEvent	ExpRomData	50
Tx	ExpRomAddr				4C
Tx	EepromCtrl		EepromData		48

TABLE 18: Rubicon I/O Register Map

CLK	BYTE 4	BYTE 3	BYTE 2	BYTE 1	ADDR OFFSET
Tx			TxStartThresh		44
					40
Rx	FlowOnThresh		FlowOffThresh		3C
Rx	RxEarlyThresh		FIFOCtrl		38
					34
Tx	AsicCtrl				30
PCI			DebugCtrl		2C
PCI	RxDMAIntCtrl				28
PCI		RxDMAPollPe-riod	RxDMAUrgent-Thresh	RxDMABurst-Thresh	24
PCI	RFDListPtr[63:32]				20
PCI	RFDListPtr[31:0]				1C
PCI		TxDMAPollPe-riod	TxDMAUrgent-Thresh	TxDMABurst-Thresh	18
PCI	TFDListPtr[63:32]				14
PCI	TFDListPtr[31:0]				10
					0C
PCI	Reserved(RxDMAStatus)				08
					04
PCI	DMACtrl				00

11.10.5.1 AsicCtrl

Class..... I/O Registers, Control and Status

I/O Base Address ..... IoBaseAddress register value

Memory Base Address . MemBaseAddress register value

Address Offset..... 0x30

Default Value ..... 0x00004000 (default values for bits 1, 5, 6, 7 are dependent on EEPROM settings, and bits 8, 9, 10 are dependent on ED signal pin states)

Access Rule..... Word, Double Word

Width ..... 32 bits

The contents of bits 7 through 0 of AsicCtrl are read from EEPROM at reset.

<table>
<tr><th>BIT</th><th>BIT NAME</th><th>R/W</th><th>BIT DESCRIPTION</th></tr>
<tr><td>0</td><td>Reserved (ExpRomDisable)</td><td>N/A</td><td>Reserved for future use. Write as zero, ignore on read.
(This bit, when set, disables accesses to the on-adapter Expansion ROM. This bit is included to allow bypassing the Expansion ROM without having to physically remove it from the board. When this bit is set, the Rio responds to any read in its configured Expansion ROM space by returning 00000000h, and it ignores writes to the Expansion ROM. This bit resets to 0.)</td></tr>
<tr><td>1</td><td>ExpRomSize</td><td>R/W</td><td>Expansion Read Only Memory Size. ExpRomSize specifies the size of the Expansion ROM to be used with the Rio.
<table>
<tr><th>BIT 1</th><th>EXPANSION ROM SIZE (KB)</th></tr>
<tr><td>0</td><td>32</td></tr>
<tr><td>1</td><td>64</td></tr>
</table>
</td></tr>
<tr><td>2</td><td>Reserved (TxLargeEnable)</td><td>N/A</td><td>Reserved for future use. Write as zero, ignore on read.
(This read/write bit, when set, enables transmission of frames that are larger than the TxFIFO. Since Rio’s TxFIFO size is 2KB, this bit can be left clear (the reset default).)</td></tr>
<tr><td>3</td><td>Reserved (RxLargeEnable)</td><td>N/A</td><td>Reserved for future use. Write as zero, ignore on read.
(This read/write bit, when set, enables reception of frames that are larger than the RxFIFO. Since Rio’s RxFIFO size is 2KB, this bit can be left clear (the reset default).)</td></tr>
<tr><td>4</td><td>PhySpeed10</td><td>R</td><td>Physical Layer Speed 10Mbps. When PhySpeed10 is a logic 1, 10Mbps operation is available from the PHY device interfaced to the Rio. When PhySpeed10 is a logic 0 the PHY device interfaced to the Rio is not 10Mbps capable.</td></tr>
</table>

<table>
<tr><th>BIT</th><th>BIT NAME</th><th>R/W</th><th>BIT DESCRIPTION</th></tr>
<tr><td>5</td><td>PhySpeed100</td><td>R</td><td>Physical Layer Speed 100Mbps. When PhySpeed100 is a logic 1, 100Mbps operation is available from the PHY device interfaced to the Rio. When PhySpeed100 is a logic 0 the PHY device interfaced to the Rio is not 100Mbps capable.</td></tr>
<tr><td>6</td><td>PhySpeed1000</td><td>R</td><td>Physical Layer Speed 1000Mbps. When PhySpeed1000 is a logic 1, 1000Mbps operation is available from the PHY device interfaced to the Rio. When PhySpeed1000 is a logic 0 the PHY device interfaced to the Rio is not 1000Mbps capable.</td></tr>
<tr><td>7</td><td>PhyMedia</td><td>R</td><td>Physical Layer Media. PhyMedia indicates the media type (fiber or copper) available from the PHY device interfaced to the Rio. When PhyMedia is a logic 1, the PHY device is utilizing fiber media. When PhyMedia is a logic 0, the PHY device is utilizing twisted-pair media.
The combination of PhyMedia, PhySpeed10, PhySpeed100, and PhySpeed1000 indicate the capability of the PHY device interfaced to the Rio.
<table>
<tr><th>PHYMEDIA</th><th>PHYSPEED1000</th><th>PHYSPEED100</th><th>PHYSPEED10</th><th>TECHNOLOGIES SUPPORTED BY PHYSICAL LAYER DEVICE INTERFACED TO THE RIO</th></tr>
<tr><td>X</td><td>0</td><td>0</td><td>0</td><td>Undefined</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>1</td><td>10Mbps copper</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>0</td><td>100Mbps copper</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>0</td><td>1000Mbps copper</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>1</td><td>10Mbps fiber</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>0</td><td>100Mbps fiber</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>0</td><td>1000Mbps fiber</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>BIT</th><th>BIT NAME</th><th>R/W</th><th>BIT DESCRIPTION</th></tr>
<tr><td>10..8</td><td>ForcedConfig</td><td>R/W</td><td>Forced Configuration. ForcedConfig is used to enable and select a Forced Configuration mode for the Rio. Forced Configuration mode is targeted toward embedded applications which do not utilize an EEPROM. In Forced Configuration mode, the Rio is accessed via a PCI bus without first performing PCI configuration or loading parameters from an EEPROM.
The ForcedConfig bits 10 through 8 are latched (with a logic inversion) from signal pins ED bits 2 through 0 respectively upon termination of a Rio reset.
<table>
<tr><th>BIT 10</th><th>BIT 9</th><th>BIT 8</th><th>FORCED CONFIGURATION MODE</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>None</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>1</td></tr>
<tr><td>0</td><td>1</td><td>x</td><td>Reserved</td></tr>
<tr><td>1</td><td>x</td><td>x</td><td>Reserved (alternate DeviceId and VendorId used)</td></tr>
</table>
In Forced Configuration mode 1, the Rio is configured as follows:
I/O base address = 0x200
I/O target cycles = enabled
Memory target cycles = disabled
Bus master cycles = enabled
Expansion ROM cycles = disabled.</td></tr>
<tr><td>11</td><td>(Reserved)
D3ResetDisable</td><td>R/W</td><td>Reserved for future use. Write as zero, ignore on read.
(D3 Power State Reset Disable. When D3ResetDisable is a logic 1 the Rio is configured for operation in a Wake-On-LAN environment (see section 11.5). When D3ResetDisable is a logic 1 and the Rio is in the D3 power state, assertion of RSTN signal pin will not reset the Rio, assertion of the RSTN signal will not reset the Rio.)</td></tr>
<tr><td>12</td><td>Reserved</td><td>N/A</td><td>Reserved for future use. Write as zero, ignore on read.</td></tr>
<tr><td>13</td><td>SpeedupMode</td><td>R/W</td><td>Speed Up Mode. SpeedupMode is used for simulation purposes only. When SpeedupMode is a logic 1 Rio operation is modified to decrease simulation time. SpeedupMode is latched (with a logic inversion) from signal pins ED bit 5 upon termination of a Rio reset.</td></tr>
</table>

<table>
<tr><th>BIT</th><th>BIT NAME</th><th>R/W</th><th>BIT DESCRIPTION</th></tr>
<tr><td>14</td><td>LEDMode</td><td>R/W</td><td>Light Emitting Diode Mode. LEDMode is used to control the LED signal pin functionality. When LEDMode is a logic 0 the LED signal pins operate in LED mode 0. When LEDMode is a logic 1 the LED signal pins operate in LED mode 1
Note, when LED signals alternate between logic 1/0, they alternate over a 41.89ms period, where the logic 0 (or LED ON) state persists for 5.24ms and the logic 1 (or LED OFF) state persists for 36.65ms..
<table>
<tr><th>LED SIGNAL PIN</th><th>MODE 0</th><th>MODE 1</th></tr>
<tr><td>EA2/LEDP-WRN</td><td>Continuous logic 0 when power is applied. Alternating logic 1/0 when frame transmission in progress.</td><td>Continuous logic 0 when power is applied.</td></tr>
<tr><td>EA1/ LEDLNK1000N</td><td>Continuous logic 0 when Ethernet link is valid. Alternating logic 1/0 when frame reception in progress.</td><td>Continuous logic 0 when Ethernet link is valid. Alternating logic 1/0 when frame reception or transmission in progress.</td></tr>
<tr><td>EA0/LED-DPLXN</td><td>Continuous logic 0 when the Rio is configured for full duplex operation. Alternating logic 1/0 when the Rio detects a collision.</td><td>Continuous logic 0 when the Rio is configured for full duplex operation. Continuous logic 1 when the Rio is configured for half duplex operation.</td></tr>
</table></td></tr>
<tr><td>15</td><td>RstOutPolarity</td><td>R/W</td><td>Reset Output Polarity. RstOutPolarity affects the polarity of the RSTOUT signal. When RstOutPolarity is a logic 1 the RSTOUT signal will assert HIGH to indicate a reset is in progress. When RstOutPolarity is a logic 0 the RSTOUT signal will assert LOW to indicate a reset is in progress.</td></tr>
<tr><td>16</td><td>GlobalReset</td><td>W</td><td>Global Reset. When GlobalReset is a logic 1, the Rio resets all of the logic functions and registers specified by the DMA, FIFO, Network, Host, AutoInit, and RstOut bits (related to both the transmit and receive processes as applicable). GlobalReset is self-clearing, and requires that the Rio RMON Statistics, Ethernet MIB Statistics, and I/O Registers be re-initialized. The PCI Configuration Registers are not affected by GlobalReset.</td></tr>
</table>

BIT	BIT NAME	R/W	BIT DESCRIPTION
17	RxReset	W	Receive Reset. When RxReset is a logic 1 the Rio resets all of the receive logic functions and registers specified by the DMA, FIFO, Network, Host, and AutoInit bits. RxReset is self-clearing, and should not be used after initialization except to recover from receive errors such as a receive FIFO over run.
18	TxReset	W	Transmit Reset. When TxReset is a logic 1 the Rio resets all of the transmit logic functions and registers specified by the DMA, FIFO, Network, Host, and AutoInit bits. TxReset is self clearing, and is required to be used after a transmit underrun error.

<table>
<tr><th>BIT</th><th>BIT NAME</th><th>R/W</th><th>BIT DESCRIPTION</th></tr>
<tr><td>19</td><td>DMA</td><td>W</td><td>DMA Reset. DMA selects (when a logic 1) or excludes (when a logic 0) the Rio DMA functions and registers for/from reset based on the value of the GlobalReset, RxReset, and TxReset bits. The DMA bit is self-clearing.
<table>
<tr><th>DMA FUNCTION/ REGISTER</th><th>RESET ON TXRESET</th><th>RESET ON RXRESET</th><th>RESET ON GLOBALRESET</th></tr>
<tr><td>Transmit DMA logic</td><td>x</td><td></td><td>x</td></tr>
<tr><td>Receive DMA logic</td><td></td><td>x</td><td>x</td></tr>
<tr><td>TFDListPtr register</td><td>x</td><td></td><td>x</td></tr>
<tr><td>RFDListPtr register</td><td></td><td>x</td><td>x</td></tr>
<tr><td>TxDMAComplete bit in the DMACtrl register</td><td>x</td><td></td><td>x</td></tr>
<tr><td>RxDMAComplete bit in the DMACtrl register</td><td></td><td>x</td><td>x</td></tr>
<tr><td>TxDMAInProg bit in the DMACtrl register</td><td>x</td><td></td><td>x</td></tr>
<tr><td>RxEarlyDisable bit in the DMACtrl register</td><td></td><td>x</td><td>x</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>BIT</th><th>BIT NAME</th><th>R/W</th><th>BIT DESCRIPTION</th></tr>
<tr><td>20</td><td>FIFO</td><td>W</td><td>FIFO Reset. FIFO selects (when a logic 1) or excludes (when a logic 0) the Rio FIFO functions and registers for/from reset based on the value of the GlobalReset, RxReset, and TxReset bits. The FIFO bit is self-clearing.
<table>
<tr><th>FIFO FUNCTION/ REGISTER</th><th>RESET ON TXRESET</th><th>RESET ON RXRESET</th><th>RESET ON GLOBALRESET</th></tr>
<tr><td>Transmit FIFO logic</td><td>x</td><td></td><td>x</td></tr>
<tr><td>Receive FIFO logic</td><td></td><td>x</td><td>x</td></tr>
<tr><td>TxStart-Thresh register</td><td>x</td><td></td><td>x</td></tr>
<tr><td>RxEarlyThresh register</td><td></td><td>x</td><td>x</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>BIT</th><th>BIT NAME</th><th>R/W</th><th>BIT DESCRIPTION</th></tr>
<tr><td>21</td><td>Network</td><td>W</td><td>Network Reset. Network selects (when a logic 1) or excludes (when a logic 0) the Rio network functions and registers for/ from reset based on the value of the GlobalReset, RxReset, and TxReset bits. The Network bit is self-clearing.
<table>
<tr><th>NETWORK FUNCTION/ REGISTER</th><th>RESET ON TXRESET</th><th>RESET ON RXRESET</th><th>RESET ON GLOBALRESET</th></tr>
<tr><td>Transmit network interface logic</td><td>x</td><td></td><td>x</td></tr>
<tr><td>Receive network interface logic</td><td></td><td>x</td><td>x</td></tr>
<tr><td>Transmit MAC logic</td><td>x</td><td></td><td>x</td></tr>
<tr><td>Receive MAC logic</td><td></td><td>x</td><td>x</td></tr>
<tr><td>TxStatus register</td><td>x</td><td></td><td>x</td></tr>
<tr><td>ReceiveMode register</td><td></td><td>x</td><td>x</td></tr>
<tr><td>All RMON Statistics and Ethernet MIB Statistics registers</td><td></td><td>x</td><td>x</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>BIT</th><th>BIT NAME</th><th>R/W</th><th>BIT DESCRIPTION</th></tr>
<tr><td>22</td><td>Host</td><td>W</td><td>Host Reset. Host selects (when a logic 1) or excludes (when a logic 0) the Rio host bus interface logic functions and registers for/from reset based on the value of the GlobalReset bit. The Host bit is self-clearing.
<table>
<tr><th>HOST FUNCTION/ REGISTER</th><th>RESET ON GLOBALRESET</th></tr>
<tr><td>Host bus interface logic</td><td>x</td></tr>
<tr><td>IntStatus register</td><td>x</td></tr>
<tr><td>IntEnable register</td><td>x</td></tr>
<tr><td>Countdown register</td><td>x</td></tr>
<tr><td>Transmit and receive DMA functions</td><td>x</td></tr>
</table></td></tr>
<tr><td>23</td><td>AutoInit</td><td>W</td><td>Automatic Initilization Reset. AutoInit selects (when a logic 1) or excludes (when a logic 0) the Rio auto-initialization logic function for/from re-loading Rio parameters from an EEPROM based on the value of the GlobalReset bit. The AutoInit bit is self-clearing.</td></tr>
<tr><td>24</td><td>RstOut</td><td>W</td><td>Reset Out Assert. RstOut selects (when a logic 1) or excludes (when a logic 0) the Rio RSTOUT signal for/from assertion (as determined by the RstOutPolarity bit) based on the value of the GlobalReset bit. The RstOut bit is self-clearing.</td></tr>
<tr><td>25</td><td>InterruptRequest</td><td>W</td><td>Interrupt Request. When InterruptRequest is a logic 1, the IntRequested bit of the IntStatus register is set to a logic 1. InterruptRequest is self-clearing.</td></tr>
<tr><td>26</td><td>ResetBusy</td><td>R</td><td>Reset Busy. When ResetBusy is a logic 1 a reset process is in progress. After asserting a reset using the GlobalReset, RxReset, or TxReset bits, the ResetBusy bit must be polled (or periodically read) until it is a logic 0 indicating the reset operation is complete.</td></tr>
<tr><td>31..27</td><td>Reserved</td><td>N/A</td><td>Reserved for future use. Write as zero, ignore on read.</td></tr>
</table>

11.10.5.2 Countdown

Class..... I/O Registers, Interrupt
I/O Base Address ..... IoBaseAddress register value
Memory Base Address . MemBaseAddress register value
Address Offset..... 0x54
Default Value 0x00000000
Access Rule..... Double Word
Width ..... 32 bits

Countdown is a programmable, decrementing counter which will generate an interrupt upon expiration (reaching a value of 0x0000).

BIT	BIT NAME	R/W	BIT DESCRIPTION
15..0	Count	R/W	Count. Count is the current value of Countdown register. When Count reaches 0x0000, it continues to decrement, wrapping to 0xFFFF.
23..16	Reserved	N/A	Reserved for future use. Write as zero, ignore on read.
24	CountdownSpeed	R/W	Count Down Speed. When operating at 1000Mbps and CountdownSpeed is a logic 0, the interval between decrements of the Countdown register is 3200 ns. When operating at 1000Mbps and CountdownSpeed is a logic 1, the interval between decrements of the Countdown register is 320 ns. When operating at 100Mbps or 10Mbps, the state of CountdownSpeed has no effect on the Countdown register interval. When operating at 100Mbps or 10Mbps, the interval between decrements of the Countdown register is 3200 ns.
25	CountdownMode	R/W	Count Down Mode. When CountdownMode is a logic 0, the Countdown register begins decrementing when a non-zero value is written to the Countdown register. When CountdownMode is a logic 1, the Countdown register begins decrementing after the TxDMAComplete bit in the IntStatus register is a logic 1.
26	CountdownIntEnabled	R	Count Down Interrupt Enabled. When CountdownIntEnabled is a logic 0, expiration of the Countdown timer will not set the IntRequested bit of the IntStatus register. When CountdownIntEnabled is a logic 1, expiration of the Countdown register will set the IntRequested bit in the IntStatus register. A logic 0 is written to CountdownIntEnabled when the IntRequested bit in the IntStatus register is a logic 1, or when a value of 0x0000 is written into the Count field. A logic 1 is written to CountdownIntEnabled when a non-zero value is written into the Count field.
31..27	Reserved	N/A	Reserved for future use. Write as zero, ignore on read.

11.10.5.3 DebugCtrl

Class..... I/O Registers, Control and Status
I/O Base Address ..... IoBaseAddress register value
Memory Base Address . MemBaseAddress register value
Address Offset..... 0x2C
Default Value 0x000C
Access Rule..... Word
Width ..... 16 bits

BIT	BIT NAME	R/W	BIT DESCRIPTION
0	GPIO0Ctrl	R/W	General Purpose Input Output 0 Control. GPIO0Ctrl specifies the direction of the Rio EA6/GPIO0 signal. When GPIO0Ctrl is a logic 0, the Rio EA6/GPIO0 signal is an input. When GPIO0Ctrl is a logic 1, the Rio EA6/GPIO0 signal is an output.
1	GPIO1Ctrl	R/W	General Purpose Input Output 1 Control. GPIO1Ctrl specifies the direction of the Rio EA7/GPIO1 signal. When GPIO1Ctrl is a logic 0, the Rio EA7/GPIO1 signal is an input. When GPIO1Ctrl is a logic 1, the Rio EA7/GPIO1 signal is an output.
2	GPIO0	R/W	General Purpose Input Output 0. GPIO0 represents the value of the Rio EA6/GPIO0 signal regardless of the state of GPIO0Ctrl. If GPIO0Ctrl is a logic 0, writes to GPIO0 are ignored.
3	GPIO1	R/W	General Purpose Input Output 1. GPIO1 represents the value of the Rio EA7/GPIO1 signal regardless of the state of GPIO1Ctrl. If GPIO1Ctrl is a logic 0, writes to GPIO1 are ignored.
4	dbDisableDnHalt	R/W	For testing purposes only.
5	dbDisableUpHalt	R/W	For testing purposes only.
6	dbSpeedSel	R/W	Speed Select determines the method Rio uses to change operation speed. If dbSpeedSel is a logic 0, the Rio speed of operation is set automatically (and indicated by the LinkSpeed field within the PhyCtrl register). If dbSpeedSel is a logic 1, the Rio speed of operation is set via dbSpeed.

