

DP83820

DP83820 10/100/1000 Mb/s PCI Ethernet Network Interface Controller



Literature Number: SNLS070B

DP83820 10/100/1000 Mb/s PCI Ethernet Network Interface Controller

General Description

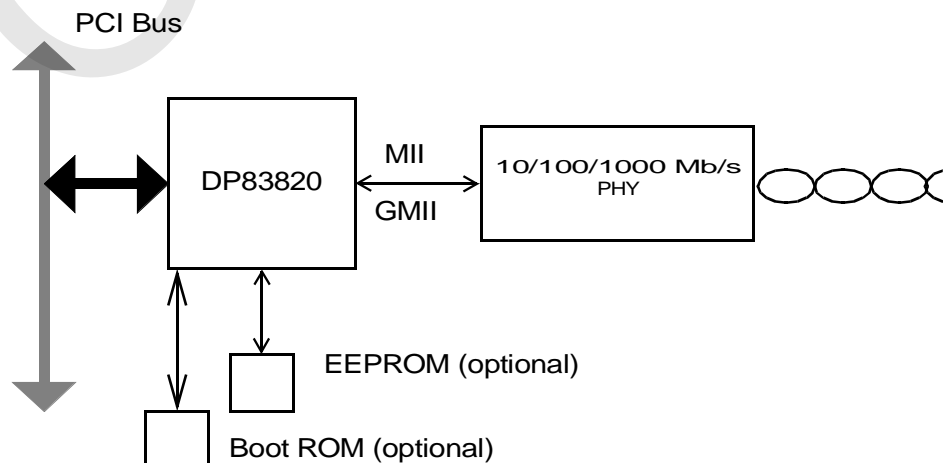
DP83820 is a single-chip 10/100/1000 Mb/s Ethernet Controller for the PCI bus. It is targeted at high-performance adapter cards and mother boards. The DP83820 fully implements the V2.2 66 MHz, 64-bit PCI bus interface for host communications with power management support. Packet descriptors and data are transferred via bus-mastering, reducing the burden on the host CPU. The DP83820 can support full duplex 10/100/1000 Mb/s transmission and reception.