<table>
<tr><th>BIT</th><th>BIT NAME</th><th>R/W</th><th>BIT DESCRIPTION</th></tr>
<tr><td>8..7</td><td>dbSpeed</td><td>R/W</td><td>Software Speed allows for manual setting of the Rio speed (10, 100, or 1000Mbps operation) based on the state of dbSpeedSel.
<table>
<tr><th>BIT8</th><th>BIT7</th><th>LINK SPEED/STATUS</th></tr>
<tr><td>0</td><td>0</td><td>Link down (default).</td></tr>
<tr><td>0</td><td>1</td><td>Link up at 10Mbps.</td></tr>
<tr><td>1</td><td>0</td><td>Link up at 100Mbps.</td></tr>
<tr><td>1</td><td>1</td><td>Link up at 1000Mbps.</td></tr>
</table></td></tr>
<tr><td>9</td><td>dbFrCurDoneAck</td><td>R/W</td><td>For testing purposes only.</td></tr>
<tr><td>10</td><td>dbFrcSpd1000</td><td>R/W</td><td>Force Speed to 1000Mbps when a logic 1 will cause the register access logic to behave as if the AsicClk speed is 62.5MHz.</td></tr>
<tr><td>15..11</td><td>Reserved</td><td>N/A</td><td>Reserved for future use. Write as zero, ignore on read.</td></tr>
</table>

11.10.5.4 DMACtrl

Class..... I/O Registers, DMA
I/O Base Address ..... IoBaseAddress register value
Memory Base Address . MemBaseAddress register value
Address Offset..... 0x00
Default Value 0x00000000
Access Rule..... Word or Double Word
Width ..... 32 bits

BIT	BIT NAME	R/W	BIT DESCRIPTION
0	Reserved (RxDMAHalted)	R	Reserved for future use. Write as zero, ignore on read. (This read-only bit is set whenever RxDMA is halted by setting the RxDMAHalt bit or an implicit halt due to fetching a RFD with RxDMAComplete in ReceiveFrameStatus already set. Cleared by setting the RxDMAResume bit.)
1	Reserved (RxDMAHalt)	N/A	Reserved for future use. Write as zero, ignore on read. (Whenever this bit is set, the RxDMA is halted. This bit is self-clearing and writing a 0 into this bit is ignored.)
2	Reserved (RxDMAResume)	N/A	Reserved for future use. Write as zero, ignore on read. (Whenever this bit is set, the RxDMA is resumed. This bit is self-clearing and writing a 0 into this bit is ignored.)
3	RxDMAComplete	R	Receive DMA Complete. RxDMAComplete is identical to the RxDMAComplete bit in the IntStatus register. RxDMAComplete is a logic 0 when the receive DMA process for a received Ethernet frame begins. RxDMAComplete is a logic 0 also when the RxDMAComplete interrupt is acknowledged.
4	RxDMAPollNow	W	Receive DMA Poll Now. When RxDMAPollNow is a logic 1 a “poll now” command is issued to the receive DMA process. The “poll now” receive DMA command resets the receive DMA poll timer which forces the current RFD RFDNextPtr and RFS fields to be read. RxDMAPollNow is a self clearing.
7..5	Reserved	N/A	Reserved for future use. Write as zero, ignore on read.
8	Reserved (TxDMAHalted)	N/A	Reserved for future use. Write as zero, ignore on read. (This read-only bit is set whenever TxDMA is halted by setting the TxDMAHalt bit. The host system should wait for this bit to be set before modifying TFD lists. Cleared by setting the TxDMAResume bit.)
9	Reserved (TxDMAHalt)	N/A	Reserved for future use. Write as zero, ignore on read. (Whenever this bit is set, the TxDMA is halted. This bit is self-clearing and writing a 0 into this bit is ignored.)

BIT	BIT NAME	R/W	BIT DESCRIPTION
10	Reserved (TxDMAResume)	N/A	Reserved for future use. Write as zero, ignore on read. (Whenever this bit is set, the TxDMA is resumed. This bit is self-clearing and writing a 0 into this bit is ignored.)
11	TxDMAComplete	R	Transmit DMA Complete. TxDMAComplete is identical to the TxDMAComplete bit in the IntStatus register. TxDMAComplete is a logic 0 when the TxDMAComplete interrupt is acknowledged.
12	TxDMAPollNow	W	Transmit DMA Poll Now. When TxDMAPollNow is a logic 1 a “poll now” command is issued to the transmit DMA process. The “poll now” transmit DMA command resets the transmit DMA poll timer which forces the current TFD TFDNextPtr and TFC fields to be read. TxDMAPollNow is a self clearing.
13	Reserved	N/A	Reserved for future use. Write as zero, ignore on read.
14	Reserved (TxDMACmplReq)	N/A	Reserved for future use. Write as zero, ignore on read. (This read-only bit is set to the value from the TxDMAIndicate field in the TFC of the current TFD.)
15	TxDMAInProg	R	Transmit DMA In Progress. When TxDMAInProg is a logic 1 a transmit DMA operation is in progress. TxDMAInProg is primarily used for underrun recovery processes which must wait for TxDMAInProg to be a logic 0 before issuing a transmit reset via the TxReset bit of the AsicCtrl register.
16	RxEarlyDisable	R/W	Receive Early Disable. RxEarlyDisable is used in conjunction with the RxEarlyThresh register to determine the start of receive DMA processes on received Ethernet frames. If a receive DMA bus-master arbitration does not begin based on the conditions of the RxEarlyThresh register, then the start of the receive DMA process depends on the state of RxEarlyDisable. When RxEarlyDisable is a logic 0, the receive DMA process will begin bus-master arbitration when 60 bytes of the received Ethernet frame are present in the receive FIFO. When RxEarlyDisable is a logic 1, the receive DMA process will begin bus-master arbitration when the entire frame is present in the receive FIFO.
17	Reserved (RxDMAOverrun-Frame)	N/A	Reserved for future use. Write as zero, ignore on read. (This read/write bit, when clear (the default), causes the RxDMA engine to discard receive overrun frames without transferring them to system memory. When this bit is set, the RxDMA engine keeps and transfers overrun frames.)
18	MWIDisable	R/W	Memory Write Invalidate Disable. When MWIDisable is a logic 1, the bus master logic will not use the PCI Memory Write Invalidate (MWI) command.

BIT	BIT NAME	R/W	BIT DESCRIPTION
19	TxWriteBackDis-able	R/W	Transmit Write Back Disable. When TxWriteBackDisable is a logic 1 the Rio will not update the TFDDone bit and will not poll the TFDDone bit within the TFC field following a transmit DMA operation. When TxWriteBackDisable is a logic 0 the Rubicon will update the TFDDone bit for and will poll the TFDDone bit within the TFC field following each transmit DMA operation.
22..20	TxBurstLimit	R/W	Transmit Burst Limit. TxBurstLimit indicates the number of TFD FragInfo fields the Rio will transfer at a time. The valid range of values is 0x1, 2, 3, 4, and 5. All non-valid values will be interpreted as the value 0x1. TxBurstLimit is a system dependent parameter which should be set such to a value corresponding to the number of fragments present in the majority of Ethernet frames. In many systems, most Ethernet frames consist of 3 or less fragments, which implies a TxBurstLimit of 3.
29..23	Reserved	N/A	Reserved for future use. Write as zero, ignore on read.
30	TargetAbort	R	Target Abort. TargetAbort is a logic 1 when the Rio experiences a target abort sequence when operating as a bus master. TargetAbort indicates a fatal error and must be cleared before further transmit or receive DMA operation can proceed. TargetAbort is cleared via the AsicCtrl register GlobalReset/ Host bits.
31	MasterAbort	R	Master Abort. MasterAbort is a logic 1 when the Rio experiences a master abort sequence when operating as a bus master. MasterAbort indicates a fatal error, and must be cleared before further transmit or receive DMA operation can proceed. MasterAbort is cleared via the AsicCtrl register GlobalReset/ Host bits.

11.10.5.5 EepromCtrl

Class..... I/O Registers, Control and Status

I/O Base Address ..... IoBaseAddress register value

Memory Base Address . MemBaseAddress register value

Address Offset..... 0x4A

Default Value 0x0000

Access Rule..... Word

Width ..... 16 bits

EepromCtrl provides the host with a method for issuing commands to the Rio's serial EEPROM controller. Individual 16-bit word locations within the EEPROM may be written, read or erased. Also, the EEPROM's WriteEnable, WriteDisable, EraseAll and WriteAll commands can be issued. Two-bit opcodes and 8-bit addresses are written to this register to cause the Rio to carry out the desired EEPROM command. If data is to written to the EEPROM, the 16-bit data word must be written to EepromData by the host prior to issuing the associated write command. Similarly, if data is to be read from the EEPROM, the read data will be available via EepromData register. The EEPROM is a particularly slow device. It is important that the host wait until the EepromBusy bit is a logic 0 before issuing a command to EepromCtrl.

BIT	BIT NAME	R/W	BIT DESCRIPTION
7..0	EepromAddress	R/W	EEPROM Address. EepromAddress identifies one of the 256 sixteen-bit words to be the target for the ReadRegister, WriteRegister and EraseRegister EEPROM commands. Bits 7 and 6 are further define EEPROM sub-commands based on the value of EepromOpcode. Bits 7 and 6 of EepromAddress define a sub-command only if EepromOpcode is 0x0..
9..8	EepromOpcode	R/W	EEPROM Operation Code. EepromOpcode specifies one of three individual commands and a single group of four sub-commands.

BIT 7	BIT 6	SUB-COMMAND
0	0	WriteDisable
0	1	WriteAII
1	0	EraseAII
1	1	WriteEnable

BIT 9	BIT 8	OPCODE COMMAND
0	0	Write Enable/Disable & Write/ Erase All sub-commands
0	1	WriteRegister
1	0	ReadRegister
1	1	EraseRegister

BIT	BIT NAME	R/W	BIT DESCRIPTION
14..10	Reserved	N/A	Reserved for future use. Write as zero, ignore on read.
15	EepromBusy	R	EEPROM Busy. EepromBusy is a logic 1 during the execution of EEPROM commands. Further commands should not be issued to EepromCtrl nor should data be read from EepromData while EepromBusy is a logic 1.

11.10.5.6 EepromData

Class..... I/O Registers, Control and Status
I/O Base Address ..... IoBaseAddress register value
Memory Base Address . MemBaseAddress register value
Address Offset..... 0x48
Default Value 0x0000
Access Rule..... Word
Width ..... 16 bits

BIT	BIT NAME	R/W	BIT DESCRIPTION
15..0	EepromData	R/W	EepromData is a 16-bit data register for use with the adapter's serial EEPROM. Data from the EEPROM can be read by the host from EepromData register after the EepromBusy bit in the EepromCtrl register is a logic 0. Data to be written to the EEPROM is written to EepromData prior to issuing the write command to EepromCtrl.

11.10.5.7 ExpRomAddr

Class..... I/O Registers, Control and Status
I/O Base Address ..... IoBaseAddress register value
Memory Base Address . MemBaseAddress register value
Address Offset..... 0x4C
Default Value 0x00000000
Access Rule..... Double Word
Width ..... 32 bits

BIT	BIT NAME	R/W	BIT DESCRIPTION
15..0	ExpRomAddr	R/W	Expansion ROM Address. ExpRomAddr holds the address to be used for direct I/O accesses of the Expansion ROM through the ExpRomData port. To access a byte in the Expansion ROM, write the address of the byte to be accessed into ExpRomAddr. Then issue either a read or a write to ExpRom-Data. For reads, the ROM value will be returned by the read instruction. For writes, the new value will be programmed into the ROM upon completion of the write instruction.
31..16	Reserved	N/A	Reserved for future use. Write as zero, ignore on read.

11.10.5.8 ExpRomData

Class..... I/O Registers, Control and Status
I/O Base Address ..... IoBaseAddress register value
Memory Base Address . MemBaseAddress register value
Address Offset..... 0x50
Default Value 0x00
Access Rule..... Byte
Width ..... 8 bits

BIT	BIT NAME	R/W	BIT DESCRIPTION
7..0	ExpRomData	R/W	Expansion ROM Data. ExpRomData is the data port for performing direct I/O byte-wide accesses of the Expansion ROM. A read of ExpRomData returns the ROM byte value from the location specified by ExpRomAddr. A write to ExpRomData causes the write data to be programmed into the ROM location specified by ExpRomAddr.

11.10.5.9 FIFOCtrl

Class..... I/O Registers, Control and Status
I/O Base Address ..... IoBaseAddress register value
Memory Base Address . MemBaseAddress register value
Address Offset..... 0x38
Default Value 0x0000
Access Rule..... Word
Width ..... 16 bits

The bits in this register provide various control and indications of TxFIFO and RxFIFO diagnostic.

BIT	BIT NAME	R/W	BIT DESCRIPTION
0	RAMTestMode	R/W	Random Access Memory Test Mode. When RAMTestMode is a logic 1 indicates the FIFO RAM is in the test mode. Random Access Memory Test Mode is also entered if bit 3 of the ED bus is a logic 0 during reset.
8..1	Reserved	N/A	Reserved for future use. Write as zero, ignore on read.
9	Reserved (RxOverrunFrame)	N/A	Reserved for future use. Write as zero, ignore on read. (This read/write bit determines how the Rio handles receive overrun frames. The default is zero, which causes the Rio to discard all overrun frames. Setting this bit causes the Rio to keep and make visible all overrun frames that have been made visible to the host, so that they may be inspected for diagnostic purposes.)
10	Reserved	N/A	Reserved for future use. Write as zero, ignore on read.
11	Reserved (RxFIFOFull)	N/A	Reserved for future use. Write as zero, ignore on read. (This read-only bit is set when the RxFIFO is full. This bit does not in itself indicate an overrun condition. However, if more data is received while this bit is set, an overrun will occur. This bit is informational in nature only. This bit is cleared as soon as the RxFIFO is no longer full.)
13..12	Reserved	N/A	Reserved for future use. Write as zero, ignore on read.
14	Transmitting	R	Transmitting. When Transmitting is a logic 1 indicates the MAC logic is transmitting or waiting to transmit (deferring).
15	Receiving	R	Receiving. When Receiving is a logic 1 indicates the Rio is receiving a frame into the receive FIFO.

11.10.5.10 FlowOffThresh

Class..... I/O Registers, Control and Status
I/O Base Address ..... IoBaseAddress register value
Memory Base Address . MemBaseAddress register value
Address Offset..... 0x3C
Default Value 0x0000
Access Rule..... Word
Width ..... 16 bits

BIT	BIT NAME	R/W	BIT DESCRIPTION
10..0	FlowOffThresh	R/W	Flow Off Threshold. FlowOffThresh sets a watermark within the receive FIFO. If the amount of occupied space (measured in 16 byte increments) within the receive FIFO falls below the value set in the FlowOffThresh register, a PAUSE MAC Control frame with pause_time set to 0x0000 is transmitted. Only one PAUSE frame is transmitted when the FlowOffThresh watermark is crossed (see section 11.7.3).
15..11	Reserved	N/A	Reserved for future use. Write as zero, ignore on read.

11.10.5.11 FlowOnThresh

Class..... I/O Registers, Control and Status
I/O Base Address ..... IoBaseAddress register value
Memory Base Address . MemBaseAddress register value
Address Offset..... 0x3E
Default Value 0x07FF
Access Rule..... Word
Width ..... 16 bits

BIT	BIT NAME	R/W	BIT DESCRIPTION
10..0	FlowOnThresh	R/W	Flow On Threshold. FlowOnThresh sets a watermark within the receive FIFO. If the amount of occupied space (measured in 16 byte increments) within the receive FIFO exceeds the value set in the FlowOnThresh register, a PAUSE MAC Control frame with pause_time set to 0xFFFF (or 65,535 * 512ns = 33.553ms) is transmitted. Only one PAUSE frame is transmitted when the FlowOnThresh watermark is crossed (see section 11.7.3).
15..11	Reserved	N/A	Reserved for future use. Write as zero, ignore on read.

11.10.5.12 HashTable

Class..... I/O Registers, Control and Status
I/O Base Address ..... IoBaseAddress register value
Memory Base Address . MemBaseAddress register value
Address Offset..... 0x8C
Default Value 0x0000000000000000
Access Rule..... Double Word
Width ..... 64 bits

BIT	BIT NAME	R/W	BIT DESCRIPTION
63..0	HashTable	R/W	Hash Table holds a 64-bit value used for selectively receiving multicast frames. Setting the ReceiveMulticastHash bit in the ReceiveMode register enables the filtering mechanism. The hash table is cleared upon reset, and must be properly set by the host. The Rio applies a cyclic-redundancy-check (the same CRC used to calculate the frame data FCS) to the destination address of all incoming multicast frames (Ethernet frames with multicast bit set in their destination address field). The least significant 6 bits of the CRC result are used as an addressing index into the hash table. If the HashTable bit addressed by the index is a logic 1, the frame is accepted by the Rio and transferred to higher layers. If the HashTable bit addressed by the index is a logic 0, the frame is discarded.

11.10.5.13 IntEnable

Class..... I/O Registers, Interrupt
I/O Base Address ..... IoBaseAddress register value
Memory Base Address . MemBaseAddress register value
Address Offset..... 0x5C
Default Value 0x0000
Access Rule..... Word
Width ..... 16 bits

IntEnable enables individual interrupts as specified in the IntStatus register. Setting a bit in IntEnable will allow the corresponding interrupt source to generate an interrupt on the PCI bus. IntEnable is cleared by a read of IntStatusAck register.

BIT	BIT NAME	R/W	BIT DESCRIPTION
0	Reserved	N/A	Reserved for future use. Write as zero, ignore on read.
1	EnHostError	R/W	Enables the HostError bit in the IntStatus register to generate an interrupt.
2	EnTxComplete	R/W	Enables the TxComplete bit in the IntStatus register to generate an interrupt.
3	EnMACControl-Frame	R/W	Enables the MACControlFrame bit in the IntStatus register to generate an interrupt.
4	EnRxComplete	R/W	Enables the RxComplete bit in the IntStatus register to generate an interrupt.
5	EnRxEarly	R/W	Enables the RxEarly bit in the IntStatus register to generate an interrupt.
6	EnInRequested	R/W	Enables the IntRequested bit in the IntStatus register to generate an interrupt.
7	EnUpdateStats	R/W	Enables the UpdateStats bit in the IntStatus register to generate an interrupt.
8	EnLinkEvent	R/W	Enables the LinkEvent bit in the IntStatus register to generate an interrupt.
9	EnTxDMACom-plete	R/W	Enables the TxDMAComplete bit in the IntStatus register to generate an interrupt.
10	EnRxDMACom-plete	R/W	Enables the RxDMAComplete bit in the IntStatus register to generate an interrupt.
11	EnRFDListEnd	R/W	Enables the RFDListEnd bit in the IntStatus register to generate an interrupt.
12	EnRxDMAPriority	R/W	Enables the RxDMAPriority bit in the IntStatus register to generate an interrupt.

BIT	BIT NAME	R/W	BIT DESCRIPTION
15..13	Reserved	N/A	Reserved for future use. Write as zero, ignore on read.

11.10.5.14 IntStatus

Class..... I/O Registers, Interrupt
I/O Base Address ..... IoBaseAddress register value
Memory Base Address . MemBaseAddress register value
Address Offset..... 0x5E
Default Value 0x0000
Access Rule..... Word
Width ..... 16 bits

IntStatus indicates the source of interrupts and indications on the Rio.

BIT	BIT NAME	R/W	BIT DESCRIPTION
0	InterruptStatus	R/W	Interrupt Status. InterruptStatus is a logic 1 when the Rio is driving the bus interrupt signal. It is a logical OR of all the interrupt-causing bits after they have been filtered through the IntEnable register.
1	HostError	R/W	Host Error. HostError is a logic 1 when a catastrophic error related to the bus interface occurs. The conditions which set HostError are PCI target abort and PCI master abort. HostError is cleared by setting the GlobalReset and Host bits in the AsicCtrl register. If HostError is a logic 1, and the EnHostError bit in the IntEnable register is also a logic 1, an interrupt will be generated.
2	TxComplete	R/W	Transmit Complete. TxComplete is a logic 1 when a frame (whose TFC field TxIndicate bit is a logic 1) has been successfully transmitted or for any frame that experiences an error during transmit. The TxComplete interrupt is acknowledged by reading the TxStatus register. If TxComplete is a logic 1, and the EnTxComplete bit in the IntEnable register is also a logic 1, an interrupt will be generated.
3	MACControlFrame	R/W	MAC Control Frame. MACControlFrame is a logic 1 when a MAC Control frame has been received by the Rio. MACControlFrame is acknowledged by writing a logic 1 to MACControlFrame. If MACControlFrame is a logic 1, and the EnMACControlFrame bit in the IntEnable register is also a logic 1, an interrupt will be generated.