Features

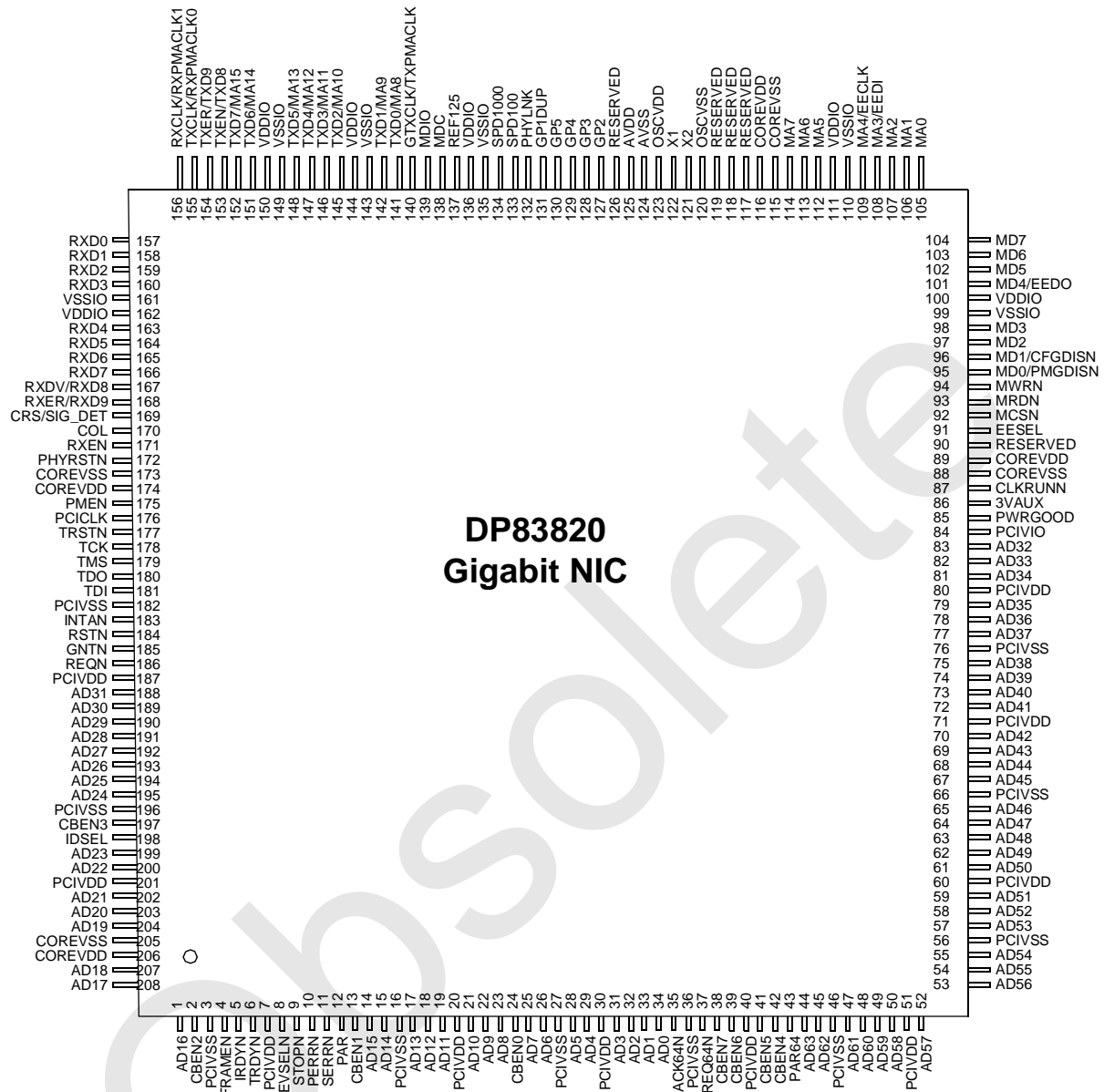
- IEEE 802.3 Compliant, 66/33 Mhz, 64/32-bit PCI V2.2 MAC/BIU supports data rates from 1 Mb/s to 1000 Mb/s. This allows support for traditional 10 Mb/s Ethernet, 100 Mb/s Fast Ethernet, as well as 1000 Mb/s Gigabit Ethernet.
- Flexible, programmable Bus master - burst sizes of up to 256 dwords (1024 bytes)
- BIU compliant with PC 97 and PC 98 Hardware Design Guides, PC 99 Hardware Design Guide draft, ACPI v1.0, PCI Power Management Specification v1, OnNow Device Class Power Management Reference Specification - Network Device Class v1.0a
- Wake on LAN (WOL) support compliant with PC98, PC99, and OnNow, including directed packets, Magic Packet with SecureOn, ARP packets, pattern match packets, and Phy status change
- GMII/MII provides IEEE 802.3 standard interface to support 10/100/1000 Mb/s physical layer devices
- Ten-Bit Interface (TBI) for support of 1000BASE-X

- Virtual LAN (VLAN) and long frame support. VLAN tag insertion support for transmit packets. VLAN tag detection and removal for receive packets
- 802.3x Full duplex flow control, including automatic transmission of Pause frames based on Rx FIFO thresholds
- IPv4 checksum task off-loading. Supports checksum generation and verification of IP, TCP, and UDP headers
- 802.1D and 802.1Q priority queueing support. Supports multiple priority queues in both transmit and receive directions.
- Extremely flexible Rx packet filtration including: single address perfect filter with MSb masking, broadcast, 2,048 entry multicast/unicast hash table, deep packet pattern matching for up to 4 unique patterns.
- Statistics gathered for support of RFC 1213 (MIB II), RFC 1398 (Ether-like MIB), IEEE 802.3 LME, reducing CPU overhead for management.
- Internal 8 KB Transmit and 32 KB Receive data FIFOs
- Supports Jumbo packets
- Serial EEPROM port with auto-load of configuration data from EEPROM at power-on
- Flash/PROM interface for remote boot support
- Full Duplex support for 10/100/1000 Mb/s data rates
- 208-pin PQFP package
- Low power CMOS design
- 3.3V powered I/Os with 5V tolerant inputs
- JTAG Boundary Scan supported