BIT	BIT NAME	R/W	BIT DESCRIPTION
4	RxComplete	R/W	Receive Complete. RxComplete is a logic 1 when one or more entire frames have been received into the receive FIFO. RxComplete is automatically acknowledged by the RxDMA Logic as it transfers frames. If RxComplete is a logic 1, and the EnRxComplete bit in the IntEnable register is also a logic 1, an interrupt will be generated. The host system should disable the RxComplete interrupt and mask RxComplete when reading IntStatus.
5	RxEarly	R/W	Receive Early. RxEarly is a logic 1 when the number of bytes of a frame which is being received is greater than the value of the RxEarlyThresh register. When the frame has been completely received by the Rio, RxEarly will be a logic 0 and RxComplete will be a logic 1. RxEarly is acknowledged by writing a logic 1 to RxEarly. If RxEarly is a logic 1, and the EnRxEarly bit in the IntEnable register is also a logic 1, an interrupt will be generated.
6	IntRequested	R/W	Interrupt Requested. IntRequested is a logic 1 when the host system requested an interrupt by setting the InterruptRequest bit in the AsicCtrl register or when the Countdown register expires. IntRequested is acknowledged by writing a logic 1 to IntRequested. If IntRequested is a logic 1, and the EnInRequested bit in the IntEnable register is also a logic 1, an interrupt will be generated.
7	UpdateStats	R/W	Update Statistics. UpdateStats is a logic 1 when one or more of the statistics counters (RMON Statistics, Ethernet MIB Statistics, or I/O Registers Statistics) is nearing an overflow condition (typically 75% of the statistic register's maximum value). The host system should respond to an UpdateStats interrupt by reading all of the statistic registers, thereby acknowledging and clearing UpdateStats bit. If UpdateStats is a logic 1, and the EnUpdateStats bit in the IntEnable register is also a logic 1, an interrupt will be generated.
8	LinkEvent	R/W	Link Event. LinkEvent is a logic 1 when there is a transition of PHYLNK10N signal (i.e. a change in the link status of the PHY device). LinkEvent is acknowledged by writing a logic 1 to LinkEvent. If LinkEvent is a logic 1, and the EnLinkEvent bit in the IntEnable register is also a logic 1, an interrupt will be generated.

BIT	BIT NAME	R/W	BIT DESCRIPTION
9	TxDMAComplete	R/W	Transmit DMA Complete. TxDMAComplete is a logic 1 when a frame transfer via transmit DMA has completed, and the TFD in associated with the transmit DMA operation had the it's TFC TxDMAIndicate bit set to a logic 1. TxDMAComplete is acknowledged by writing a logic 1 to TxDMAComplete. To determine which frame(s) have been transferred to the Rio via transmit DMA, the host may examine the TFDDone bits in the TFC fields of each TFD in the TFDList (see section 11.3.1 for more details). If TxDMAComplete is a logic 1, and the EnTxDMAComplete bit in the IntEnable register is also a logic 1, an interrupt will be generated.
10	RxDMAComplete	R/W	Receive DMA Complete. RxDMAComplete is a logic 1 when a frame transfer via receive DMA has completed. RxDMAComplete is acknowledged by writing a logic 1 to RxDMAComplete. Based on the configuration of the RxDMAIntCtrl register, RxDMAComplete interrupts may occur for each RFD which is transferred to the Rio, or only after multiple RFDs have been transferred (see section 11.3.4 for more details). If RxDMAComplete is a logic 1, and the EnRxDMAComplete bit in the IntEnable register is also a logic 1, an interrupt will be generated.
11	RFDListEnd	R/W	RFD List End. RFDListEnd is a logic 1 when the RxDMA Logic has reached the end of the RFD list (as indicated by a logic 1 in the RFDDone bit in the RFS field, and a 0x0000000000000000 value in the RFDNextPtr field of the RFD). RFDListEnd is acknowledged by writing a 1 to RFDListEnd. If RFDListEnd is a logic 1, and the EnRFDListEnd bit in the IntEnable register is also a logic 1, an interrupt will be generated.
12	RxDMAPriority	R/W	Receive DMA Priority. RxDMAPriority is a logic 1 when an Ethernet frame has been received with a priority tag of equal or higher priority than the value set in the PriorityThresh field of the RxDMAIntCtrl register. RxDMAPriority is acknowledged by writing a 1 to RxDMAPriority. If RxDMAPriority is a logic 1, and the EnRxDMAPriority bit in the IntEnable register is also a logic 1, an interrupt will be generated.
15..13	Reserved	N/A	Reserved for future use. Write as zero, ignore on read.

11.10.5.15 IntStatusAck

Class..... I/O Registers, Interrupt

I/O Base Address ..... IoBaseAddress register value

Memory Base Address . MemBaseAddress register value

Address Offset..... 0x5A

Default Value 0x0000

Access Rule..... Word

Width ..... 16 bits

IntStatusAck is another version of the IntStatus register, having the same bit definition as IntStatus, but providing additional functionality to reduce the number of I/O operations required to perform common tasks related to interrupt handling. In addition to returning the IntStatus value for the specified interrupt, when read IntStatusAck also acknowledges the TxDMAComplete, RxDMAComplete, RFDListEnd, RxEarly, IntRequested, MACControlFrame, RxDMAPriority, and LinkEvent bits within the IntStatus register (if they are set), and clears the IntEnable register preventing subsequent events from generating an interrupt.

BIT	BIT NAME	R/W	BIT DESCRIPTION
0	InterruptStatus	R/W	See description in IntStatus register description.
1	HostError	R/W	See description in IntStatus register description.
2	TxComplete	R/W	See description in IntStatus register description.
3	MACControlFrame	R/W	See description in IntStatus register description.
4	RxComplete	R/W	See description in IntStatus register description.
5	RxEarly	R/W	See description in IntStatus register description.
6	IntRequested	R/W	See description in IntStatus register description.
7	UpdateStats	R/W	See description in IntStatus register description.
8	LinkEvent	R/W	See description in IntStatus register description.
9	TxDMAComplete	R/W	See description in IntStatus register description.
10	RxDMAComplete	R/W	See description in IntStatus register description.
11	RFDListEnd	R/W	See description in IntStatus register description.
12	RxDMAPriority	R/W	See description in IntStatus register description.
15..13	Reserved	N/A	Reserved for future use. Write as zero, ignore on read.

11.10.5.16 IPCheckSumErrors

Class..... I/O Registers, Statistics
I/O Base Address ..... IoBaseAddress register value
Memory Base Address . MemBaseAddress register value
Address Offset..... 0xC2
Default Value 0x0000
Access Rule..... Word
Width ..... 16 bits

BIT	BIT NAME	R/W	BIT DESCRIPTION
15..0	IPChecksumErrors	R/W	IP Check Sum Errors is a count of received frames which contain IP datagrams, which fail the IP checksum as defined in RFC 791. IPCheckSumErrors will wrap around to zero after reaching 0xFFFF. An UpdateStats interrupt (UpdateStats bit within the IntStatus register) will occur when IPCheckSumErrors reaches a value of 0xC000. IPCheckSumErrors is enabled by writing a logic 1 to the StatisticsEnable bit in the MACCtrl register, and a logic 0 to the IPCheckSumErrors bit within the StatisticsMask register.

11.10.5.17 MACCtrl

Class..... I/O Registers, Control and Status
I/O Base Address ..... IoBaseAddress register value
Memory Base Address . MemBaseAddress register value
Address Offset..... 0x6C
Default Value 0x00000000
Access Rule..... Word, Double Word
Width ..... 32 bits

BIT	BIT NAME	R/W	BIT DESCRIPTION
1..0	IFSSelect	R/W	Inter-Frame Spacing. IFSSelect indicates the minimum number of bit times between the end of one Ethernet frame, and the beginning of another when the Rio is deferring after a collision with the Rio the last device to successfully acquire the network (in half duplex mode). By selecting a large value for IFSSelect, the Rio will become less “aggressive” on the network and may defer more often (preventing the Rio from “capturing” the network). The performance of the Rio may decrease as the IFSSelect value is increased from the standard value.
4..2	Reserved	N/A	Reserved for future use. Write as zero, ignore on read.
5	DuplexSelect	R/W	Duplex Select. DuplexSelect configures the Rio to function in half or full duplex mode. If DuplexSelect is a logic 0, the Rio operates in half duplex mode. If DuplexSelect is a logic 1, the Rio operates in full duplex mode. In full duplex mode the transmitter deference to receive traffic functionality is disabled, allowing simultaneous receive and transmit traffic. Full duplex operation has the side-effect of disabling the CarrierSenseErrors statistic register. Following a change to DuplexSelect, the Rio transmit and receive functions must be reset by setting the AsicCtrl register TxReset and RxReset bits both to a logic 1.

BIT1	BIT0	INTER-FRAME SPACING IN BIT TIMES
0	0	96 (802.3 standard value, and default)
0	1	1024
1	0	1792
1	1	4352

BIT	BIT NAME	R/W	BIT DESCRIPTION
6	Reserved (RcvLargeFrames)		Reserved for future use. Write as zero, ignore on read. (This bit determines the frame size at which the OversizedFrame error is generated for receive frames. When RcvLargeFrames is cleared, minimum OversizedFrame size is 1514 bytes. When RcvLargeFrames is set, minimum OversizedFrame size is 4491 bytes. (This value was the maximum FDDI frame size of 4500 bytes, subtracting bytes for fields that have no Ethernet equivalent.) The frame size at which an OversizedFrame error will be flagged includes the destination and source addresses, the type/length field, and the FCS field.)
7	TxFlowControlEnable	R/W	Transmit Flow Control Enable. When TxFlowControlEnable is a logic 1 the receive FIFO threshold values defined in FlowOnThresh and FlowOffThresh registers are used to determine automatic transmission of PAUSE frames by the Rio. It is the responsibility of the host system to assure that if TxFlowControlEnable is a logic 1, the MACCtrl register DuplexSelect bit must also be a logic 1.
8	RxFlowControlEnable	R/W	Receive Flow Control Enable. When RxFlowControlEnable is a logic 0 the Rio will treat all incoming frames (even PAUSE frames) as data, and will not take any action if a PAUSE frame is received. When RxFlowControlEnable is a logic 1, receive flow control is enabled and the Rio will act upon incoming flow control PAUSE frames. It is the responsibility of the host system to assure that if RxFlowControlEnable is a logic 1, the MACCtrl register DuplexSelect bit must also be a logic 1.
9	RcvFCS	R/W	Receive Frame Check Sequence. When RcvFCS is a logic 1 the receive frame's FCS field is passed to the host as part of the data in the receive FIFO. The state of RcvFCS does not affect the Rio's checking of the frame's FCS field and posting of the FCS error status. The value of RcvFCS should only be changed during initialization of the Rio.
10	FIFOLoopback	R/W	FIFO Loopback. When FIFOLoopback is a logic 1 the Rio will apply data from the output of the transmit FIFO directly to the input of the receive FIFO. When using FIFOLoopback, it is the host system's responsibility to ensure that the proper interframe spacing by not loading more than one transmit frame at a time into the transmit FIFO. Following a change to the FIFOLoopback bit, the Rio transmit and receive functions must be reset by setting the AsicCtrl register TxReset and RxReset bits both to a logic 1.

BIT	BIT NAME	R/W	BIT DESCRIPTION
11	MACLoopback	R/W	Media Access Control Loopback. When MACLoopback is a logic 1 the Rio will apply transmit data at the output of the Media Access Control (MAC) logic transmit interface to the MAC receive interface. Following a change to the MACLoopback bit, the Rio transmit and receive functions must be reset by setting the AsicCtrl register TxReset and RxReset bits both to a logic 1.
12	AutoVLANtagging	R/W	Automatic VLAN Tagging. When AutoVLANtagging is a logic 1 the Rio will apply a VLAN tag as specified by the VLANTag register to every Ethernet frame prior to transmission. A logic 0 indicates the Rio should not insert a VLAN tag.
13	AutoVLANuntagging	R/W	Automatic VLAN Tag Removal. When AutoVLANuntagging is a logic 1 the Rio will remove the VLAN tag (if any) in all received Ethernet frames. A logic 0 indicates the Rio should not remove VLAN tags from received frames.
15..14	Reserved	N/A	Reserved for future use. Write as zero, ignore on read.
16	CollisionDetect	R	Collision Detect. CollisionDetect provides a real-time indication of the state of the COL signal within the Rio.
17	CarrierSense	R	Carrier Sense. CarrierSense provides a real-time indication of the state of the CRS signal within the Rio.
18	Reserved (TxInProg)	N/A	Reserved for future use. Write as zero, ignore on read. (A real-time indication that a frame is being transmitted. This bit is used by drivers during underrun recovery to delay issuing a TxReset.)
19	Reserved (TxError)	N/A	Reserved for future use. Write as zero, ignore on read. (If a TxUnderrun occurs, this bit is set, indicating that the transmitter needs to be reset with the TxReset.)
20	Reserved	N/A	Reserved for future use. Write as zero, ignore on read.
21	StatisticsEnable	W	Statistics Enable. When StatisticsEnable is a logic 1 the Rio will increment the various statistics counters and registers as applicable. StatisticsEnable is self-clearing.
22	StatisticsDisable	W	Statistics Disable. When StatisticsDisable is a logic 1 the Rio will not increment any statistics registers. The values in the statistics registers will remain unchanged. StatisticsDisable is self-clearing.
23	StatisticsEnabled	R	Statistics Enabled. When StatisticsEnabled is a logic 1 the statistic registers will increment as applicable.
24	TxEnable	W	Transmit Enable. When TxEnable is a logic 1 the transmitter logic is enabled. TxEnable is self-clearing.

BIT	BIT NAME	R/W	BIT DESCRIPTION
25	TxDisable	W	Transmit Disable. When TxDisable is a logic 1 the transmitter logic is disabled. TxDisable is self-clearing.
26	TxEnabled	R	Transmit Enabled. When TxEnabled is a logic 1 the transmitter is enabled.
27	RxEnable	W	Receive Enable. When RxEnable is a logic 1 the receive logic is enabled. RxEnable is self-clearing.
28	RxDisable	W	Receive Disable. When RxDisable is a logic 1 the receive logic is disabled. RxDisable is self-clearing.
29	RxEnabled	R	Receive Enabled. When RxEnabled is a logic 1 the receive logic is enabled.
30	Paused	R	Paused. When Paused is a logic 1 a PAUSE MAC Control frame has been received and caused the transmit MAC logic to halt for the duration of the pause_time value specified in the PAUSE frame. Paused is cleared when MAC transmission resumes.
31	Reserved	N/A	Reserved for future use. Write as zero, ignore on read.

11.10.5.18 MaxFrameSize

Class..... I/O Registers, Control and Status
I/O Base Address ..... IoBaseAddress register value
Memory Base Address . MemBaseAddress register value
Address Offset..... 0x86
Default Value 0x05EA
Access Rule..... Word
Width ..... 16 bits

BIT	BIT NAME	R/W	BIT DESCRIPTION
13..0	MaxFrameSize	R/W	Maximum Frame Size indicates the maximum expected size (in bytes) of received Ethernet frames measured from the start of the Destination Address field, to the end of the Data field. If the number of bytes in a receive frame is equal to or greater than the value in the MaxFrameSize register (or the value in the MaxFrameSize register plus 4 bytes for received frames with VLAN tags), the RxOversizedFrame bit in the RFS field of the current RFD is a logic 1.
15..14	Reserved	N/A	Reserved for future use. Write as zero, ignore on read.

11.10.5.19 PhyCtrl

Class..... I/O Registers, Control and Status

I/O Base Address ..... IoBaseAddress register value

Memory Base Address . MemBaseAddress register value

Address Offset..... 0x76

Default Value 0x00

Access Rule..... Byte

Width ..... 8 bits

PhyCtrl contains control bits for the GMII Management Interface. The GMII Management Interface is used to access registers in a GMII PHY device or the Rio 1000BASE-X PCS management registers. The Management Interface is a two-wire serial interface connecting Rio to any GMII compliant PHY devices residing in the system. The host system operates the Management Interface by writing and reading bit patterns to PhyCtrl which correspond to the physical wave forms required on the interface signals. For more information on the Management Interface signal protocols, refer to the Media Independent Interface standard of IEEE 802.3u Specification. For reference, the timing parameters of the MDC and MDIO signals from the IEEE specification, Clause 22.2.11, and 22.3.4 are given in Table 19.

PARAMETER	VALUE	UNITS
MDC minimum low time	160	ns
MDC minimum high time	160	ns
MDC minimum period	400	ns
MDIO to MDC minimum setup time (when MDIO is driven by the Rio)	10	ns
MDIO to MDC minimum hold time (when MDIO is driven by the Rio)	10	ns
MDIO to MDC maximum delay time (when MDIO is driven by an internal PCS or external PHY)	300	ns

TABLE 19: MgmtClk/MgmtData Timing Requirements

BIT	BIT NAME	R/W	BIT DESCRIPTION
0	MgmtClk	R/W	GMII Management Clock. MgmtClk directly drives the management clock, either the internal MDC signal to the PCS layer, or the MDC/EWRAP signal to external PHY device(s).
1	MgmtData	R/W	GMII Management Data Bit. When the MgmtDir bit is a logic 1, the value written to MgmtData is driven onto the MDIO signal. When MgmtDir is a logic 0, data driven by the PHY device can be read from MgmtData.
2	MgmtDir	R/W	GMII Management Data Direction. Setting MgmtDir causes Rio to drive MDIO with the data bit written into MgmtData.
3	PhyDuplexPolarity	R/W	PHY Duplex Polarity. When PhyDuplexPolarity is a logic 0 the PHYDPLXN input pin is active low.

<table>
<tr><th>BIT</th><th>BIT NAME</th><th>R/W</th><th>BIT DESCRIPTION</th></tr>
<tr><td>4</td><td>PhyDuplexStatus</td><td>R</td><td>PHY Duplex Status. PhyDuplexStatus provides a real-time indication of the duplex status of the PHY. If PhyDuplexStatus is a logic 1, the PHY is operating in full duplex mode.</td></tr>
<tr><td>5</td><td>PhyLnkPolarity</td><td>R/W</td><td>PHY Link Polarity. When PhyLnkPolarity is a logic 0 the PHYLNK10N, PHYLNK100N, and PHYLNK100N input pins are active low.</td></tr>
<tr><td>7..6</td><td>LinkSpeed</td><td>R</td><td>PHY Link Speed/Status. LinkSpeed provides a real-time indication of the link status of the PHY.
<table>
<tr><th>BIT7</th><th>BIT6</th><th>LINK SPEED/STATUS</th></tr>
<tr><td>0</td><td>0</td><td>Link down (default).</td></tr>
<tr><td>0</td><td>1</td><td>Link up at 10Mbps.</td></tr>
<tr><td>1</td><td>0</td><td>Link up at 100Mbps.</td></tr>
<tr><td>1</td><td>1</td><td>Link up at 1000Mbps.</td></tr>
</table>
</td></tr>
</table>

11.10.5.20 ReceiveMode

Class..... I/O Registers, Control and Status
I/O Base Address ..... IoBaseAddress register value
Memory Base Address . MemBaseAddress register value
Address Offset..... 0x88
Default Value 0x00
Access Rule..... Word
Width ..... 16 bits

Each bit in ReceiveMode, when set, enables reception of a different type of frame.

BIT	BIT NAME	R/W	BIT DESCRIPTION
0	ReceiveUnicast	R/W	Receive Unicast Frames when a logic 1 enables the Rio to receive unicast frames that match the 48-bit value in the StationAddress register of the Rio.
1	ReceiveMulticast	R/W	Receive Multicast Frames when a logic 1 enables the Rio to receive all multicast frames, including broadcast.
2	ReceiveBroadcast	R/W	Receive Broadcast Frames when a logic 1 enables the Rio to receive all broadcast frames.
3	ReceiveAllFrames	R/W	Receive All Frames when a logic 1 enables the Rio to receive all frames promiscuously.
4	ReceiveMulticast-Hash	R/W	Receive Multicast Frames Using Hash Table, when a logic 1 enables the Rio to receive frames that pass the hash filtering mechanism defined in the HashTable register.
5	ReceiveIPMulticast	R/W	Receive IP Multicast Frames when a logic 1 enables the Rio to receive all multicast IP datagrams, which are mapped into Ethernet multicast frames with destination address of 01:00:5e:xx:xx:xx as defined in RFC 1112 and RFC 1700. The first 3 bytes require exact match, and the last 3 bytes are ignored.
7..6	Reserved	N/A	Reserved for future use. Write as zero, ignore on read.
8	ReceiveVLAN-Match	R/W	Receive Frames With Matching VLAN Tags when a logic 1 indicates the Rio will filter (or drop) receive frames which do not match one of the VLAN tags specified in the VLANId register. Before verifying VLAN tags, all receive frames must pass the filtering specified by the ReceiveUnicast, ReceiveMulticast, ReceiveBroadcast, ReceiveAllFrames, ReceiveMulticast-Hash, and ReceiveIPMulticast bit configurations. Frames without a VLAN tag are dropped.

BIT	BIT NAME	R/W	BIT DESCRIPTION
9	ReceiveVLANHash	R/W	Receive Frames Using VLAN Tag Hash Table, enables the Rio to filter (or drop) receive frames which do not pass the hash filtering mechanism defined in the VLANHashTable register. Before verifying VLAN tags, all receive frames must pass the filtering specified by the ReceiveUnicast, ReceiveMulticast, ReceiveBroadcast, ReceiveAllFrames, ReceiveMulticastHash, and ReceiveIPMulticast bit configurations. Frames without a VLAN tag are dropped.
15..10	Reserved	N/A	Reserved for future use. Write as zero, ignore on read.

11.10.5.21 RFDListPtr

Class..... I/O Registers, DMA
I/O Base Address ..... IoBaseAddress register value
Memory Base Address . MemBaseAddress register value
Address Offset..... 0x1C
Default Value 0x0000000000000000
Access Rule..... Double Word
Width ..... 64 bits

BIT	BIT NAME	R/W	BIT DESCRIPTION
39..0	RFDListPtr	R/W	RFD List Pointer. RFDListPtr holds the physical address within host system memory of the current RFD in the RFD list. A value of 0x0000000000000000 for RFDListPtr indicates that no more RFDs are available to accept received Ethernet frame data. RFDListPtr can only specify host system memory addresses which are on 8-byte boundaries (i.e. bits 2 through 0 must be 0), therefore RFDs must be aligned on 8-byte physical address boundaries. RFDListPtr must be written directly by the host system initially to indicate the head of a newly created RFD list and is subsequently updated by the Rio as it processes RFDs in the RFD list. As the Rio finishes processing a RFD, it loads RFDListPtr with the value from the current RFD's RFDNextPtr field. If the Rio reads a value of 0x0000000000000000 from the current RFD's RFDNextPtr field, the receive DMA process enters the polling state, waiting for a non-zero value to be written to current RFD's RFDNextPtr.
63..40	Reserved	N/A	Reserved for future use. Write as zero, ignore on read.

11.10.5.22 RMONStatisticsMask

Class..... I/O Registers, Statistics
I/O Base Address ..... IoBaseAddress register value
Memory Base Address . MemBaseAddress register value
Address Offset..... 0x98
Default Value 0x00000000
Access Rule..... Double Word
Width ..... 32 bits

BIT	BIT NAME	R/W	BIT DESCRIPTION
0	EtherStatsCR-CAlignErrors	R/W	EtherStatsCRCAlignErrors Mask when a logic 1 indicates the EtherStatsCRCAlignErrors RMON statistic register will not increment.
1	EtherStatsUnder-sizePkts	R/W	EtherStatsUndersizePkts Mask when a logic 1 indicates the EtherStatsUndersizePkts RMON statistic register will not increment.
2	EtherStatsFrag-ments	R/W	EtherStatsFragments Mask when a logic 1 indicates the EtherStatsFragments RMON statistic register will not increment.
3	EtherStatsJabbers	R/W	EtherStatsJabbers Mask when a logic 1 indicates the EtherStatsJabbers RMON statistic register will not increment.
4	EtherStatsOctets/ EtherStatsPkts	R/W	EtherStatsOctets/EtherStatsPkts Mask when a logic 1 indicates the EtherStatsOctets and EtherStatsPkts RMON statistic registers will not increment.
5	EtherStatsPkts64Octets	R/W	EtherStatsPkts64Octets Mask when a logic 1 indicates the EtherStatsPkts64Octets RMON statistic register will not increment.
6	EtherStatsPkts65to127Octets	R/W	EtherStatsPkts65to127Octets Mask when a logic 1 indicates the EtherStatsPkts65to127Octets RMON statistic register will not increment.
7	EtherStatsPkts128to255Octets	R/W	EtherStatsPkts128to255Octets Mask when a logic 1 indicates the EtherStatsPkts128to255Octets RMON statistic register will not increment.
8	EtherStatsPkts256to511Octets	R/W	EtherStatsPkts256to511Octets Mask when a logic 1 indicates the EtherStatsPkts256to511Octets RMON statistic register will not increment.
9	EtherStatsPkts512to1023Octets	R/W	EtherStatsPkts512to1023OctetsTransmit Mask when a logic 1 indicates the EtherStatsPkts512to1023Octets RMON statistic register will not increment.

BIT	BIT NAME	R/W	BIT DESCRIPTION
10	EtherStatsPkts1024to1518Octets	R/W	EtherStatsPkts1024to1518Octets Mask when a logic 1 indicates the EtherStatsPkts1024to1518Octets RMON statistic register will not increment.
11	EtherStatsCollisions	R/W	EtherStatsCollisions Mask when a logic 1 indicates the EtherStatsCollisions RMON statistic register will not increment.
12	EtherStatsOctetsTransmit/EtherStatsPktsTransmit	R/W	EtherStatsOctetsTransmit/EtherStatsPktsTransmit Mask when a logic 1 indicates the EtherStatsOctetsTransmit and EtherStatsPktsTransmit RMON statistic registers will not increment.
13	EtherStatsPkts64OctetsTransmit	R/W	EtherStatsPkts64OctetsTransmit Mask when a logic 1 indicates the EtherStatsPkts64OctetsTransmit RMON statistic register will not increment.
14	EtherStatsPkts65to127OctetsTransmit	R/W	EtherStatsPkts65to127OctetsTransmit Mask when a logic 1 indicates the EtherStatsPkts65to127OctetsTransmit RMON statistic register will not increment.
15	EtherStatsPkts128to255OctetsTransmit	R/W	EtherStatsPkts128to255OctetsTransmit Mask when a logic 1 indicates the EtherStatsPkts128to255OctetsTransmit RMON statistic register will not increment.
16	EtherStatsPkts256to511OctetsTransmit	R/W	EtherStatsPkts256to511OctetsTransmit Mask when a logic 1 indicates the EtherStatsPkts256to511OctetsTransmit RMON statistic register will not increment.
17	EtherStatsPkts512to1023OctetsTransmit	R/W	EtherStatsPkts512to1023OctetsTransmit Mask when a logic 1 indicates the EtherStatsPkts512to1023OctetsTransmit RMON statistic register will not increment.
18	EtherStatsPkts1024to1518OctetsTransmit	R/W	EtherStatsPkts1024to1518OctetsTransmit Mask when a logic 1 indicates the EtherStatsPkts1024to1518OctetsTransmit RMON statistic register will not increment.
31..19	Reserved	N/A	Reserved for future use. Write as zero, ignore on read.

11.10.5.23 RxDMABurstThresh

Class..... I/O Registers, DMA
I/O Base Address ..... IoBaseAddress register value
Memory Base Address . MemBaseAddress register value
Address Offset..... 0x24
Default Value 0x08
Access Rule..... Byte
Width ..... 8 bits

BIT	BIT NAME	R/W	BIT DESCRIPTION
7..0	RxDMABurst-Thresh	R/W	Receive DMA Burst Threshold. RxDMABurstThresh sets the threshold by which the Rubicon determines when to assert receive DMA bus master requests. The threshold is specified based on the amount of occupied space in the receive FIFO in increments of 32 bytes. When the number of bytes in the receive FIFO occupied by receive data rises above the value specified by RxDMABurstThresh, the Rubicon may make a receive DMA request on the PCI bus. If the number of bytes in the receive FIFO occupied by receive data exceeds the value in the FragLen sub field of the FragInfo field of the current RFD, the Rio will make receive DMA bus request regardless of the value in RxDMABurstThresh. The maximum value of RxDMABurstThresh is 0xFF, or 8192 bytes (256 * 32 = 8192). The minimum value of RxDMABurst-Thresh is 0x08, or 256 bytes, and any smaller value is interpreted as 0x08.

11.10.5.24 RxDMAIntCtrl

Class..... I/O Registers, DMA
I/O Base Address ..... IoBaseAddress register value
Memory Base Address . MemBaseAddress register value
Address Offset..... 0x28
Default Value 0x00000000
Access Rule..... Double Word
Width ..... 32 bits

The RxDMAIntCtrl register is used for configuring the receive DMA interrupt coalescing feature of the Rio (see section 11.3.4). Receive DMA interrupts may be coalesced (that is grouped such that a single interrupt is issued to indicate multiple Ethernet frames have been received) and the frequency of receive DMA interrupts can be set based on either the number of Ethernet frames received (via the RxFrameCount field), or after a fixed amount of time following receipt of an Ethernet frame (via the RxDMAWaitTime field). It is not possible to completely disable the assertion of receive DMA interrupts if a receive DMA operation has taken place. If a receive DMA operation has completed, a RxDMAComplete interrupt will be asserted for based on either condition described by the RxDMAIntCtrl register, whichever occurs first.

BIT	BIT NAME	R/W	BIT DESCRIPTION
7..0	RxFrameCount	R/W	Receive Frame Count. RxFrameCount Indicates the maximum number of frames which will be transferred via a receive DMA transfer before the RxDMAComplete bit in the IntStatus register is set to a logic 1 (a RxDMAComplete interrupt). The minimum value of RxFrameCount is 0x01 (which indicates a RxDMAComplete interrupt will occur for every receive DMA transfer), and any smaller value is interpreted as 0x01.
9..8	Reserved	N/A	Reserved for future use. Write as zero, ignore on read.
12..10	PriorityThresh	R/W	Priority Threshold. PriorityThresh holds the value which is compared to the Priority field within received Ethernet frames which contain Tag Control Information (TCI) fields in the frame header. If a received Ethernet frame contains a TCI field with a priority value equal to or greater than the value in PriorityThresh, the RxDMAPriority bit in the IntStatus register is set to a logic 1 (a RxDMAPriority interrupt).
15..13	Reserved	N/A	Reserved for future use. Write as zero, ignore on read.

PRIORITYTHRESH FIELD BIT	ETHERNET FRAME TCI PRIORITY FIELD BIT
12	8
11	7
10	6

BIT	BIT NAME	R/W	BIT DESCRIPTION
31..16	RxDMAWaitTime	R/W	Receive DMA Wait Time. RxDMAWaitTime indicates the maximum amount of time (in 64ns increments) between completion of a receive DMA transfer operation and setting of the RxDMAComplete bit in the IntStatus register to a logic 1 (a RxDMAComplete interrupt). A value of 0x0000 indicates a RxDMAComplete interrupt will occur immediately following each receive DMA transfer.

11.10.5.25 RxDMAPollPeriod

Class..... I/O Registers, DMA
I/O Base Address ..... IoBaseAddress register value
Memory Base Address . MemBaseAddress register value
Address Offset..... 0x26
Default Value 0xFF
Access Rule..... Byte
Width ..... 8 bits.

BIT	BIT NAME	R/W	BIT DESCRIPTION
7..0	RxDMAPollPeriod	R/W	Receive DMA Poll Periond. RxDMAPollPeriod determines the rate at which the current RFD in the receive DMA process is read for changes in the RFDNextPtr and RFS fields. RxDMAPollPeriod is specified in 320 ns increments. The maximum value is 256 (or 81.92 us). The minimum valid value is 0x01 (or 320 ns), and any smaller value is interpreted as 0x01. Since RxDMAPollPeriod cannot be set to 0x00, polling is always in effect during receive DMA operation.

11.10.5.26 Reserved(RxDMAStatus)

Class..... I/O Registers, DMA

I/O Base Address ..... IoBaseAddress register value

Memory Base Address . MemBaseAddress register value

Address Offset..... 0x08

Access Rule..... Double Word

Default Value 0x00000000

Width ..... 32 bits

Receive DMA Status shows the status of various operations in the receive DMA process. Reserved(RxDMAStatus) should be read only while the receive DMA process is in the polling state (see section 11.2.2). The format of this register is identical to that of the RFS field of a RFD. The contents of Reserved(RxDMAStatus) are written into the RFS field of the current RFD upon completion of the receive DMA process.

For bit definitions, see 11.10.2.2 bits 0 through 31.

11.10.5.27 RxDMAUrgentThresh

Class..... I/O Registers, DMA
I/O Base Address ..... IoBaseAddress register value
Memory Base Address . MemBaseAddress register value
Address Offset..... 0x25
Default Value 0x04
Access Rule..... Byte
Width ..... 8 bits

BIT	BIT NAME	R/W	BIT DESCRIPTION
7..0	RxDMAUrgent-Thresh	R/W	Receive DMA Urgent Threshold. RxDMAUrgentThresh sets the threshold by which the Rio determines when to assert receive urgent DMA bus master requests. The threshold is specified based on the amount of unoccupied space in the receive FIFO in increments of 32 bytes. When the number of bytes in the receive FIFO not occupied by receive data falls below the value specified by RxDMAUrgentThresh, the Rio may make an urgent receive DMA request on the PCI bus. The maximum value of RxDMAUrgentThresh is 0xFF, or 8192 bytes (256 * 32 = 8192). The minimum value of RxDMAUrgent-Thresh is 0x04, or 128 bytes, and any smaller value is interpreted as 0x04.

11.10.5.28 RxEarlyThresh

Class..... I/O Registers, Control and Status
I/O Base Address ..... IoBaseAddress register value
Memory Base Address . MemBaseAddress register value
Address Offset..... 0x3A
Default Value 0x07FF
Access Rule..... Word
Width ..... 16 bits.

BIT	BIT NAME	R/W	BIT DESCRIPTION
10..0	RxEarlyThresh	R/W	Receive Early Threshold. RxEarlyThresh defines the number of quad-words (64 bits) of an Ethernet frame (beginning with the Destination Address field) which must be received before a RxEarly interrupt is issued, and before the start of the receive DMA process. If the RxEarlyDisable bit in DMACtrl register is a logic 1, the receive DMA process for a frame will not begin until the number of quad-words indicated by RxEarlyThresh have been received The minimum value of RxEarlyThresh is 0x01, any value smaller than this will be interpreted as 0x01.
15..11	Reserved	N/A	Reserved for future use. Write as zero, ignore on read.

11.10.5.29 RxJumboFrames

Class..... I/O Registers, Statistics
I/O Base Address ..... IoBaseAddress register value
Memory Base Address . MemBaseAddress register value
Address Offset..... 0xBC
Default Value 0x0000
Access Rule..... Word
Width ..... 16 bits

BIT	BIT NAME	R/W	BIT DESCRIPTION
15..0	RxJumboFrames	R/W	Jumbo Frames Received is a count of the number of frames received successfully whose frame length (measured from the Destination Address field to the end of the Frame Check Sequence field) is greater than 1518 bytes (1522 bytes for VLAN tagged frames). RxJumboFrames will wrap around to zero after reaching 0xFFFF. An UpdateStats interrupt (UpdateStats bit within the IntStatus register) will occur when RxJumboFrames reaches a value of 0xC000. RxJumboFrames is enabled by writing a logic 1 to the StatisticsEnable bit in the MACCtrl register, and a logic 0 to the RxJumboFrames bit within the StatisticsMask register.

11.10.5.30 StationAddress

Class..... I/O Registers, Control and Status
I/O Base Address ..... IoBaseAddress register value
Memory Base Address . MemBaseAddress register value
Address Offset..... 0x78
Default Value 0x000000000000
Access Rule..... Word, Double Word
Width ..... 48 bits

BIT	BIT NAME	R/W	BIT DESCRIPTION
47..0	StationAddress	R/W	StationAddress is used to define the value which the Rio will compare against the Destination Address field within received Ethernet frames. Ethernet addresses are generally specified as a collection of six 8-bit hexadecimal values separated by colons as in the following example: 01:23:45:67:89:AB This example address would be written to StationAddress as the following 48 bit hexadecimal value: 0xAB8967452301

The address comparison logic within the Rio will compare the bits within the Destination Address field (in order, with the first received bit as bit 0, and the last received bit as bit 47) of a received Ethernet frame with the corresponding bit within StationAddress. A match is detected if each bit in the Destination Address field of the received Ethernet frame matches the corresponding bit in StationAddress.

The value in StationAddress is not inserted into the Source Address field of Ethernet frames transmitted by the Rio. The Source Address field for each Ethernet frame must be specified by the host system as part of the frame contents.

11.10.5.31 StatisticsMask

Class..... I/O Registers, Statistics
I/O Base Address ..... IoBaseAddress register value
Memory Base Address . MemBaseAddress register value
Address Offset..... 0x9C
Default Value 0x00000000
Access Rule..... Double Word
Width ..... 32 bits

BIT	BIT NAME	R/W	BIT DESCRIPTION
0	OctetRcvOk/ FramesRcvdOk	R/W	OctetRcvOk/FramesRcvdOk Mask when a logic 1 indicates the OctetRcvOk and FramesRcvdOk statistic registers will not increment.
1	McstOctetRcvdOk/ McstFramesRcv-dOk	R/W	McstOctetRcvdOk/McstFramesRcvdOk Mask when a logic 1 indicates the McstOctetRcvdOk and McstFramesRcvdOk statistic registers will not increment.
2	BcstOctetRcvOk/ BcstFramesRcv-dOk	R/W	BcstOctetRcvOk/BcstFramesRcvdOk Mask when a logic 1 indicates the BcstOctetRcvOk and BcstFramesRcvdOk statistic registers will not increment.
3	RxJumboFrames	R/W	RxJumboFrames Mask when a logic 1 indicates the RxJumbo-Frames statistic register will not increment.
4	TCPCheckSumEr-rors	R/W	TCPCheckSumErrors Mask when a logic 1 indicates the TCPCheckSumErrors statistic register will not increment.
5	IPCheckSumErrors	R/W	IPCheckSumErrors Mask when a logic 1 indicates the IPCheckSumErrors statistic register will not increment.
6	UDPCheckSumEr-rors	R/W	UDPCheckSumErrors Mask when a logic 1 indicates the UDPCheckSumErrors statistic register will not increment.
7	MacControl-FramesRcvd	R/W	MacControlFramesRcvd Mask when a logic 1 indicates the MacControlFramesRcvd statistic register will not increment.
8	FrameTooLongEr-rors	R/W	FrameTooLongErrors Mask when a logic 1 indicates the FrameTooLongErrors statistic register will not increment.
9	InRangeLengthEr-rors	R/W	InRangeLengthErrors Mask when a logic 1 indicates the InRangeLengthErrors statistic register will not increment.
10	FramesCheckSe-qErrors	R/W	FramesCheckSeqErrors Mask when a logic 1 indicates the FramesCheckSeqErrors statistic register will not increment.
11	FramesLostRxEr-rors	R/W	FramesLostRxErrors Mask when a logic 1 indicates the FramesLostRxErrors statistic register will not increment.

BIT	BIT NAME	R/W	BIT DESCRIPTION
12	OctetXmtOk/ FramesXmtdOk	R/W	OctetXmtOk/FramesXmtdOk Mask when a logic 1 indicates the OctetXmtOk and FramesXmtdOk statistic registers will not increment.
13	McstOctetXmtOk/ McstFramesXmt-dOk	R/W	McstOctetXmtOk/McstFramesXmtdOk Mask when a logic 1 indicates the McstOctetXmtOk and McstFramesXmtdOk statistic registers will not increment.
14	BcstOctetXmtOk/ BcstFramesXmt-dOk	R/W	BcstOctetXmtOk/BcstFramesXmtdOk Mask when a logic 1 indicates the BcstOctetXmtOk and BcstFramesXmtdOk statistic registers will not increment.
15	FramesWDe-ferredXmt	R/W	FramesWDeferredXmt Mask when a logic 1 indicates the FramesWDeferredXmt statistic register will not increment.
16	LateCollisions	R/W	LateCollisions Mask when a logic 1 indicates the LateCollisions statistic register will not increment.
17	MultiColFrames	R/W	MultiColFrames Mask when a logic 1 indicates the MultiColFrames statistic register will not increment.
18	SingleColFrames	R/W	SingleColFrames Mask when a logic 1 indicates the SingleColFrames statistic register will not increment.
19	TxJumboFrames	R/W	TxJumboFrames Mask when a logic 1 indicates the TxJumboFrames statistic register will not increment.
20	CarrierSenseErrors	R/W	CarrierSenseErrors Mask when a logic 1 indicates the CarrierSenseErrors statistic register will not increment.
21	MacControl-FramesXmtd	R/W	MacControlFramesXmtd Mask when a logic 1 indicates the MacControlFramesXmtd statistic register will not increment.
22	FramesAbortX-SColls	R/W	FramesAbortXSColls Mask when a logic 1 indicates the FramesAbortXSColls statistic register will not increment.
23	FramesWEXDe-feral	R/W	FramesWEXDeferal Mask when a logic 1 indicates the FramesWEXDeferal statistic register will not increment.
31..24	Reserved	N/A	Reserved for future use. Write as zero, ignore on read.

11.10.5.32 TCPCheckSumErrors

Class..... I/O Registers, Statistics
I/O Base Address ..... IoBaseAddress register value
Memory Base Address . MemBaseAddress register value
Address Offset..... 0xC0
Default Value 0x0000
Access Rule..... Word
Width ..... 16 bits

BIT	BIT NAME	R/W	BIT DESCRIPTION
15..0	TCPChecksumErrors	R/W	TCP Check Sum Errors is a count of received frames which contain TCP segments within IP datagrams, which fail the TCP checksum as defined in RFC 793. TCPCheckSumErrors will wrap around to zero after reaching 0xFFFF. An UpdateStats interrupt (UpdateStats bit within the IntStatus register) will occur when TCPCheckSumErrors reaches a value of 0xC000. TCPCheckSumErrors is enabled by writing a logic 1 to the StatisticsEnable bit in the MACCtrl register, and a logic 0 to the TCPCheckSumErrors bit within the StatisticsMask register.

11.10.5.33 TFDListPtr

Class..... I/O Registers, DMA
I/O Base Address ..... IoBaseAddress register value
Memory Base Address . MemBaseAddress register value
Address Offset..... 0x10
Default Value 0x0000000000000000
Access Rule..... Double Word
Width ..... 64 bits

BIT	BIT NAME	R/W	BIT DESCRIPTION
39..0	TFDListPtr	R/W	TFD List Pointer. TFDListPtr holds the physical address within host system memory of the current TFD in the TFD list. A value of 0x0000000000000000 for TFDListPtr indicates that no more TFDs are containing Ethernet frame data to transmit are available. TFDListPtr can only specify host system memory addresses which are on 8-byte boundaries (i.e. bits 2 through 0 must be 0), therefore TFDs must be aligned on 8-byte physical address boundaries. TFDListPtr must be written directly by the host system initially to indicate the head of a newly created TFD list and is subsequently updated by the Rio as it processes TFDs in the TFD list. As the Rio finishes processing a TFD, it loads TFDListPtr with the value from the current TFD's TFDNextPtr field. If the Rio reads a value of 0x0000000000000000 from the current TFD's TFDNextPtr field, the transmit DMA process enters the polling state, waiting for a non-zero value to be written to current TFD's TFDNextPtr field.
63..40	Reserved	N/A	Reserved for future use. Write as zero, ignore on read.