System Diagram



1.0 Connection Diagram



**Order Number DP83820VUW
See NS Package Number NVUW208A**

2.0 Pin Descriptions

PCI Interface

| Symbol | Pin No(s) | Direction | Description |
|---------|--|-----------|---|
| AD31-0 | 188, 189, 190, 191, 192, 193, 194, 195, 199, 200, 202, 203, 204, 207, 208, 1, 14, 15, 17, 18, 19, 21, 22, 23, 25, 26, 28, 29, 31, 32, 33, 34 | I/O | Address and Data: Multiplexed address and data bus. As a bus master, the DP83820 will drive address during the first bus phase. During subsequent phases, the DP83820 will either read or write data expecting the target to increment its address pointer. As a bus target, the DP83820 will decode each address on the bus and respond if it is the target being addressed. |
| CBEN3-0 | 197, 2, 13, 24 | I/O | Bus Command/Byte Enable: During the address phase these signals define the "bus command" or the type of bus transaction that will take place. During the data phase these pins indicate which byte lanes contain valid data. CBEN0 applies to byte 0 (bits 7-0) and CBEN3 applies to byte 3(bits 31-24). |
| PCICLK | 176 | I | Clock: This PCI Bus clock provides timing for all bus phases. The rising edge defines the start of each phase. The clock frequency ranges from 0 to 66 MHz. |
| DEVSELN | 8 | I/O | Device Select: As a target, the DP83820 asserts this signal low when it recognizes its address after FRAMEN is asserted. As a bus master, the DP83820 samples this signal to insure that the destination address for the data transfer is recognized by a PCI target. |
| FRAMEN | 4 | I/O | Frame: As a bus master, this signal is asserted low to indicate the beginning and duration of a bus transaction. Data transfer takes place when this signal is asserted. It is de-asserted before the transaction is in its final phase. As a target, the device monitors this signal before decoding the address to check if the current transaction is addressed to it. |
| GNTN | 185 | I | Grant: This signal is asserted low to indicate to the DP83820 that it has been granted ownership of the bus by the central arbiter. This input is used when the DP83820 is acting as a bus master. |
| IDSEL | 198 | I | Initialization Device Select: This pin is sampled by the DP83820 to identify when configuration read and write accesses are intended for it. |
| INTAN | 183 | O | Interrupt A: This signal is asserted low when an interrupt condition as defined in the Interrupt Status Register, Interrupt Mask, and Interrupt Enable registers occurs. |
| IRDYN | 5 | I/O | Initiator Ready: As a bus master, this signal will be asserted low when the DP83820 is ready to complete the current data phase transaction. This signal is used in conjunction with the TRDYN signal. Data transaction takes place at the rising edge of PCICLK when both IRDYN and TRDYN are asserted low. As a target, this signal indicates that the master has put the data on the bus. |
| PAR | 12 | I/O | Parity: This signal indicates even parity across AD31-0 and CBEN3-0 including the PAR pin. As a master, PAR is asserted during address and write data phases. As a target, PAR is asserted during read data phases. |
| PERRN | 10 | I/O | Parity Error: The DP83820 as a master or target will assert this signal low to indicate a parity error on any incoming data (except for special cycles). As a bus master, it will monitor this signal on all write operations (except for special cycles). |
| REQN | 186 | O | Request: The DP83820 will assert this signal low to request the ownership of the bus to the central arbiter. |
| RSTN | 184 | I | Reset: When this signal is asserted all outputs of DP83820 will be tri-stated and the device will be put into a known state. |