11.10.5.34 TxDMABurstThresh

Class..... I/O Registers, DMA
I/O Base Address ..... IoBaseAddress register value
Memory Base Address . MemBaseAddress register value
Address Offset..... 0x18
Default Value 0x08
Access Rule..... Byte
Width ..... 8 bits

BIT	BIT NAME	R/W	BIT DESCRIPTION
7..0	TxDMABurst-Thresh	R/W	Transmit DMA Burst Threshold. TxDMABurstThresh sets the threshold by which the Rio determines when to assert transmit DMA bus master requests. The threshold is specified based on the amount of unoccupied space in the transmit FIFO in increments of 32 bytes. When the number of bytes in the transmit FIFO unoccupied by transmit data rises above the value specified by TxDMABurstThresh, the Rubicon may make a transmit DMA request on the PCI bus. If the number of bytes in the transmit FIFO occupied by transmit data exceeds the value in the FragLen sub field of the FragInfo field of the current TFD, Rubicon will make transmit DMA bus request regardless of the value in TxDMABurstThresh. The maximum value of TxDMABurstThresh is 0xFF, or 8192 bytes (256 * 32 = 8192). The minimum value of TxDMABurst-Thresh is 0x08, or 256 bytes, and any smaller value is interpreted as 0x08.

11.10.5.35 TxDMAPollPeriod

Class..... I/O Registers, DMA
I/O Base Address ..... IoBaseAddress register value
Memory Base Address . MemBaseAddress register value
Address Offset..... 0x1A
Default Value 0xFF
Access Rule..... Byte
Width ..... 8 bits

BIT	BIT NAME	R/W	BIT DESCRIPTION
7..0	TxDMAPollPeriod	R/W	Transmit DMA Poll Periond. TxDMAPollPeriod determines the rate at which the current TFD in the transmit DMA process is read for changes in the TFDNextPtr and TFC fields. TxDMAPollPeriod is specified in 320 ns increments. The maximum value is 256 (or 81.92 us). The minimum valid value is 0x01 (or 320 ns), and any smaller value is interpreted as 0x01. Since TxDMAPollPeriod cannot be set to 0x00, polling is always in effect during transmit DMA operation.

11.10.5.36 TxDMAUrgentThresh

Class..... I/O Registers, DMA
I/O Base Address ..... IoBaseAddress register value
Memory Base Address . MemBaseAddress register value
Address Offset..... 0x19
Default Value 0x04
Access Rule..... Byte
Width ..... 8 bits.

BIT	BIT NAME	R/W	BIT DESCRIPTION
7..0	TxDMAUrgent-Thresh	R/W	Transmit DMA Urgent Threshold. TxDMAUrgentThresh sets the threshold by which the Rio determines when to assert transmit urgent DMA bus master requests. The threshold is specified based on the amount of occupied space in the transmit FIFO in increments of 32 bytes. When the number of bytes in the transmit FIFO occupied by transmit data falls below the value specified by TxDMAUrgentThresh, the Rio may make an urgent transmit DMA request on the PCI bus. An urgent transmit DMA request will have priority over the receive DMA process, unless the receive DMA process is also making an urgent request. The maximum value of TxDMAUrgentThresh is 0xFF, or 8192 bytes (256 * 32 = 8192). The minimum value of TxDMAUrgent-Thresh is 0x04, or 128 bytes, and any smaller value is interpreted as 0x04.

11.10.5.37 TxJumboFrames

Class..... I/O Registers, Statistics
I/O Base Address ..... IoBaseAddress register value
Memory Base Address . MemBaseAddress register value
Address Offset..... 0xF4
Default Value 0x0000
Access Rule..... Word
Width ..... 16 bits

BIT	BIT NAME	R/W	BIT DESCRIPTION
15..0	TxJumboFrames	R/W	Jumbo Frames Transmitted is a count of the number of frames transmitted successfully whose frame length (measured from the Destination Address field to the end of the Frame Check Sequence field) is greater than 1518 bytes (1522 bytes for VLAN tagged frames). TxJumboFrames will wrap around to zero after reaching 0xFFFF. An UpdateStats interrupt (UpdateStats bit within the IntStatus register) will occur when TxJumboFrames reaches a value of 0xC000. TxJumboFrames is enabled by writing a logic 1 to the StatisticsEnable bit in the MACCtrl register, and a logic 0 to the TxJumboFrames bit within the StatisticsMask register.

11.10.5.38 TxStartThresh

Class..... I/O Registers, Control and Status
I/O Base Address ..... IoBaseAddress register value
Memory Base Address . MemBaseAddress register value
Address Offset..... 0x44
Default Value 0x0FFF
Access Rule..... Word
Width ..... 16 bits

BIT	BIT NAME	R/W	BIT DESCRIPTION
11..0	TxStartThresh	R/W	Transmit Start Threshold. TxStartThresh is used to control when frames are transmitted. Transmission of a frame begins when the number of double words (32 bit values) for the frame which have been transferred into the transmit FIFO is greater than the value specified by TxStartThresh. If the TxStartThresh is set too low, the transmit FIFO may experience underruns due to transfer rate differences between the transmit DMA process, and the Ethernet network. Use of TxStartThresh is not recommended for Gigabit Ethernet networks, except during transfer of jumbo frames. If the IPChecksumEnable, TCPChecksumEnable, or UDPChecksumEnable bits of the frame's TFC are a logic 1, the TxStartThresh value is ignored (since calculation of checksums requires the entire frame be transferred via transmit DMA prior to the start of frame transmission).
15..12	Reserved	N/A	Reserved for future use. Write as zero, ignore on read.

11.10.5.39 TxStatus

Class..... I/O Registers, Control and Status
I/O Base Address ..... IoBaseAddress register value
Memory Base Address . MemBaseAddress register value
Address Offset..... 0x60
Default Value 0x00000000
Access Rule..... Byte
Width ..... 32 bits

TxStatus returns the status of frame transmission or transmission attempts. TxStatus is cleared when read.

BIT	BIT NAME	R/W	BIT DESCRIPTION
0	TxError	R	Transmit Error. TxError indicates an error occurred during transmission of the frame indicated by the TxFrameId field of TxStatus. TxError is a logic 1 when either LateCollision, MaxCollisions, or TxUnderrun is a logic 1.
1	Reserved	N/A	Reserved for future use. Write as zero, ignore on read.
2	LateCollision	R	Late Collision. When LateCollision is a logic 1 the frame experienced a collision after the slot time (4,096 bit times). The TxEnable bit in the MACCtrl register must be set to a logic 1 to recover from this condition. The frame is kept in the transmit FIFO, so when the MAC is re-enabled the frame will be transmitted again. To prevent re-transmission of this frame, the FIFO must be cleared via the FIFO bit in the AsicCtrl register prior to re-enabling the MAC for transmission.
3	MaxCollisions	R	Maximum Collisions. When MaxCollisions is a logic 1 the frame was not successfully transmitted due to encountering 16 collisions. The TxEnable bit in the MACCtrl register must be set to a logic 1 to recover from this condition. The frame is discarded from the transmit FIFO, so the host system should resubmit the frame for transmission.
4	TxUnderrun	R	Transmit Underrun. When TxUnderrun is a logic 1 the frame experienced an underrun during the transmit process because the host was unable to supply the frame data fast enough to keep up with the network data rate. An underrun will halt the transmitter and the transmit FIFO. The TxReset bit within the AsicCtrl register and TxEnable bit within the MACCtrl register must be set to a logic 1 prior to re-starting any frame.
5	Reserved	N/A	Reserved for future use. Write as zero, ignore on read.
6	TxIndicateReqd	R	Transmit Indicate Requested. When TxIndicateReqd is a logic 1 the TxIndicate bit in the TFC field was a logic 1 when the transmitted frame's TFD was transferred via the transmit DMA process.

BIT	BIT NAME	R/W	BIT DESCRIPTION
7	TxComplete	R	Transmit Complete. When TxComplete is a logic 1 the bits in the TxStatus register are valid and can be read by the host system. All fields within the TxStatus register are updated after transmission (either successfully or with errors) of an Ethernet frame.
15..8	Reserved	N/A	Reserved for future use. Write as zero, ignore on read.
31..16	TxFrameId	R	Transmit Frame Identifier. TxFrameId holds the value of the Ethernet frame for which the TxStatus register fields apply.

11.10.5.40 UDPCheckSumErrors

Class..... I/O Registers, Statistics
I/O Base Address ..... IoBaseAddress register value
Memory Base Address . MemBaseAddress register value
Address Offset..... 0xC4
Default Value 0x0000
Access Rule..... Word
Width ..... 16 bits

BIT	BIT NAME	R/W	BIT DESCRIPTION
15..0	UDPCheckSumErrors	R/W	UDP Check Sum Errors is a count of received frames which contain UDP segments within IP datagrams, which fail the UDP checksum as defined in RFC 768. UDPCheckSumErrors will wrap around to zero after reaching 0xFFFF. An UpdateStats interrupt (UpdateStats bit within the IntStatus register) will occur when UDPCheckSumErrors reaches a value of 0xC000. UDPCheckSumErrors is enabled by writing a logic 1 to the StatisticsEnable bit in the MACCtrl register, and a logic 0 to the UDPCheckSumErrors bit within the StatisticsMask register.

11.10.5.41 VLANHashTable

Class..... I/O Registers, Control and Status
I/O Base Address ..... IoBaseAddress register value
Memory Base Address . MemBaseAddress register value
Address Offset..... 0x8A
Default Value 0x0000
Access Rule..... Word
Width ..... 16 bits

BIT	BIT NAME	R/W	BIT DESCRIPTION
15..0	VLANHashTable	R/W	VLAN Hash Table holds a 16-bit value used for selectively receiving VLAN tagged frames. Setting the ReceiveVLANHash bit in the ReceiveMode register enables the filtering mechanism. The hash table is cleared upon reset, and must be properly set by the host. The Rio applies a 4-bit one's complement checksum algorithm to the 12-bit VID field of a VLAN tagged receive frame. The 4-bit checksum result is used as an addressing index into the VLAN hash table. If the VLANHashTable bit addressed by the index is a logic 1, the frame is accepted by the Rio and transferred to higher layers. If the VLANHashTable bit addressed by the index is a logic 0, the frame is discarded. The checksum algorithm used by the Rio is described below: $VID = a_{11}a_{10}a_9a_8a_7a_6a_5a_4a_3a_2a_1a_0$ $c_0b_3b_2b_1b_0 = a_{11}a_{10}a_9a_8 + a_7a_6a_5a_4$ $s_3s_2s_1s_0 = a_3a_2a_1a_0 + b_3b_2b_1b_0 + c_0$ $Checksum = s_3's_2's_1's_0'$

11.10.5.42 VLANId

Class..... I/O Registers, Control and Status
I/O Base Address ..... IoBaseAddress register value
Memory Base Address . MemBaseAddress register value
Address Offset..... 0x80
Default Value 0x000000000000
Access Rule..... Word
Width ..... 48 bits

BIT	BIT NAME	R/W	BIT DESCRIPTION
11..0	VLANId0	R/W	VLAN Identification 0 specifies one, 12 bit VLAN Identifier (corresponding to the VID field of the VLAN tag portion of a receive frame) used for receive frame filtering as defined by the ReceiveMode register ReceiveVLANMatch bit. Bit 11 of VLANId0 corresponds to the most significat bit of the VID.
23..12	VLANId1	R/W	VLAN Identification 1 specifies one, 12 bit VLAN Identifier (corresponding to the VID field of the VLAN tag portion of a receive frame) used for receive frame filtering as defined by the ReceiveMode register ReceiveVLANMatch bit. Bit 23 of VLANId1 corresponds to the most significat bit of the VID.
35..24	VLANId2	R/W	VLAN Identification 2 specifies one, 12 bit VLAN Identifier (corresponding to the VID field of the VLAN tag portion of a receive frame) used for receive frame filtering as defined by the ReceiveMode register ReceiveVLANMatch bit. Bit 35 of VLANId2 corresponds to the most significat bit of the VID.
47..36	VLANId3	R/W	VLAN Identification 3 specifies one, 12 bit VLAN Identifier (corresponding to the VID field of the VLAN tag portion of a receive frame) used for receive frame filtering as defined by the ReceiveMode register ReceiveVLANMatch bit. Bit 47 of VLANId3 corresponds to the most significat bit of the VID.

11.10.5.43 VLANTag

Class..... I/O Registers, Control and Status
I/O Base Address ..... IoBaseAddress register value
Memory Base Address . MemBaseAddress register value
Address Offset..... 0x70
Default Value 0x81000000
Access Rule..... Double Word
Width ..... 32 bits

The VLANTag register contains the information used by the Rio when inserting a VLAN tag within a frame based on the state of the AutoVLANtagging bit of the MACCtrl register. The format of a VLAN tag as defined in IEEE 802.3Q is shown in Figure 5.

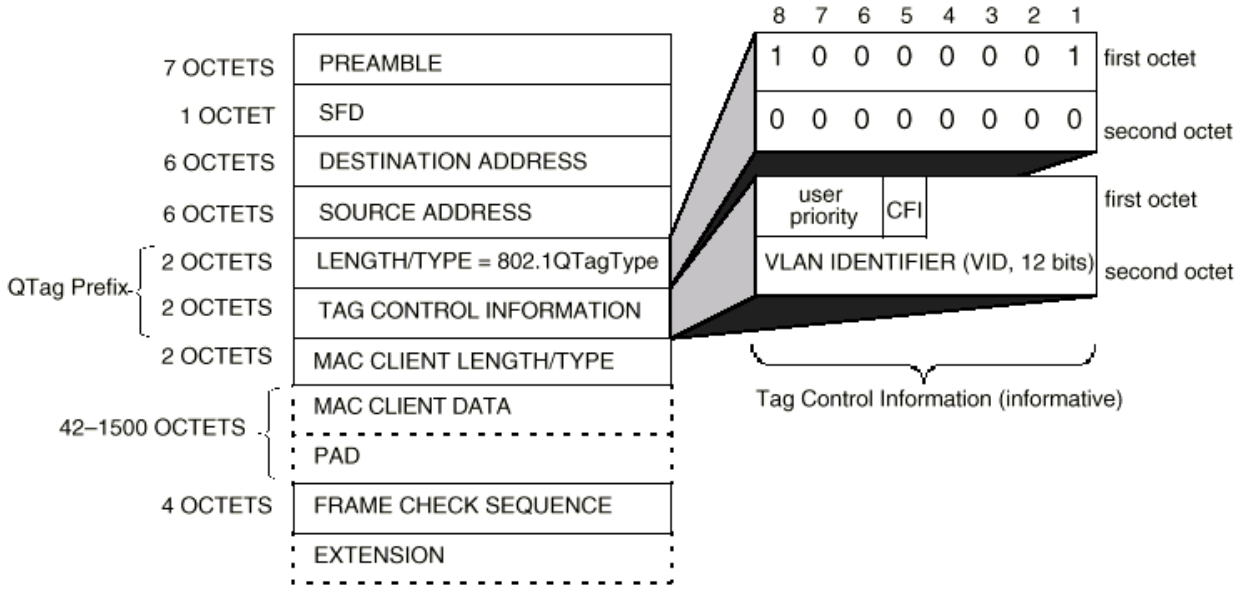


FIGURE 5: VLAN Tag Format from IEEE 802.3 Standard

BIT	BIT NAME	R/W	BIT DESCRIPTION
11..0	VID	R/W	VLAN Identifier. VID indicates the VLAN to which the Ethernet frame belongs. The VID is encoded as an unsigned binary number. A value of 0x000 indicates the Tag Header contains only “user_priority” information; no VLAN identifier is present in the frame. The values 0x001 and 0xFFF are reserved, all other values may be used as valid VLAN identifiers.

BIT	BIT NAME	R/W	BIT DESCRIPTION
12	CFI	R/W	Canonical Format Indicator. When CFI is a logic 0, all MAC address information that may be present in the MAC data carried by the frame is in Canonical format. When CFI is a logic 1, the E-RIF field is present in the Tag Header, and that the NCFI bit in the RIF field determines whether MAC address information that may be present in the MAC data carried by the frame is in Canonical or Non-canonical format. See IEEE 802.3Q for more information.
15..13	UserPriority	R/W	User Priority. UserPriority indicates the priority of the Ethernet frame. Values of 0 through 7 are valid. See IEEE 802.3Q for more information.
23..16	TPID2	R/W	Tag Protocol Identifier Octet 2. TPID2 is the second octet transmitted (or received) of the QTag Prefix field within an Ethernet frame. TPID2 value must always be 0x00.
31..24	TPID1	R/W	Tag Protocol Identifier Octet 1. TPID1 is the first octet transmitted (or received) of the QTag Prefix field within an Ethernet frame. TPID1 value must always be 0x81.

11.10.5.44 WakeEvent

Class..... I/O Registers, Control and Status
I/O Base Address ..... IoBaseAddress register value
Memory Base Address . MemBaseAddress register value
Address Offset..... 0x51
Default Value 0x08
Access Rule..... Byte
Width ..... 8 bits

WakeEvent contains enable bits to control which types of events can generate a wake event to the host system. WakeEvent also contains status bits indicating the specific wake events which have occurred.

BIT	BIT NAME	R/W	BIT DESCRIPTION
0	WakePktEnable	R/W	Wake Packet Enable. When WakePktEnable is a logic 1 the Rio generates wake events via a PCI interrupt due to reception of a Wake Packet (see section 11.4 for more details). The PmeEn bit in the PowerMgmtCtrl register must be a logic 1 in order for WakePktEnable to be recognized. WakePktEnable has no effect in power mode D0.
1	MagicPktEnable	R/W	Magic Packet Enable. When MagicPktEnable is a logic 1 the Rio generates wake events via a PCI interrupt due to reception of a Magic Packet (see section 11.4 for more details). The PmeEn bit in the PowerMgmtCtrl register must be a logic 1 in order for MagicPktEnable to be recognized. MagicPktEnable has no effect in power mode D0.
2	LinkEventEnable	R/W	Link Event Enable. When LinkEventEnable is a logic 1 the Rio generates wake events via a PCI interrupt due to a change in the PHYLNK10N signal (i.e. a change in the link status). The PmeEn bit in the PowerMgmtCtrl register must be a logic 1 in order for LinkEventEnable to be recognized. LinkEventEnable has no effect in power mode D0.
3	WakePolarity	R/W	Wake Polarity. When WakePolarity is a logic 1, the WAKE signal is asserted in the HIGH state. When WakePolarity is a logic 0, the WAKE signal is asserted in the LOW state.
4	WakePktEvent	R	Wake Packet Event. When WakePktEvent is a logic 1, a Wake Packet which meets the reception criteria set by the host system (see section 11.4 for more details) has been received. WakePktEnable must be a logic 1 in order for WakePktEvent to indicate Wake Packet reception. WakePktEvent is cleared when the WakeEvent register is read.

BIT	BIT NAME	R/W	BIT DESCRIPTION
5	MagicPktEvent	R	Magic Packet Event. When MagicPktEvent is a logic 1, a Magic Packet which meets the reception criteria set by the host system (see section 11.4 for more details) has been received. MagicPktEnable must be a logic 1 in order for MagicPktEvent to indicate Wake Packet reception. MagicPktEvent is cleared when the WakeEvent register is read.
6	LinkEvent	R	Link Event. When LinkEvent is a logic 1, a change in the PHYLNK10N signal (i.e. a change in the link status) has occurred. LinkEventEnable must be a logic 1 in order for LinkEvent to indicate Wake Packet reception. LinkEvent is cleared when the WakeEvent register is read.
7	WakeOnLanEnable	R/W	Wake On LAN Enable. When WakeOnLanEnable is a logic 1 the Rio is placed in the WakeOnLan Mode (see section 11.5 for more details) regardless of the PowerMgmtCtrl register settings.

11.10.6 PCI Configuration Registers

PCI based systems use a slot-specific block of configuration registers to perform configuration of devices on the PCI bus. The configuration registers are accessed with PCI Configuration Cycles. The PCI bus supports two types of Configuration Cycles. Type 0 cycles are used to configure devices on the local PCI bus. Type 1 cycles are used to pass a configuration request to a PCI bus at a different hierarchical level.

PCI Configuration Cycles are directed at one out of eight possible PCI logical functions within a single physical PCI device. A Rio based PCI bus master device responds only to Type 0 Configuration Cycles, directed at function 0. Type 1 cycles, and Type 0 cycles directed at functions other than 0, are ignored by the Rio.

Each PCI bus device is required to decode 256 bytes of configuration registers. Of these, the first 64 bytes are pre-defined by the PCI Specification. The remaining registers may be used as needed for PCI device-specific configuration registers. In PCI Configuration Cycles, the host system provides a slot-specific decode signal (IDSEL) which informs the PCI device that a configuration cycle is in progress. The PCI device responds by asserting DEVSELN, and decoding the specific configuration register from the address bus and the byte enable signals. See the PCI Expansion ROM specification for information on generating configuration cycles from driver software.

Table 20 shows the PCI configuration registers implemented by Rio. All locations marked “Reserved”, and all of the locations within the 256-byte configuration space (0xFF through 0x00) that are not shown in the table, are not implemented and return zero when read..

TABLE 20: Rubicon PCI Register Layout

CLK	BYTE 4	BYTE 3	BYTE 2	BYTE 1	ADDR OFFSET
PCI	Data	Reserved	PowerMgmtCtrl		54
PCI	PowerMgmtCap		NextItemPtr	CapId	50
PCI	Reserved				4C
PCI	Reserved				48
PCI	Reserved				44
PCI	Reserved				40
PCI	MaxLat	MinGnt	InterruptPin	InterruptLine	3C
PCI	Reserved				38
PCI	Reserved			CapPtr	34
PCI	ExpRomBaseAddress				30
PCI	SubsystemId		SubsystemVendorId		2C
PCI	Reserved				28
PCI	Reserved				24
PCI	Reserved				20
PCI	Reserved				1C

TABLE 20: Rubicon PCI Register Layout

CLK	BYTE 4	BYTE 3	BYTE 2	BYTE 1	ADDR OFFSET
PCI	Reserved				18
PCI	MemBaseAddress				14
PCI	IoBaseAddress				10
PCI	Reserved	HeaderType	LatencyTimer	CacheLineSize	0C
PCI	ClassCode			RevisionId	08
PCI	ConfigStatus		ConfigCommand		04
PCI	DeviceId		VendorId		00

11.10.6.1 CacheLineSize

Class..... PCI Configuration Registers
I/O Base Address ..... PCI device configuration header start
Address Offset..... 0x0C
Default Value 0x00
Access Rule..... Byte
Width ..... 8 bits

BIT	BIT NAME	R/W	BIT DESCRIPTION
7..0	CacheLineSize	R/W	Cache Line Size. The system BIOS writes the system's cache line size into CacheLineSize. The host system uses CacheLineSize to optimize PCI bus master operation (choosing the best memory command, etc.). The value in CacheLineSize represents the number of double words in a cache. CacheLineSize values must be a power of two, from 0x04 to 0x40 (giving a range of 16 to 256 bytes). CacheLineSize values which are not a power of two, between 4 and 64 are interpreted as 0x00.

11.10.6.2 CapId

Class..... PCI Configuration Registers, Power Management
I/O Base Address ..... PCI device configuration header start
Address Offset..... 0x50
Default Value 0x01
Access Rule..... Byte
Width ..... 8 bits

BIT	BIT NAME	R/W	BIT DESCRIPTION
7..0	CapId	R	Capabilities ID. CapId indicates the type of the capability data structure for the Rio. CapId is set to the value 0x01 to indicate a PCI Power Management structure.

11.10.6.3 CapPtr

Class..... PCI Configuration Registers
I/O Base Address ..... PCI device configuration header start
Address Offset..... 0x34
Default Value 0x50
Access Rule..... Byte
Width ..... 8 bits

BIT	BIT NAME	R/W	BIT DESCRIPTION
7..0	CapPtr	R	Capabilities Pointer. CapPtr indicates the beginning of a chain of registers which describe enhanced functions. CapPtr register returns 0x50, which is the address of the first in a series of power management registers.

11.10.6.4 ClassCode

Class..... PCI Configuration Registers
I/O Base Address ..... PCI device configuration header start
Address Offset..... 0x09
Default Value 0x020000
Access Rule..... Byte
Width ..... 24 bits

BIT	BIT NAME	R/W	BIT DESCRIPTION
23..0	ClassCode	R	Class Code. ClassCode identifies the general function of the PCI device. A value of 0x020000 indicates an Ethernet network controller.

11.10.6.5 ConfigCommand

Class..... PCI Configuration Registers

I/O Base Address ..... PCI device configuration header start

Address Offset..... 0x04

Default Value 0x0000

Access Rule..... Word

Width ..... 16 bits

ConfigCommand provides control over the Rio's ability to generate and respond to PCI cycles. When ConfigCommand is a logic 0, the Rio is logically disconnected from the PCI bus, except for configuration cycles.

BIT	BIT NAME	R/W	BIT DESCRIPTION
0	IoSpace	R/W	I/O Space. When IoSpace is a logic the Rio can respond to I/O space accesses (if the Rio is in the D0 power state).
1	MemorySpace	R/W	Memory Space. When MemorySpace, and the AddressDecodeEnable bit in the ExpRomBaseAddress register are both a logic 1, and if the Rio is in the D0 power state, the Rio is able to decode accesses to an Expansion ROM (if present).
2	BusMaster	R/W	Bus Master. When BusMaster is a logic 1 the Rio is able to initiate bus master cycles (if the adapter is in the D0 power state).
3	Reserved	N/A	Reserved for future use. Write as zero, ignore on read.
4	MWIEnable	R/W	Memory Write and Invalidate Enable. When MWIEnable is a logic 1 the Rio is permitted to use the MWI command.
5	Reserved	N/A	Reserved for future use. Write as zero, ignore on read.
6	ParityErrorResponse	R/W	Parity Error Response. When ParityErrorResponse is a logic 1 the Rio responds to parity errors as defined within the PCI specification. When ParityErrorResponse is a logic 0, the Rio ignores parity errors.
7	Reserved	N/A	Reserved for future use. Write as zero, ignore on read.
8	SERREnable	R/W	System Error Enable. When SERREnable is a logic 1, the SERRN signal is allowed to transition as appropriate. When SERREnable is a logic 0, the SERRN signal is a continuous logic 0.
15..9	Reserved	N/A	Reserved for future use. Write as zero, ignore on read.

11.10.6.6 ConfigStatus

Class..... PCI Configuration Registers

I/O Base Address ..... PCI device configuration header start

Address Offset..... 0x06

Default Value 0x0230

Access Rule..... Word

Width ..... 16 bits

ConfigStatus is used to record status information for PCI bus events. Read/write bits within ConfigStatus can only be set to a logic 0, not to a logic 1. Bits are set to a logic 0 by writing a logic 1 to the appropriate bit.

BIT	BIT NAME	R/W	BIT DESCRIPTION
3..0	Reserved	N/A	Reserved for future use. Write as zero, ignore on read.
4	Capabilities	R	Capabilities. Capabilities is a logic 1 to indicate a set of extended capabilities registers exists for the Rio. The CapPtr register indicates the first address location of the extended capabilities register set.
5	66MHzCapable	R	66MHz Capable. When 66MHzCapable is a logic 1 operation of the Rio PCI bus interface at 66MHz is supported.
6	Reserved	N/A	Reserved for future use. Write as zero, ignore on read.
7	FastBackToBack	R	Fast Back to Back. When FastBackToBack is a logic 1 the Rio when operating as a Target, supports fast back-to-back transactions as defined by the criteria in the section 3.4.2 of the PCI specification.
8	DataParityReported	R	Data Parity Reported. When DataParityReported is a logic 1, the Rio when operating as a Master, has detected the PERRN signal asserted, and the ParityErrorResponse bit in the ConfigCommand register as a logic 1.
10..9	DevselTiming	R	Device Select Timing. DevselTiming is used to encode the slowest time with which the Rio asserts the DEVSELN signal. A value of 0x1 for DevselTiming indicates support for “medium” speed DEVSELN assertion.
11	SignaledTargetAbort	R	Signaled Target Abort. The Rio sets SignaledTargetAbort to a logic 1 when the Rio terminates a bus transaction with target-abort.
12	ReceivedTargetAbort	R	Received Target Abort. The Rio sets ReceivedTargetAbort to a logic 1 when, operating as a bus master, a Rio bus transaction is terminated with target-abort.
13	ReceivedMasterAbort	R	Received Master Abort. The Rio sets ReceivedMasterAbort to a logic 1 when, operating as a bus master, a Rio bus transaction is terminated with master-abort.

BIT	BIT NAME	R/W	BIT DESCRIPTION
14	SignaledSystemError	R	Signaled System Error. When SignaledSystemError is a logic 1, the Rio asserts the SERRN signal.
15	DetectedParityError	R	Detected Parity Error. When DetectedParityError is a logic 1 the Rio has detected a parity error, regardless of whether parity error handling is enabled.

11.10.6.7 Data

Class..... PCI Configuration Registers
I/O Base Address ..... PCI device configuration header start
Address Offset..... 0x57
Default Value 0x0000
Access Rule..... Byte
Width ..... 8 bits

<table>
<tr><th>BIT</th><th>BIT NAME</th><th>R/W</th><th>BIT DESCRIPTION</th></tr>
<tr><td>7..0</td><td>Data</td><td>R</td><td>Data reports power consumption and dissipation of the Rio at worst case conditions. To properly interpret the value read from Data, it must be scaled by the factor indicated in the Data_Scale field of the PowerMgmtCtrl register. The value of Data depends on the value of the Data_Select field of the PowerMgmtCtrl register..
<table>
<tr><th>DATA_SELECT</th><th>DATA</th></tr>
<tr><td>0x0</td><td>40 * Data_Scale Watts
D0 Power Consumption</td></tr>
<tr><td>0x1</td><td>40 * Data_Scale Watts
D1 Power Consumption</td></tr>
<tr><td>0x2</td><td>40 * Data_Scale Watts
D2 Power Consumption</td></tr>
<tr><td>0x3</td><td>40 * Data_Scale Watts
D3 Power Consumption</td></tr>
<tr><td>0x4</td><td>40 * Data_Scale Watts
D4 Power Dissipated</td></tr>
<tr><td>0x5</td><td>40 * Data_Scale Watts
D5 Power Dissipated</td></tr>
<tr><td>0x6</td><td>40 * Data_Scale Watts
D6 Power Dissipated</td></tr>
<tr><td>0x7</td><td>40 * Data_Scale Watts
D7 Power Dissipated</td></tr>
<tr><td>0x8 through 0xF</td><td>0x00
Reserved.</td></tr>
</table>
</td></tr>
</table>

11.10.6.8 DeviceId

Class..... PCI Configuration Registers
I/O Base Address ..... PCI device configuration header start
Address Offset..... 0x02
Access Rule..... Word
Default Value 0x1021
Width ..... 16 bits

BIT	BIT NAME	R/W	BIT DESCRIPTION
15..0	DeviceId	R/W	Device ID. DeviceId contains the 16-bit device identifier for the Rio.

11.10.6.9 ExpRomBaseAddress

Class..... PCI Configuration Registers, Configuration

I/O Base Address ..... PCI device configuration header start

Address Offset..... 0x30

Default Value 0x00000000

Access Rule..... Double Word

Width 0x32

ExpRomBaseAddress defines the base address for an Expansion ROM which may be interfaced to the Rio.

BIT	BIT NAME	R/W	BIT DESCRIPTION
0	AddressDecodeEnable	R/W	Address Decode Enable. When AddressDecodeEnable is a logic 0 accesses to an Expansion ROM are disabled. When AddressDecodeEnable is a logic 1 and the MemorySpace bit in the ConfigCommand register is also a logic 1, accesses to an Expansion ROM are enabled.
14..1	Reserved	N/A	Reserved for future use. Write as zero, ignore on read.
31..15	RomBaseAddress	R/W	ROM Base Address. RomBaseAddress contains the expansion ROM base address, or the upper 16 bits (or 15 bits, depending on the state of the ExpRomSize bit in the AsicCtrl register) of the Expansion ROM address range. If the ExpRomSize bit in the AsicCtrl register is a logic 0, all 16 bits of RomBaseAddress are valid. If the ExpRomSize bit in the AsicCtrl register is a logic 1, bits 31 through 16 of RomBaseAddress are valid, with bit 15 ignored (set to a logic 0) during write operations.

11.10.6.10 HeaderType

Class..... PCI Configuration Registers
I/O Base Address ..... PCI device configuration header start
Address Offset..... 0x0E
Default Value 0x00
Access Rule..... Byte
Width ..... 8 bits

BIT	BIT NAME	R/W	BIT DESCRIPTION
7..0	HeaderType	R	Header Type. HeaderType is set to 0x00 identifying the Rio as a single-function PCI device and specifying the configuration register layout.

11.10.6.11 InterruptLine

Class..... PCI Configuration Registers
I/O Base Address ..... PCI device configuration header start
Address Offset..... 0x3C
Default Value 0x00
Access Rule..... Byte
Width ..... 8 bits

BIT	BIT NAME	R/W	BIT DESCRIPTION
7..0	InterruptLine	R/W	Interrupt Line. InterruptLine specifies the interrupt level used by the Rio. By setting InterruptLine the host system may configure the appropriate interrupt vector for its Interrupt Service Routine. For 80x86 processor based host systems, InterruptLine corresponds to the IRQ number (0x00 through 0x0F), with the value 0xFF corresponding to disabled interrupts.

11.10.6.12 InterruptPin

Class..... PCI Configuration Registers
I/O Base Address ..... PCI device configuration header start
Address Offset..... 0x3D
Default Value 0x01
Access Rule..... Byte
Width ..... 8 bits

BIT	BIT NAME	R/W	BIT DESCRIPTION
7..0	InterruptPin	R	Interrupt Pin. InterruptPin indicates which PCI interrupt signal the Rio will utilize. The Rio always utilizes the INTAN interrupt signal, corresponding to an InterruptPin value of 0x01.

11.10.6.13 IoBaseAddress

Class..... PCI Configuration Registers

I/O Base Address ..... PCI device configuration header start

Address Offset..... 0x10

Default Value 0x00000000

Access Rule..... Double Word

Width ..... 32 bits

The host uses IoBaseAddress to define the I/O base address for the Rio. PCI system requires that I/O base addresses be set as if the host system used 32-bit I/O addressing. The upper 24 bits of IoBaseAddress are accessible, indicating that the Rio requires 256 bytes in the host system I/O address space.

BIT	BIT NAME	R/W	BIT DESCRIPTION
0	IoBaseAddrInd	R/W	I/O Base Address Indicator. When IoBaseAddrInd is a logic 1, IoBaseAddress contains the valid I/O base address for the Rio.
7..1	Reserved	N/A	Reserved for future use. Write as zero, ignore on read.
31..8	IoBaseAddress	R/W	I/O Base Address. IoBaseAddress contains the 24 bit I/O base address value. With 24 bits, the Rio uses 256 bytes of I/O address space.

11.10.6.14 LatencyTimer

Class..... PCI Configuration Registers
I/O Base Address ..... PCI device configuration header start
Address Offset..... 0x0D
Default Value 0x00
Access Rule..... Byte
Width ..... 8 bits

BIT	BIT NAME	R/W	BIT DESCRIPTION
2..0	Reserved	N/A	Reserved for future use. Write as zero, ignore on read.
7..3	LatencyTimer	R/W	Latency Timer. LatencyTimer indicates, in increments of 8 bus clocks, the length of time which the Rio may hold the PCI bus in the presence of other bus requestors. Whenever the Rio asserts the FRAMEN signal, the latency timer is started. When the latency timer count expires, the Rio must relinquish the bus as soon as its GNTN signal has been de-asserted.

11.10.6.15 MaxLat

Class..... PCI Configuration Registers, Configuration
I/O Base Address ..... PCI device configuration header start
Address Offset..... 0x3F
Default Value 0x05
Access Rule..... Byte
Width ..... 8 bits

BIT	BIT NAME	R/W	BIT DESCRIPTION
7..0	MaxLat	R	Maximum Latency. MaxLat specifies, in 250 ns increments, how often the Rio requires bus access while operating as a bus master. The value for MaxLat is loaded from the Config-Parm field within an EEPROM during auto initialization of the Rio.

11.10.6.16 MemBaseAddress

Class..... PCI Configuration Registers

I/O Base Address ..... PCI device configuration header start

Address Offset..... 0x34

Default Value 0x00000000

Access Rule..... Double Word

Width ..... 32 bits

MemBaseAddress can be disabled via loading of the ConfigParm field from an EEPROM during auto-initialization of the Rio.

<table>
<tr><th>BIT</th><th>BIT NAME</th><th>R/W</th><th>BIT DESCRIPTION</th></tr>
<tr><td>0</td><td>MemBaseAddrInd</td><td>R/W</td><td>Memory Base Address Indicator. When MemBaseAddrInd is a logic 1, MemBaseAddress contains the valid memory base address.</td></tr>
<tr><td>2..1</td><td>MemMapType</td><td></td><td>Memory Map Type. MemMapType defines how the host system maps the Rio's registers within the host system memory space. Bit 2 of MemMapType is always a logic 0, while bit 1 is loaded from the Lower1Meg bit of the ConfigParm field within an EEPROM during auto initialization of the Rio.
<table>
<tr><th>BIT 1</th><th>BIT 0</th><th>ETHERNET FRAME TCI PRIORITY FIELD BIT</th></tr>
<tr><td>0</td><td>0</td><td>Anywhere within a 32 bit address space</td></tr>
<tr><td>0</td><td>1</td><td>Lower 1 megabyte of 32 bit address space</td></tr>
<tr><td>1</td><td>x</td><td>Undefined</td></tr>
</table>
</td></tr>
<tr><td>8..3</td><td>Reserved</td><td>N/A</td><td>Reserved for future use. Write as zero, ignore on read.</td></tr>
<tr><td>31..9</td><td>MemBaseAddress</td><td>R/W</td><td>Memory Base Address. MemBaseAddress contains the 23 bit memory base address value. With 23 bits, the Rio uses 512 bytes of I/O space.</td></tr>
</table>

11.10.6.17 MinGnt

Class..... PCI Configuration Registers
I/O Base Address ..... PCI device configuration header start
Address Offset..... 0x3E
Default Value 0x0A
Access Rule..... Byte
Width ..... 8 bits

BIT	BIT NAME	R/W	BIT DESCRIPTION
7..0	MinGnt	R	Minimum Grant Time. MinGnt specifies, in 250 ns increments, how long a burst period the Rio requires when operating as a bus master. The value for MinGnt is loaded from the Config-Parm field within an EEPROM during auto initialization of the Rio.

11.10.6.18 NextItemPtr

Class..... PCI Configuration Registers
I/O Base Address ..... PCI device configuration header start
Address Offset..... 0x51
Default Value 0x00
Access Rule..... Byte
Width ..... 8 bits

BIT	BIT NAME	R/W	BIT DESCRIPTION
7..0	NextItemPtr	R	Next Item Pointer. NextItemPtr indicates the next capability data structure in the capabilities list. NextItemPtr is set to the value 0x00 to indicate there are no further data structures.

11.10.6.19 PowerMgmtCap

Class..... PCI Configuration Registers
I/O Base Address ..... PCI device configuration header start
Address Offset..... 0x52
Default Value 0x7602
Access Rule..... Word
Width ..... 16 bits

<table>
<tr><th>BIT</th><th>BIT NAME</th><th>R/W</th><th>BIT DESCRIPTION</th></tr>
<tr><td>2..0</td><td>Version</td><td>R</td><td>Version. Version is set to 0x2, indicating PCI Bus Power Management Specification Revision 1.1.</td></tr>
<tr><td>8..3</td><td>Reserved</td><td>N/A</td><td>Reserved for future use. Write as zero, ignore on read.</td></tr>
<tr><td>9</td><td>D1Support</td><td>R</td><td>D1 Power State Support. When D1Support is a logic 1, the Rio supports the D1 power state (see section 11.4). D1Support is loaded from the ConfigParm field of an EEPROM during auto initialization of the Rio.</td></tr>
<tr><td>10</td><td>D2Support</td><td>R</td><td>D2 Power State Support. When D2Support is a logic 1, the Rio supports the D2 power state (see section 11.4). D2Support is loaded from the ConfigParm field of an EEPROM during auto initialization of the Rio.</td></tr>
<tr><td>15..11</td><td>PmeSupport</td><td>R</td><td>Power Management Event Support. PmeSupport indicates the power states from which the Rio is able to generate a power management event by asserting the WAKE signal. Each bit corresponds to a power state. A logic 1 in a particular bit position indicates that evens can be generated from the indicated power state.
<table>
<tr><th>BIT 15</th><th>BIT 14</th><th>BIT 13</th><th>BIT 12</th><th>BIT 11</th><th>POWER MANAGEMENT EVENT MAY BE GENERATED FROM STATE</th></tr>
<tr><td>x</td><td>x</td><td>x</td><td>x</td><td>1</td><td>D0</td></tr>
<tr><td>x</td><td>x</td><td>x</td><td>1</td><td>x</td><td>D1</td></tr>
<tr><td>x</td><td>x</td><td>1</td><td>x</td><td>x</td><td>D2</td></tr>
<tr><td>x</td><td>1</td><td>x</td><td>x</td><td>x</td><td>D3Hot</td></tr>
<tr><td>1</td><td>x</td><td>x</td><td>x</td><td>x</td><td>D3Cold</td></tr>
</table>
PmeSupport bits 14 and 11 are always set to a logic 1 while bits 12, 13, and 15 are loaded from the ConfigParm field of an EEPROM during auto initialization of the Rio.</td></tr>
</table>

11.10.6.20 PowerMgmtCtrl

Class..... PCI Configuration Registers
I/O Base Address ..... PCI device configuration header start
Address Offset..... 0x54
Default Value 0x0000
Access Rule..... Word
Width ..... 16 bits

BIT	BIT NAME	R/W	BIT DESCRIPTION
1..0	PowerState	R/W	Power State. PowerState indicates the current power state of the Rio. If PowerState is set to a value other than 0x0, the Rio will not respond to PCI I/O or memory cycles, nor will the Rio be able to generate PCI bus master cycles.