2.0 Pin Descriptions (Continued)

PCI Interface

| Symbol | Pin No(s) | Direction | Description |
|---------|--|-----------|--|
| SERRN | 11 | I/O | System Error: This signal is asserted low by DP83820 during address parity errors and system errors if enabled. |
| STOPN | 9 | I/O | Stop: This signal is asserted low by the target device to request the master device to stop the current transaction. |
| TRDYN | 6 | I/O | Target Ready: As a target, this signal will be asserted low when the (slave) device is ready to complete the current data phase transaction. This signal is used in conjunction with the IRDYN signal. Data transaction takes place at the rising edge of PCICLK when both IRDYN and TRDYN are asserted low. As a master, this signal indicates that the target is ready for the data during write operation and with the data during read operation. |
| PMEN | 175 | O | Power Management Event: This signal is asserted low by DP83820 to indicate that a power management event has occurred. |
| 3VAUX | 86 | I | PCI Aux Voltage Sense: This pin is used to sense the presence of a 3.3v auxiliary supply in order to define the PME Support available. This pin pad has an internal weak pull down. |
| PWRGOOD | 85 | I | PCI bus power good: Connected to PCI bus 3.3v power, this pin is used to sense the presence of PCI bus power during the D3 power management state. This pin pad has an internal weak pull down. |
| CLKRUNN | 87 | I/O | Clockrun: This signal is asserted low by DP83820 to indicate that a Clockrun Event has occurred. |
| AD63-32 | 44, 45, 47, 48, 49, 50, 52, 53, 54, 55, 57, 58, 59, 61, 62, 63, 64, 65, 67, 68, 69, 70, 72, 73, 74, 75, 77, 78, 79, 81, 82, 83 | I/O | 64-bit Extension Address and Data: Multiplexed address and data bus. Provides upper address bits during 64-bit DAC command. During data phase, used for transferring upper 32-bits of a 64-bit data transaction. |
| CBEN7-4 | 38, 39, 41, 42 | I/O | 64-bit Extension Bus Command/Byte Enables: During the address phase these signals define the "bus command" for a 64-bit DAC command. During a 64-bit data phase these pins indicate which byte lanes contain valid data. CBEN4 applies to byte 4(bits 39-32) and CBEN7 applies to byte 7(bits 63-56). |
| REQ64N | 37 | I/O | Request 64-bit Transfer: The DP83820 will assert this signal low to request a 64-bit transfer of data. This pin is sampled by the DP83820 during reset to determine if the device is connected to a 64-bit datapath. |
| ACK64N | 35 | I | Acknowledge 64-bit Transfer: The DP83820 will samples this signal on bus master cycles when it has requested a 64-bit data transfer. If both REQ64N and ACK64N are asserted, then a 64-bit transfer will be performed. As a target, the DP83820 only supports 32-bit transfers, so it will never assert ACK64N. |
| PAR64 | 43 | I/O | Parity Upper DWORD: This signal indicates even parity across AD63-32 and CBEN7-4 including the PAR64 pin. As a master, PAR64 is driven during address and write data phases. As a target, the DP83820 only supports 32-bit transfers, so it will not drive PAR64. |
| PCIVIO | 84 | I | PCI Bus VIO: This pin should be connected to the VIO pins of the PCI bus. It provides a direct connection to the ESDPLUS ring for biasing. It may be connected to 5V if available. It should not be connected to 3.3V unless all signaling is 3.3V as this will interfere with 5V tolerance. Care should be taken in connecting this to power supplies when power management functions are enabled. |

2.0 Pin Descriptions (Continued)

Media Independent Interface (MII) - and Gigabit Media Independent Interface (GMII).