BIT 1	BIT 0	POWER STATE
0	0	D0
0	1	D1
1	0	D2
1	1	D3

BIT	BIT NAME	R/W	BIT DESCRIPTION
7..2	Reserved	N/A	Reserved for future use. Write as zero, ignore on read.
8	PmeEn	R/W	Power Management Event Enable. When PmeEn is a logic 1, the Rio is allowed to report wake events on the WAKE signal. The criteria for generating wake events is defined by the WakeEvent register. PmeEn is loaded from the ConfigParm field of an EEPROM during auto initialization of the Rio.
12..9	Data_Select	R/W	Data Select is used to select which data is to be reported through the Data register and **Data_Scale** field.
14..13	Data_Scale	R	Data Scale Only indicates the scaling factor to be used when interpreting the value of the Data register. The interpreation of the scale values is defined as follows:

DATA_SCALE	SCALE FACTOR
0x0	Unknown
0x1	0.1
0x2	0.01
0x3	0.001

BIT	BIT NAME	R/W	BIT DESCRIPTION
15	PmeStatus	R/W	Power Management Event Status. When PmeStatus is a logic 1 a wake event has occurred. PmeStatus may be a logic 1 regardless of the value of PmeEn. Writing a logic 1 to PmeStatus will set PmeStatus to a logic 0. Writing a logic 0 to PmeStatus has no effect.

11.10.6.21 RevisionId

Class..... PCI Configuration Registers
I/O Base Address ..... PCI device configuration header start
Address Offset..... 0x08
Default Value 0x05
Access Rule..... Byte
Width ..... 8 bits

BIT	BIT NAME	R/W	BIT DESCRIPTION
7..0	RevisionId	R	Revision ID. RevisionId contains a revision code for the Rio.

11.10.6.22 SubsystemId

Class..... PCI Configuration Registers
I/O Base Address ..... PCI device configuration header start
Address Offset..... 0x2E
Default Value 0x0000
Access Rule..... Byte
Width ..... 16 bits

BIT	BIT NAME	R/W	BIT DESCRIPTION
15..0	SubsystemId	R	Subsystem ID. SubsystemId contains the value loaded from the ConfigParm field within an EEPROM during auto initialization of the Rio.

11.10.6.23 SubsystemVendorId

Class..... PCI Configuration Registers
I/O Base Address ..... PCI device configuration header start
Address Offset..... 0x2C
Default Value 0x0000
Access Rule..... Word
Width ..... 16 bits

BIT	BIT NAME	R/W	BIT DESCRIPTION
15..0	SubsystemVendorId	R	Subsystem Vendor ID. SubsystemVendorId contains the value loaded from the ConfigParm field within an EEPROM during auto initialization of the Rio.

11.10.6.24 VendorId

Class..... PCI Configuration Registers
I/O Base Address ..... PCI device configuration header start
Address Offset..... 0x00
Default Value 0x13F0
Access Rule..... Word
Width ..... 16 bits

BIT	BIT NAME	R/W	BIT DESCRIPTION
15..0	VendorId	R	Vendor ID. VendorId contains the unique 16-bit manufacturer's ID as allocated by the PCI Special Interest Group. Sundance Technology's manufacturer ID is 0x13F0.

11.10.7 EEPROM Fields

Table 21 summarizes the layout of the data stored in an EEPROM connected to the Rio. Most defined EEPROM fields are read from the EEPROM and loaded into unique register bit positions within the Rio during auto initialization (see 11.10.5.1).

In Table 21, all locations marked “Unused” and all locations not shown are not utilized by the Rio..

TABLE 21: Rio EEPROM Field Layout

16 BIT WORD	ADDR OFFSET
StationAddress	12
StationAddress	11
StationAddress	10
Unused	F
Unused	E
Unused	D
Unused	C
Unused	B
Unused	A
Unused	9
Unused	8
Unused	7
Unused	6
Unused	5
Unused	4
SubsystemId	3
SubsystemVendorId	2
AsicCtrl	1
ConfigParm	0

11.10.7.1 AsicCtrl

Class..... EEPROM Fields

I/O Base Address ..... 0x00, accessed via the EepromCtrl register

Address Offset..... 0x01

Access Rule..... Word

Width ..... 16 bits

ASIC Control supplies the value for several bits of the AsicCtrl register and the WakeEvent register.

BIT	BIT NAME	BIT DESCRIPTION
0	Reserved	Reserved for future use. Write as zero, ignore on read.
1	ExpRomSize	Expansion ROM Size. ExpRomSize corresponds to the ExpRomSize bit in the AsicCtrl register.
2	Reserved	Reserved for future use. Write as zero, ignore on read.
3	Reserved	Reserved for future use. Write as zero, ignore on read.
4	PhySpeed10	Physical Layer Device Speed 10. PhySpeed10 corresponds to the PhySpeed10 bit in the AsicCtrl register.
5	PhySpeed100	Physical Layer Device Speed 100. PhySpeed100 corresponds to the PhySpeed100 bit in the AsicCtrl register.
6	PhySpeed1000	Physical Layer Device Speed 1000. PhySpeed1000 corresponds to the PhySpeed1000 bit AsicCtrl register.
7	PhyMedia	Physical Layer Device Media. PhyMedia corresponds to the PhyMedia bit in the AsicCtrl register.
14..8	Reserved	Reserved for future use. Write as zero, ignore on read.
15	WakeOnLanPolarity	Wake On LAN Polarity. WakeOnLanPolarity corresponds to the WakeOnLanEnable bit in the WakeEvent register.

11.10.7.2 ConfigParm

Class..... EEPROM Fields
I/O Base Address ..... 0x00, accessed via the EepromCtrl register
Address Offset..... 0x00
Access Rule..... Word
Width ..... 16 bits

BIT	BIT NAME	BIT DESCRIPTION
0	FastBackToBack	Fast Back to Back. FastBackToBack corresponds to the FastBackToBack bit of the ConfigStatus register.
1	Lower1Meg	Lower 1 Megabyte. Lower1Meg corresponds to bit 1 of the MemMapType field in the MemBaseAddress register.
2	DisableMemBase	Disable Memory Base Address Register. DisableMemBase does not correspond directly to any register accessible by the host system. If DisableMemBase is a logic 1 during auto initialization of the Rio, the MemBaseAddress register will be disabled. When disabled, the value returned when the MemBaseAddress register is read is undefined.
3	D3ColdPme	D3 Cold Power Management Event. D3ColdPme corresponds to bit 15 of the PmeSupport field within the PowerMgmtCap register.
4	D1Support	D1 Power State Support. D1Support corresponds to the D1Support bit of the PowerMgmtCap register, and bit 12 of the PmeSupport field within the PowerMgmtCap register.
5	D2Support	D2 Power State Support. D2Support corresponds to the D2Support bit of the PowerMgmtCap register, and bit 13 of the PmeSupport field within the PowerMgmtCap register.
6	PmeEn	Power Management Event Enable. PmeEn corresponds to the PmeEn bit in the PowerMgmtCtrl register.
10..7	MinGnt	Minimum Grant. MinGnt corresponds to bits 4 through 1 of the MinGnt register.
15..11	MaxLat	Maximum Latency. MaxLat corresponds to bits 5 through 1 of the MaxLat register.

11.10.7.3 StationAddress

Class..... EEPROM Fields
I/O Base Address ..... 0x00, accessed via the EepromCtrl register
Address Offset..... 0x10
Access Rule..... Byte
Width ..... 48 bits

BIT	BIT NAME	BIT DESCRIPTION
47..0	StationAddress	Station Address. StationAddress corresponds to the StationAddress register.

11.10.7.4 SubsystemId

Class..... EEPROM Fields
I/O Base Address ..... 0x00, accessed via the EepromCtrl register
Address Offset..... 0x06
Access Rule..... Word
Width ..... 16 bits

BIT	BIT NAME	BIT DESCRIPTION
15..0	SubsystemId	Subsystem ID. SubsystemId corresponds to the SubsystemId register.

11.10.7.5 SubsystemVendorId

Class..... EEPROM Fields
I/O Base Address ..... 0x00, accessed via the EepromCtrl register
Address Offset..... 0x02
Access Rule..... Word
Width ..... 16 bits

BIT	BIT NAME	BIT DESCRIPTION
15..0	SubsystemVendorId	Subsystem Vendor ID. SubsystemVendorId corresponds to the SubsystemVendorId register.

11.10.8 PCS Management Registers

The Rio implementation of the IEEE 1000BASE-X Physical Coding Sublayer (or PCS) includes several management registers, defined in the IEEE 802.3 standard clause 37.2.5.1. These registers can only be accessed if the Rio is in TBI mode (see the GMII signal definition). The PCS management registers are similar to registers which would be present in external PHY devices incorporating a 1000BASE-X PCS implementation. The PCS management registers are accessed similarly to external PHY device registers using the Rio PhyCtrl register MgmtClk, MgmtData, and MgmtDir bits (see IEEE 802.3 1998 Edition for details on accessing management registers). When accessing the PCS management registers within the Rio, the PHY Address for all register accesses is 0x01.

Table 22 shows a layout of the registers implemented in the Rio 1000BASE-X PCS implementation.

TABLE 22: Rio 1000BASE-X PCS Management Registers

16 BIT WORD	ADDR OFFSET
ExtendedStatus	F
Unused	E
Unused	D
Unused	C
Unused	B
Unused	A
Unused	9
LinkPartnerNextPage	8
NextPage	7
Expansion	6
LinkPartnerBasePage	5
Advertisement	4
Unused	3
Unused	2
Status	1
Control	0

11.10.8.1 Advertisement

Class..... PCS Management
PHY Address 0x01
Register 0x04
Default Value 0x01E0
Width ..... 16 bits

<table>
<tr><th>BIT</th><th>BIT NAME</th><th>R/W</th><th>BIT DESCRIPTION</th></tr>
<tr><td>15</td><td>Next Page</td><td>R/W</td><td>1 = indicates additional Next Pages follow
0 = indicates last page</td></tr>
<tr><td>14</td><td>Reserved</td><td>R</td><td>Reserved for future use. Write as zero, ignore on read.</td></tr>
<tr><td>13..12</td><td>Remote Fault</td><td>R/W</td><td>
<table>
<tr><th>BIT 13</th><th>BIT 12</th><th>REMOTE FAULT</th></tr>
<tr><td>0</td><td>0</td><td>No error, link OK (default)</td></tr>
<tr><td>1</td><td>0</td><td>Offline</td></tr>
<tr><td>0</td><td>1</td><td>Link failure</td></tr>
<tr><td>1</td><td>1</td><td>Auto-negotiation failure</td></tr>
</table>
</td></tr>
<tr><td>11..9</td><td>Reserved</td><td>R</td><td>Reserved for future use. Write as zero, ignore on read.</td></tr>
<tr><td>8..7</td><td>Pause</td><td>R/W</td><td>
<table>
<tr><th>BIT 8</th><th>BIT 7</th><th>PAUSE</th></tr>
<tr><td>0</td><td>0</td><td>No PAUSE</td></tr>
<tr><td>1</td><td>0</td><td>Asymmetric PAUSE toward link partner</td></tr>
<tr><td>0</td><td>1</td><td>Symmetric PAUSE</td></tr>
<tr><td>1</td><td>1</td><td>Both Symmetric PAUSE and Asymmetric PAUSE toward local device</td></tr>
</table>
</td></tr>
<tr><td>6</td><td>Half Duplex</td><td>R/W</td><td>1 = indicates support for half duplex
0 = indicates no support for half duplex</td></tr>
<tr><td>5</td><td>Full Duplex</td><td>R/W</td><td>1 = indicates support for full duplex
0 = indicates no support for full duplex</td></tr>
<tr><td>4..0</td><td>Reserved</td><td>R</td><td>Reserved for future use. Write as zero, ignore on read.</td></tr>
</table>

11.10.8.2 Control

Class..... PCS Management

PHY Address 0x01

Register 0x00

Default Value 0x1140

Width ..... 16 bits

The Rio PCS Management Control register differs from the Control Register definition in the IEEE 802.3 standard as follows:

- Bit 13 is read only since the Rio only implements 1000 Mbp/s operation.
- Bit 11 is read only since the Rio does not implement separate power down of the PCS functionality.
- Bit 10 is read only since the Rio does not implement GMII electrical isolation.
- Bit 6 is read only since the Rio only implements 1000 Mbp/s operation.

BIT	BIT NAME	R/W	BIT DESCRIPTION
15	Reset	R/W	1 = PHY Reset (reset the PCS implementation within Rio) 0 = normal operation
14	Loopback	R/W	1 = enable loopback mode (asserts the MDC/EWRAP signal HIGH to indicate to the PHY device that transmit data should be wrapped to the receive interface). 0 = disable loopback mode
13	SpeedSelection (LSB)	R	0 = combined with bit 6 (bit 6 = 1) indicates 1000 Mb/s operation only.
12	Auto-Negotiation Enable	R/W	1 = Enable Auto-Negotiation Process 0 = Disable Auto-Negotiatio Process
11	Power Down	R	0 = normal operation
10	Isolate	R	0 = normal operation
9	Restart Auto-Negotiation	R/W	1 = Restart Auto-Negotiation Process 0 = normal operation
8	Duplex Mode	R/W	1 = Full Duplex 0 = Half Duplex
7	Collision Test	R/W	1 = enable COL signal test 0 = disable COL signal test
6	Speed Selection (MSB)	R	1 = combined with bit 13 (bit 13 = 0) indicates 1000 Mb/s operation only.
5..0	Reserved	R	Reserved for future use. Write as zero, ignore on read.

11.10.8.3 Expansion

Class..... PCS Management
PHY Address 0x01
Register 0x06
Default Value 0x0004
Width ..... 16 bits

BIT	BIT NAME	R/W	BIT DESCRIPTION
15..3	Reserved	R	Reserved for future use. Write as zero, ignore on read..
2	Next Page Able	R	1 = indicates next page able 0 = indicates not next page able
1	Page Received	R	1 = indicates a new page has been received 0 = indicates a new page has not yet been received
0	Reserved	R	Reserved for future use. Write as zero, ignore on read.

11.10.8.4 ExtendedStatus

Class..... PCS Management
PHY Address 0x01
Register 0x0F
Default Value 0xC000
Width ..... 16 bits

BIT	BIT NAME	R/W	BIT DESCRIPTION
15	1000BASE-X Full Duplex	R	1 = PHY able to perform full duplex 1000BASE-X 0 = PHY not able to perform full duplex 1000BASE-X
14	1000BASE-X Half Duplex	R	1 = PHY able to perform half duplex 1000BASE-X 0 = PHY not able to perform half duplex 1000BASE-X
13	1000BASE-T Full Duplex	R	1 = PHY able to perform full duplex 1000BASE-T 0 = PHY not able to perform full duplex 1000BASE-T
12	1000BASE-T Half Duplex	R	1 = PHY able to perform half duplex 1000BASE-T 0 = PHY not able to perform half duplex 1000BASE-T
11..0	Reserved	R	Reserved for future use. Write as zero, ignore on read.

11.10.8.5 LinkPartnerBasePage

Class..... PCS Management
PHY Address 0x01
Register 0x05
Default Value 0x0000
Width ..... 16 bits

<table>
<tr><th>BIT</th><th>BIT NAME</th><th>R/W</th><th>BIT DESCRIPTION</th></tr>
<tr><td>15</td><td>Next Page</td><td>R</td><td>1 = indicates additional Next Pages follow
0 = indicates last page</td></tr>
<tr><td>14</td><td>Acknowledge</td><td>R</td><td>1 = indicates received link partner’s code word
0 = indicates link partner’s code word not yet received</td></tr>
<tr><td>13..12</td><td>Remote Fault</td><td>R</td><td>
<table>
<tr><th>BIT 13</th><th>BIT 12</th><th>REMOTE FAULT</th></tr>
<tr><td>0</td><td>0</td><td>No error, link OK (default)</td></tr>
<tr><td>1</td><td>0</td><td>Offline</td></tr>
<tr><td>0</td><td>1</td><td>Link failure</td></tr>
<tr><td>1</td><td>1</td><td>Auto-negotiation failure</td></tr>
</table>
</td></tr>
<tr><td>11..9</td><td>Reserved</td><td>R</td><td>Reserved for future use. Write as zero, ignore on read.</td></tr>
<tr><td>8..7</td><td>Pause</td><td>R</td><td>
<table>
<tr><th>BIT 8</th><th>BIT 7</th><th>PAUSE</th></tr>
<tr><td>0</td><td>0</td><td>No PAUSE</td></tr>
<tr><td>1</td><td>0</td><td>Asymmetric PAUSE toward link partner</td></tr>
<tr><td>0</td><td>1</td><td>Symmetric PAUSE</td></tr>
<tr><td>1</td><td>1</td><td>Both Symmetric PAUSE and Asymmetric PAUSE toward local device</td></tr>
</table>
</td></tr>
<tr><td>6</td><td>Half Duplex</td><td>R</td><td>1 = indicates link partner supports half duplex
0 = indicates link partner does not support half duplex</td></tr>
<tr><td>5</td><td>Full Duplex</td><td>R</td><td>1 = indicates link partner supports full duplex
0 = indicates link partner does not support full duplex</td></tr>
<tr><td>4..0</td><td>Reserved</td><td>R</td><td>Reserved for future use. Write as zero, ignore on read.</td></tr>
</table>

11.10.8.6 LinkPartnerNextPage

Class..... PCS Management
PHY Address 0x01
Register 0x08
Default Value 0x0000
Width ..... 16 bits

BIT	BIT NAME	R/W	BIT DESCRIPTION
15	Next Page	R	1 = indicates additional Next Pages follow 0 = indicates last page
14	Acknowledge	R	1 = indicates received link partner's code word 0 = indicates link partner's code word not yet received
13	Message Page	R	1 = indicates a Message Page 0 = indicates an Unformatted Page
12	Acknowledge 2	R	1 = indicates able to comply with message 0 = indicates not able to comply with message
11	Toggle	R	1 = indicates previous value of toggle bit in code word = 0 0 = indicates previous value of toggle bit in code word = 1
10..0	Message/Unformatted Code Field	R	11 bit message code or unformatted code.

11.10.8.7 NextPage

Class..... PCS Management
PHY Address 0x01
Register 0x07
Default Value 0x2001
Width ..... 16 bits

BIT	BIT NAME	R/W	BIT DESCRIPTION
15	Next Page	R/W	1 = indicates additional Next Pages follow 0 = indicates last page
14	Reserved	R	Reserved for future use. Write as zero, ignore on read.
13	Message Page	R/W	1 = indicates a Message Page 0 = indicates an Unformatted Page
12	Acknowledge 2	R/W	1 = indicates able to comply with message 0 = indicates not able to comply with message
11	Toggle	R	1 = indicates previous value of toggle bit in code word = 0 0 = indicates previous value of toggle bit in code word = 1
10..0	Message/Unformatted Code Field	R/W	11 bit message code or unformatted code.

11.10.8.8 Status

Class..... PCS Management
PHY Address 0x01
Register 0x01
Default Value 0x0109
Width ..... 16 bits

BIT	BIT NAME	R/W	BIT DESCRIPTION
15	100BASE-T4	R	0 = Not able to perform 100BASE-T4
14	100BASE-X Full Duplex	R	0 = Not able to perform 100BASE-X Full Duplex
13	100BASE-X Half Duplex	R	0 = Not able to perform 100BASE-X Half Duplex
12	10 Mb/s Full Duplex	R	0 = Not able to perform 10 Mb/s Full Duplex
11	10 Mb/s Half Duplex	R	0 = Not able to perform 10 Mb/s Half Duplex
10	100BASE-T2 Full Duplex	R	0 = Not able to perform 100BASE-T2 Full Duplex
9	100BASE-T2 Half Duplex	R	0 = Not able to perform 100BASE-T2 Half Duplex
8	Extended Status	R	1 = Extended status information in Register 15
7	Reserved	R	Reserved for future use. Write as zero, ignore on read.
6	MF Preamble Suppression	R	0 = will not accept management frames with preamble suppressed
5	Auto-Negotiation Complete	R	1 = Auto-Negotiation process completed 0 = Auto-Negotiation process not completed
4	Remote Fault	R	1 = remote fault condition detected 0 = no remote fault condition detected
3	Auto-Negotiation Ability	R	1 = able to perform Auto-Negotiation 0 = not able to perform Auto-Negotiation
2	Link Status	R	1 = link is up 0 = link is down
1	Jabber Detect	R	0 = no jabber condition detected
0	Extended Capability	R	1 = extended register capabilities

12.0 Signal Requirements

12.1 Absolute Maximum Ratings

Storage Temperature -65ºC to +150ºC

Ambient Temperature -65ºC to +70ºC

Supply Voltage -0.3V to +?V

Environmental stresses above those listed in Absolute Maximum Ratings may cause permanent damage resulting in device failure. Functionality at or above the limits listed below is not guaranteed. Exposure to the environmental stress at the levels listed below for extended periods may adversely affect device reliability.

12.2 Operating Ranges

Commercial Devices

Temperature (T_A) 0ºC to +70ºC

Supply Voltages (V_{CC} to GND) +3.3V ±5%

Input voltages +5V ±5%

Operating ranges define the limits of guaranteed device functionality.

12.3 DC characteristics

DC characteristics are defined over commercial operating ranges unless specified otherwise.

PARAMETER SYMBOL	PARAMETER DESCRIPTION	TEST CONDITIONS	MIN	MAX	UNIT
PIN TYPE IT (TTL, PCI INPUT BUFFER 5V SIGNALLING)					
V_{IH}	Input high voltage			2	V
V_{IL}	Input low voltage			0.8	V
I_{IH}	Input HIGH leakage current	V_{IN} = 2.7V		10	µA
I_{IL}	Input LOW leakage current	V_{IN} = 0.5V		-10	µA
PIN TYPE ITU (TTL, PCI INPUT BUFFER 5V SIGNALLING WITH PULL UP)					
V_{IH}	Input high voltage			2	V
V_{IL}	Input low voltage			0.8	V
I_{IH}	Input HIGH leakage current	V_{IN} = 2.7V		10	µA
I_{IL}	Input LOW leakage current	V_{IN} = 0.5V		-10	µA
PIN TYPE ITD (TTL, PCI INPUT BUFFER 5V SIGNALLING WITH PULL DOWN)					
V_{IH}	Input high voltage			2	V
V_{IL}	Input low voltage			0.8	V

TABLE 23: DC Characteristics

PARAMETER SYMBOL	PARAMETER DESCRIPTION	TEST CONDITIONS	MIN	MAX	UNIT
I_{IH}	Input HIGH leakage current	V_{IN} = 2.7V		10	µA
I_{IL}	Input LOW leakage current	V_{IN} = 0.5V		-10	µA
PIN TYPE OT4/OC4 (TTL, CMOS OUTPUT BUFFER)					
V_{OH}	Output high voltage	I_{OH} = -4mA	2.4		V
V_{OL}	Output low voltage	I_{OL} = 4mA		0.4	V
I_{OZ}	Output leakage current			±10	µA
PIN TYPE OP3 (PCI OUTPUT BUFFER 5V SIGNALLING)					
V_{OH}	Output high voltage	I_{OH} = -2mA	2.4		V
V_{OL}	Output low voltage	I_{OL} = 3mA		0.55	V
I_{OZ}	Output leakage current			±10	µA
PIN TYPE OP6 (PCI OUTPUT BUFFER 5V SIGNALLING WITH PULL UP)					
V_{OH}	Output high voltage	I_{OH} = -2mA	2.4		V
V_{OL}	Output low voltage	I_{OL} = 6mA		0.55	V
I_{OZ}	Output leakage current			±10	µA

TABLE 23: DC Characteristics

PARAMETER SYMBOL	PARAMETER DESCRIPTION	TEST CONDITIONS	MIN	MAX	UNIT
PIN TYPE OD6 (OPEN DRAIN OUTPUT BUFFER)					
V_{OL}	Output low voltage	I_{OL} = 6mA		0.4	V
I_{OZ}	Output leakage current			±10	µA

TABLE 23: DC Characteristics

PIN TYPE	PINS
PCI INTERFACE	
IT	RSTN, PCICLK, GNTN, IDSEL, ACK64N, VDET
IT/OP3	AD, CBEN, PAR, PAR64
ITU/OP6	FRAMEN, IRDYN, TRDYN, DEVSELN, STOPN, PERRN
OD6	INTAN, WAKE, SERRN
OP3	REQN, REQ64N
EXPANSION ROM INTERFACE	
ITU/OT4	ED
OT4	EWEN, EOEN, EA
EEPROM INTERFACE	
IT	EA5/EEDO
OT4	EA3/EEDI, EA4/EESK, EECS
GMII INTERFACE	
IT	CRS/SIGDET, COL, RXER/RXD9, RXDV/RXD8, RXD, RXCLK0/RXCLK, RXCLK1
OT4	GTXCLK, TXD, TXEN/TXD8, TXER/TXD9, MDC/EWRAP
IT/OT4	MDIO

TABLE 24: Pin Type Assignment

PIN TYPE	PINS
MISC	
IT	PHYLNK10N, PHYDPLXN, TEST, GMII
ITU/OT4	EA6/GPIO0, EA7/GPIO1
OT4	RSTOUT
OD8	EA0/LEDDPLXN, EA, EA2/LEDPWRN
OC4	CLK125

TABLE 24: Pin Type Assignment

12.4 AC Characteristics

PARAMETER SYMBOL	PARAMETER DESCRIPTION	TEST CONDITIONS	MIN	MAX	UNIT
PCI INTERFACE					
T_{rc}	RSTN cycle		300	-	-
T_{cc}	PCICLK cycle		30	-	-
T_{ch}	PCICLK high		11	-	-
T_{cl}	PCICLK low		11	-	-
T_{rv}	PCICLK rise to bused signal valid		2	11	-
T_{rvp}	PCICLK rise to REQN, GNTN valid		2	12	-
T_{rzo}	PCICLK rise to signal on		2	-	-
T_{roz}	PCICLK rise to signal off		-	28	-
T_{su}	bused signal setup wrt PCI-CLK rise		7	-	-
T_{sup1}	GNTN setup wrt PCICLK rise		10	-	-
T_{sup2}	REQN setup wrt PCICLK rise		12	-	-
T_{hd}	signal hold wrt PCICLK rise		0	-	-
T_{rstoff}	RSTN low to output signal float		-	40	-

TABLE 25: Switching Characteristics

PARAMETER SYMBOL	PARAMETER DESCRIPTION	TEST CONDITIONS	MIN	MAX	UNIT
EXPANSION ROM INTERFACE - READ					
T_{adv}	ED valid from EA stable		0	150	-
T_{odv}	ED valid from EOEN low		0	70	-
T_{dvz}	ED tri-stated from EOEN high		0	40	-
EXPANSION ROM INTERFACE - LOAD					
T_{as}/T_{os}	EA, EOEN setup wrt EWEN fall		0		-
T_{ah}	EA hold wrt EWEN fall		50	-	-
T_{ds}	ED setup wrt EWEN rise		35	-	-
T_{dh}/T_{oh}	ED, EOEN hold wrt EWEN fall		0		-
T_{wh}	EWEN write cycle high		100	-	-
T_{wl}	EWEN write cycle low		90	-	-
EEPROM INTERFACE					
T_{skc}	EESK cycle		1	-	us
T_{skh}	EESK high		250	-	ns
T_{skl}	EESK low		250	-	ns
T_{cs}	EECS low		250	-	-
T_{pd}	EEDI valid wrt EESK rise		100	-	-
T_{csk}	EECS setup wrt EESK rise		50	-	-
T_{csh}	EECS hold wrt EESK fall		0	-	-
T_{dos}	EEDO setup wrt EESK rise		70	500	-
T_{doh}	EEDO hold wrt EESK rise		-	500	-
GMII INTERFACE - TRANSMIT					
T_{cc}	TXCLK cycle		-	-	40T

TABLE 25: Switching Characteristics

PARAMETER SYMBOL	PARAMETER DESCRIPTION	TEST CONDITIONS	MIN	MAX	UNIT
T_{ch}	TXCLK high		14T	26T	-
T_{cl}	TXCLK low	T = 1 when 100Mb/s; 10 when 10Mb/s	14T	26T	-
T_{rv}	TXCLK rise to TXD, TXEN valid			20	
T_{rh}	TXD, TXEN hold after TXCLK rise		5	-	-
GMII INTERFACE - RECEIVE					
T_{cc}	RXCLK cycle		-	-	40T
T_{ch}	RXCLK high		14T	26T	-
T_{cl}	RXCLK low	T = 1 when 100Mb/s; 10 when 10Mb/s	14T	26T	-
T_{su}	RXD,RXER,RXDV setup wrt RXCLK rise		10	-	-
T_{hd}	RXD,RXER,RXDV hold wrt RXCLK rise		5	-	-
GMII INTERFACE - MANAGEMENT					
T_{cc}	MDC cycle		400	-	-
T_{ch}	MDC high		160	-	-
T_{cl}	MDC low		160	-	-
T_{su}	MDIO setup wrt MDC rise		10	-	-
T_{hd}	MDIO hold wrt MDC rise		10	-	-
T_{rv}	MDC rise to MDIO valid		-	20	-
MISC INTERFACE					
T_{cc}	CLK25 cycle		-	-	40
T_{ch}	CLK25 high		16	24	-
T_{cl}	CLK25 low		16	24	-

TABLE 25: Switching Characteristics

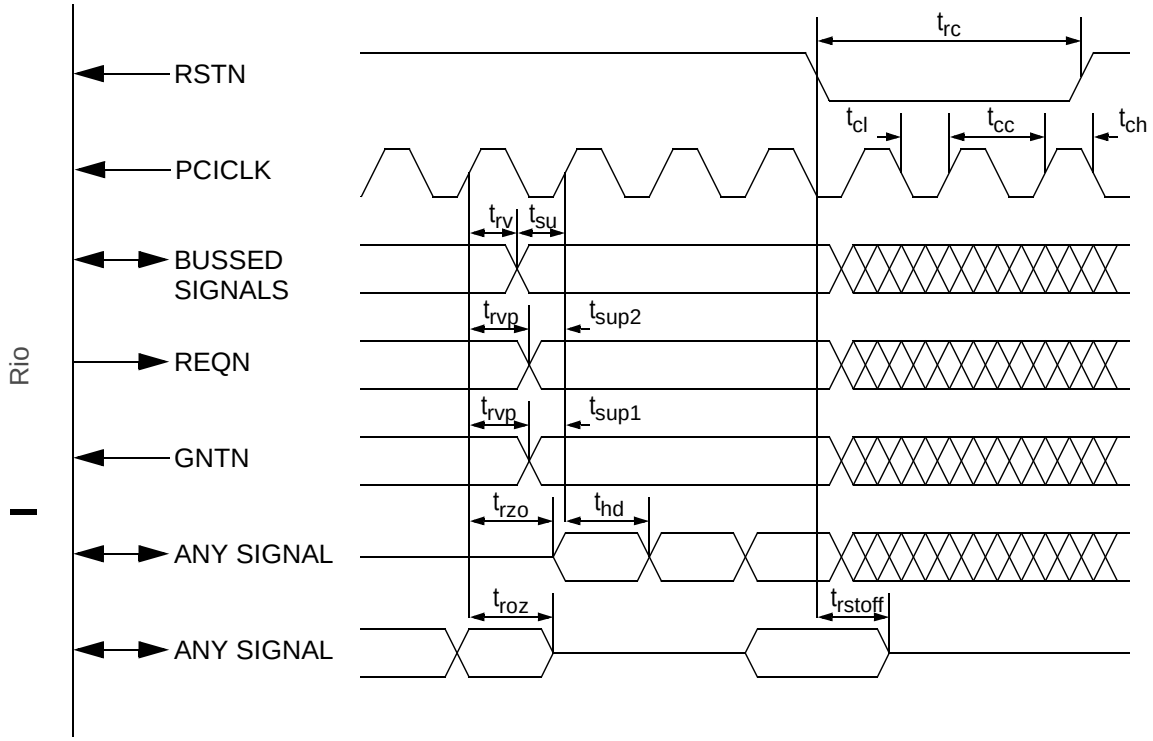


FIGURE 6: PCI Switching Characteristics

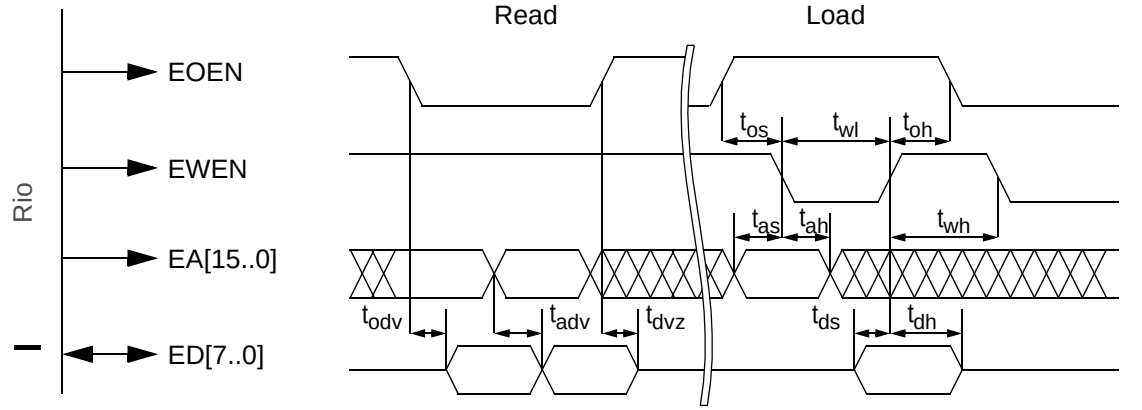


FIGURE 7: Expansion ROM Switching Characteristics

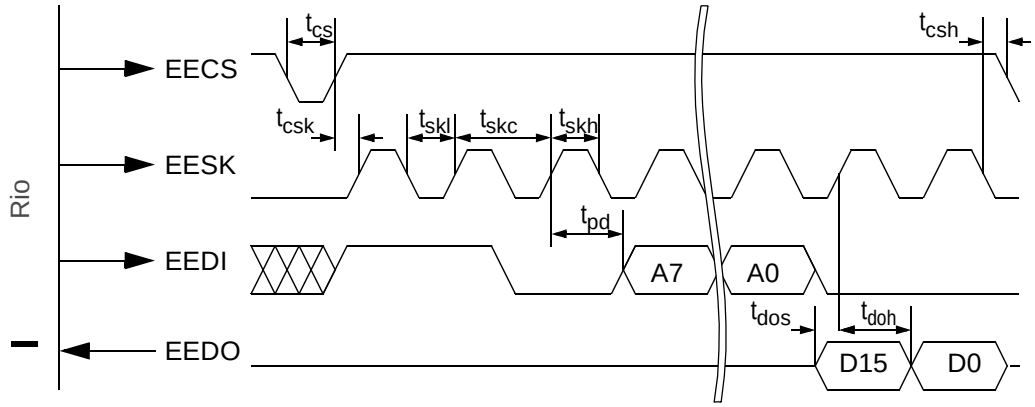


FIGURE 8: EEPROM Switching Characteristics

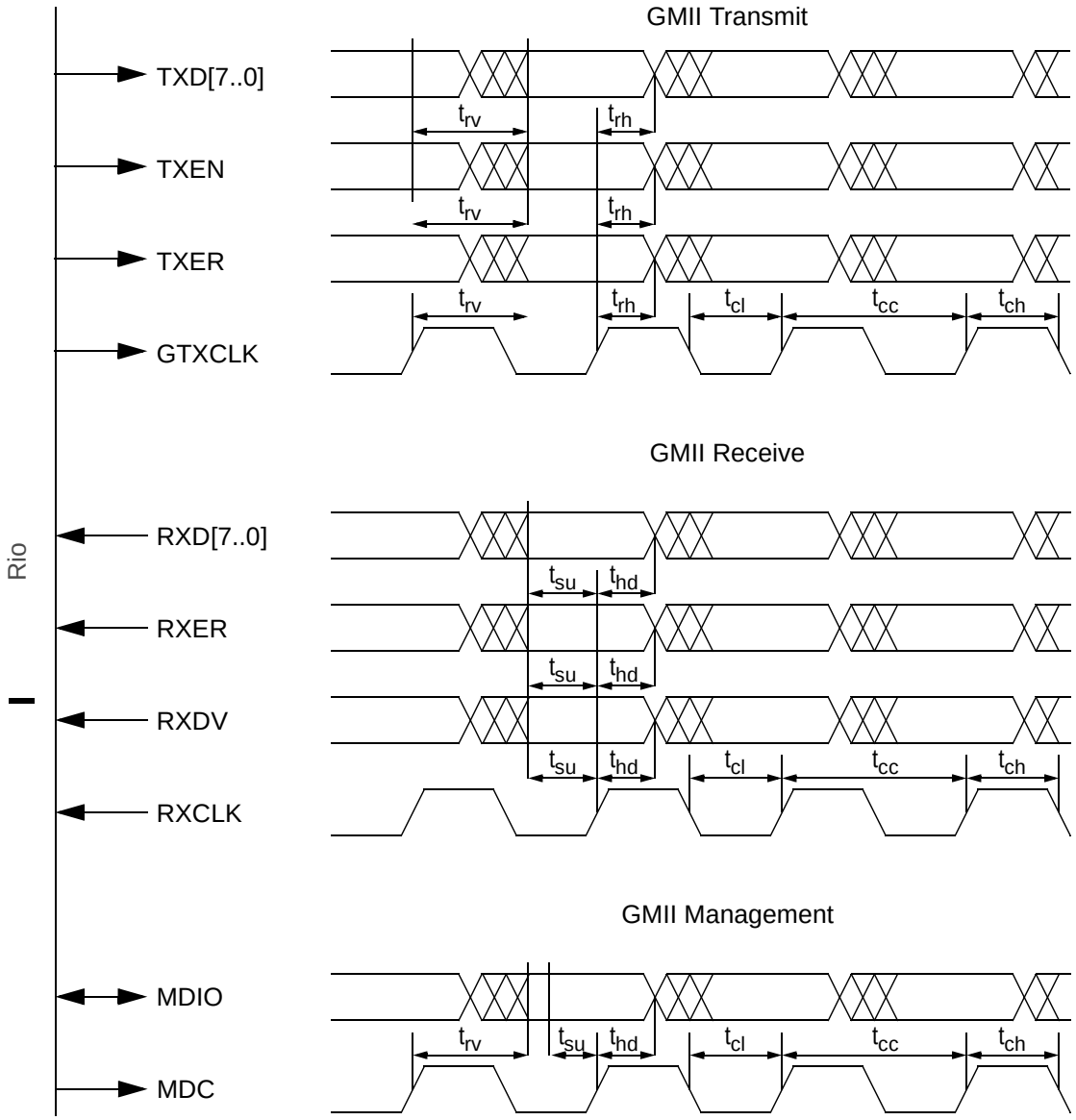
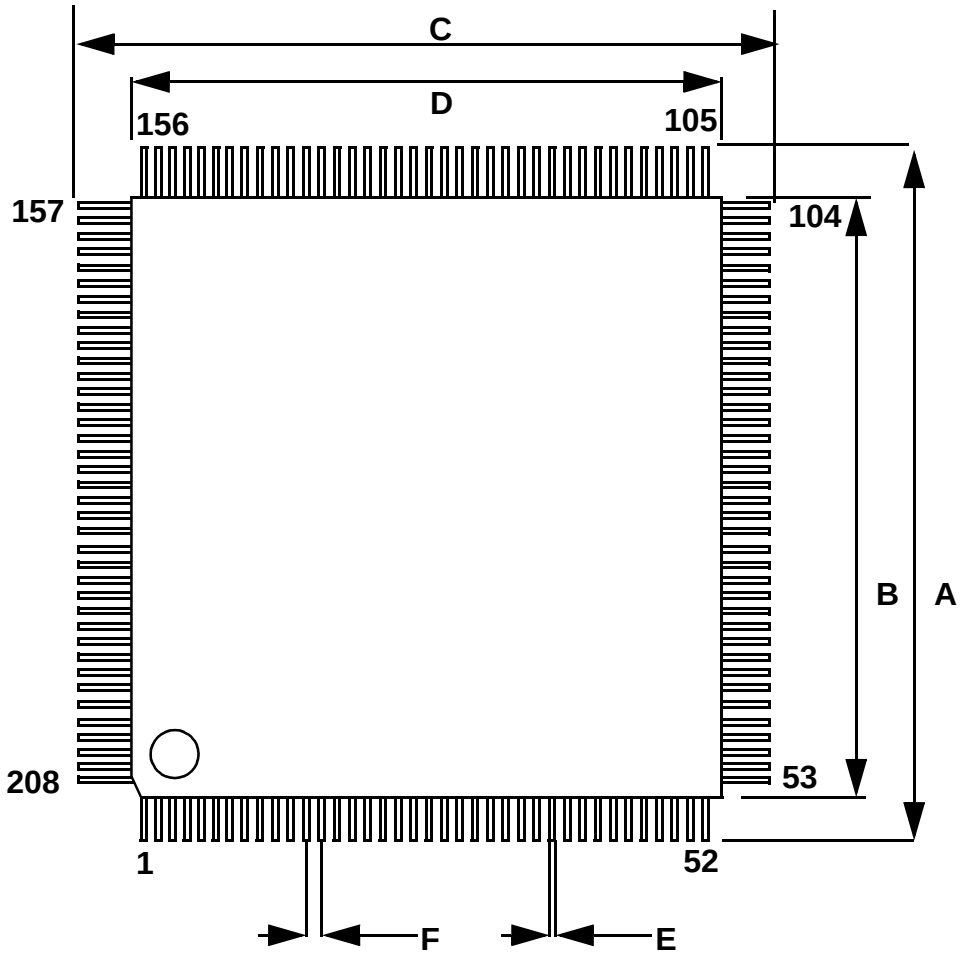


FIGURE 9: MII Switching Characteristics

13.0 Physical Dimensions



SYMBOL	DIMENSION IN MILLIMETERS		
	MINIMUM	NOMINAL	MAXIMUM
A	30.35	30.60	30.85
B	27.90	28.00	28.10
C	30.35	30.60	30.85
D	27.9	28.00	28.10
E	0.18		0.28
F		0.50	
Thickness	3.92		4.07

TABLE 26: Rio Mechanical Dimensions