| Symbol | Pin No(s) | Direction | Description |
|---|---|-----------|--|
| COL | 170 | I | Collision Detect: The COL signal is asserted high asynchronously by the external PMD upon detection of a collision on the medium. It will remain asserted as long as the collision condition persists. |
| CRS/SIGDET | 169 | I | Carrier Sense: This signal is asserted high asynchronously by the external physical unit upon detection of a non-idle medium. Signal Detect: In TBI mode, this signal is used to bring in the Signal Detect indication from the Phy. |
| MDC | 138 | O | Management Data Clock: Clock signal with a maximum rate of 2.5 MHz used to transfer management data for the external PMD on the MDIO pin. |
| MDIO | 139 | I/O | Management Data I/O: Bidirectional signal used to transfer management information for the external PMD. Requires an external 4.7 K Ω pullup resistor. |
| RXCLK/ RXPMACLK1 | 156 | I | Receive Clock: A continuous clock, sourced by an external PMD device, that is recovered from the incoming data. During 1000 Mb/s mode RX_CLK is 125 MHz, during 100 Mb/s operation RX_CLK is 25 MHz and during 10 Mb/s this is 2.5 MHz. Receive PMA Clock 1: In TBI mode, this 62.5Mhz clock is used in conjunction with RXPMACLK0 to clock 10-bit TBI data into the DP83820. The rising edge of RXPMACLK1 clocks the even-numbered bytes. |
| RXD7, RXD6, RXD5, RXD4, RXD3, RXD2, RXD1, RXD0 | 166, 165, 164, 163, 160, 159, 158, 157 | I | Gigabit Receive Data: This is a group of 8 signals, sourced from an external PMD, that contains data aligned on byte boundaries and are driven synchronous to the RX_CLK. RXD7 is most significant bit. Receive Data: This is a group of 4 signals, sourced from an external PMD, that contains data aligned on nibble boundaries and are driven synchronous to the RX_CLK. RXD3 is the most significant bit and RXD0 is the least significant bit. RXD7 through RXD4 are not used in this mode. TBI Receive Data: In TBI mode, these bits are the lower 8 bits of the 10-bit TBI Receive data. |
| RXDV/RXD8 | 167 | I | Receive Data Valid: This indicates that the external PMD is presenting recovered and decoded nibbles on the RXD signals, and that RX_CLK is synchronous to the recovered data in 100 Mb/s operation. This signal will encompass the frame, starting with the Start-of-Frame delimiter (JK) and excluding any End-of-Frame delimiter (TR). TBI Receive Data: In TBI mode, this is RXD8 of the 10-bit TBI Receive data. |
| RXER/RXD9 | 168 | I | Receive Error: This signal is asserted high synchronously by the external PMD whenever it detects a media error and RXDV is asserted in 100 Mb/s or 1000 Mb/s operation. TBI Receive Data: In TBI mode, this is RXD9 of the 10-bit TBI Receive data. |
| RXEN | 171 | O | Receive Output Enable: This pin is used to disable an external PMD while the BIOS ROM is being accessed. |
| TXCLK/ RXPMACLK0 | 155 | I | MII Transmit Clock: A continuous clock that is sourced by the external PMD. During 100 Mb/s operation this is 25 MHz +/- 100 ppm. During 10 Mb/s operation this clock is 2.5 MHz +/- 100 ppm. Receive PMA Clock 0: In TBI mode, this 62.5Mhz clock is used in conjunction with RXPMACLK1 to clock 10-bit TBI data into the DP83820. The rising edge of RXPMACLK0 clocks the odd-numbered bytes. |

2.0 Pin Descriptions (Continued)

Media Independent Interface (MII) - and Gigabit Media Independent Interface (GMII).

| Symbol | Pin No(s) | Direction | Description |
|---|---|-----------|---|
| TXD7/MA15, TXD6/MA14, TXD5/MA13, TXD4/MA12, TXD3/MA11, TXD2/MA10, TXD1/MA9, TXD0/MA8 | 152, 151, 148, 147, 146, 145, 142, 141 | O | <p>Gigabit Transmit Data: This is a group of 8 signals which are driven synchronous to GTXCLK. TXD7 is the most significant bit.</p> <p>Transmit Data: This is a group of 4 data signals which are driven synchronous to the TXCLK for transmission to the external PMD. TXD3 is the most significant bit and TXD0 is the least significant bit. TXD7 through TXD4 are not used in this mode</p> <p>TBI Transmit Data: In TBI mode, this is the lower 8 bits of the 10-bit TBI Transmit data.</p> <p>BIOS ROM Address: During external BIOS ROM access, these signals become part of the ROM address.</p> |
| TXEN/TXD8 | 153 | O | <p>Transmit Enable: This signal is synchronous to TXCLK and provides precise framing for data carried on TXD3-0 for the external PMD. It is asserted when TXD3-0 contains valid data to be transmitted.</p> <p>TBI Transmit Data: In TBI mode, this is TXD8 of the 10-bit TBI Transmit data.</p> |
| TXER/TXD9 | 154 | O | <p>Transmit Error: This signal is synchronous to TXCLK and provides error indications and also is used for 1000 Mb/s half-duplex carrier extension and packet bursting functions. The DP83820 will only assert this signal in 1000 Mb/s mode of operation.</p> <p>TBI Transmit Data: In TBI mode, this is TXD9 of the 10-bit TBI Transmit data.</p> |
| GTCLK/ TXPMACK | 140 | O | <p>GMII transmit Clock: A continuous clock used for 1000 Mb/s. It is output to an external PMD and is the reference clock for Transmit GMII signaling. The clock frequency is 125 MHz.</p> <p>TBI Transmit Clock: In TBI mode, this is the 125MHz transmit clock to an external PMD and is the reference for Transmit TBI signaling.</p> |
| REF125 | 137 | I | <p>125 MHz Reference Clock: May be optionally connected to a 125 MHz oscillator for 1000 Mb/s mode. If not used should be tied high.</p> |

BIOS ROM/Flash Interface

| Symbol | Pin No(s) | Direction | Description |
|--|---|-----------|--|
| MCSN | 92 | O | <p>BIOS PROM/Flash Chip Select: During a BIOS ROM/Flash access, this signal is used to select the ROM device.</p> |
| MD7, MD6, MD5, MD4/EEDO, MD3, MD2, MD1/CFGDISN, MD0/PMGDISN | 104, 103, 102, 101, 98, 97, 96, 95, | I/O | <p>BIOS ROM/Flash Data Bus: During a BIOS ROM/Flash access these signals are used to transfer data to or from the ROM/Flash device.</p> <p>MD5:0 and MD7 pin pads have an internal weak pull up.</p> <p>MD6 pin pad has an internal weak pull down.</p> |
| MA15/TXD7, MA14/TXD6, MA13/TXD5, MA12/TXD4, MA11/TXD3, MA10/TXD2, MA9/TXD1, MA8/TXD0, MA7, MA6, MA5, MA4/EECLK, MA3/EEDI, MA2, MA1, MA0 | 152, 151, 148, 147, 146, 145, 142, 141, 114, 113, 112, 109, 108, 107, 106, 105 | O | <p>BIOS ROM/Flash Address: During a BIOS ROM/Flash access, these signals are used to drive the ROM/Flash address.</p> |

2.0 Pin Descriptions (Continued)

BIOS ROM/Flash Interface

| Symbol | Pin No(s) | Direction | Description |
|--------|-----------|-----------|--|
| MWRN | 94 | O | BIOS ROM/Flash Write: During a BIOS ROM/Flash access, this signal is used to enable data to be written to the Flash device. |
| MRDN | 93 | O | BIOS ROM/Flash Read: During a BIOS ROM/Flash access, this signal is used to enable data to be read from the Flash device. |

Note: DP83820 supports NM27LV010 for the ROM interface device.

Clock Interface

| Symbol | Pin No(s) | Direction | Description |
|--------|-----------|-----------|---|
| X1 | 122 | I | Crystal/Oscillator Input: This pin is the primary clock reference input for the DP83820 IC and must be connected to a 25MHz 0.005% (50ppm) clock source. The DP83820 device supports either an external crystal resonator connected across pins X1 and X2, or an external CMOS-level oscillator source connected to pin X1 only. |
| X2 | 121 | O | Crystal Output: This pin is used in conjunction with the X1 pin to connect to an external 25MHz crystal resonator device. This pin must be left unconnected if an external CMOS oscillator clock source is utilized. For more information see the definition for pin X1. |

Phy And General Purpose Interface

| Symbol | Pin No(s) | Direction | Description |
|---------|-----------|-----------|---|
| GP1DUP | 131 | I/O | General Purpose Pin 1 or Duplex Status: By default, this pin can be used to input the Full Duplex status from an external Phy. The pin can also be programmed as a general purpose I/O. This pin pad has an internal weak pull up. |
| GP2 | 127 | I/O | General Purpose Pin 2: This pin is a general purpose I/O pin that can be programmed as an input or output. This pin has an internal weak pull up. |
| GP3 | 128 | I/O | General Purpose Pin 3: This pin is a general purpose I/O pin that can be programmed as an input or output. This pin has an internal weak pull up. |
| GP4 | 129 | I/O | General Purpose Pin 4: This pin is a general purpose I/O pin that can be programmed as an input or output. This pin has an internal weak pull up. |
| GP5 | 130 | I/O | General Purpose Pin 5: This pin is a general purpose I/O pin that can be programmed as an input or output. This pin has an internal weak pull up. |
| PHYLNK | 132 | I | Phy Link Status: This can be used to input the Phy Link Status. This allows the value to be read back from the MAC register space. |
| SPD100 | 133 | I | 100 Mb/s Speed Status: This can be used to input the 100 Mb/s Speed Status from an external phy. This is used along with the SPD1000 bit to determine current speed status of the Phy. |
| SPD1000 | 134 | I | 1000 Mb/s Speed Status: This can be used to input the 1000 Mb/s Speed Status from an external phy. This is used along with the SPD100 bit to determine current speed status of the Phy. |
| PHYRSTN | 172 | O | Phy Reset: This pin can be used to reset an External Phy. |

2.0 Pin Descriptions (Continued)

Serial EEPROM Interface

| Symbol | Pin No(s) | Direction | Description |
|-----------|-----------|-----------|--|
| EESEL | 91 | O | EEPROM Chip Select: This signal is used to enable the external EEPROM device. |
| MA4/EECLK | 109 | O | EEPROM Clock: During an EEPROM access (EESEL asserted), this pin is an output used to drive the serial clock to an external EEPROM device. |
| MA3/EEDI | 108 | O | EEPROM Data In: During an EEPROM access (EESEL asserted), this output is drives opcode, address, and data to an external serial EEPROM device. |
| MD4/EEDO | 101 | I | EEPROM Data Out: During an EEPROM access (EESEL asserted), this pin is an input used to retrieve EEPROM serial read data. This pin pad has an internal weak pull up. |

Note: DP83820 supports NMC93C46 for the eeprom interface device.

JTAG Interface

| Symbol | Pin No(s) | Direction | Description |
|--------|-----------|-----------|------------------|
| TCK | 178 | I | Test Clock |
| TDI | 181 | I | Test Data Input |
| TDO | 180 | O | Test Output |
| TMS | 179 | I | Test Mode Select |
| TRSTN | 177 | I | Test Reset |

Supply Pins

| Symbol | Pin No(s) | Direction | Description |
|-------------|---|-----------|---|
| COREVDD | 89, 116, 174, 206 | S | Mac/BIU digital core VDD - connect to Aux 1.8V supply VDD |
| COREVSS | 88, 115, 173, 205 | S | Mac/BIU digital core VSS. |
| OSCVDD | 123 | S | Oscillator VDD - connect to Aux 1.8V supply VDD |
| OSCVSS | 120 | S | Oscillator VSS. |
| PCIVDD | 187, 201, 7, 20, 30, 40, 51, 60, 71, 80 | S | PCI IO VDD - connect to PCI bus 3.3V VDD |
| PCIVSS | 182, 196, 3, 16, 27, 36, 46, 56, 66, 76 | S | PCI IO VSS |
| VDDIO, AVDD | 100, 111, 136, 144, 150, 162, 125 | S | Misc. IO VDD, Analog VDD - connect to Aux 3.3V supply VDD |
| VSSIO, AVSS | 99, 110, 135, 143, 149, 161, 124 | S | Misc. IO VSS, Analog VSS |

No Connects

| Symbol | Pin No(s) | Direction | Description |
|----------|------------------------|-----------|--|
| Reserved | 90, 117, 118, 119, 126 | | This pin is reserved and cannot be connected to any external logic or net. |

3.0 Functional Description

DP83820 consists of a PCI bus interface, BIOS ROM and EEPROM interfaces, Receive and Transmit Data Buffer Managers, an 802.3 Media Access Controller (MAC), SRAM, and miscellaneous support logic.

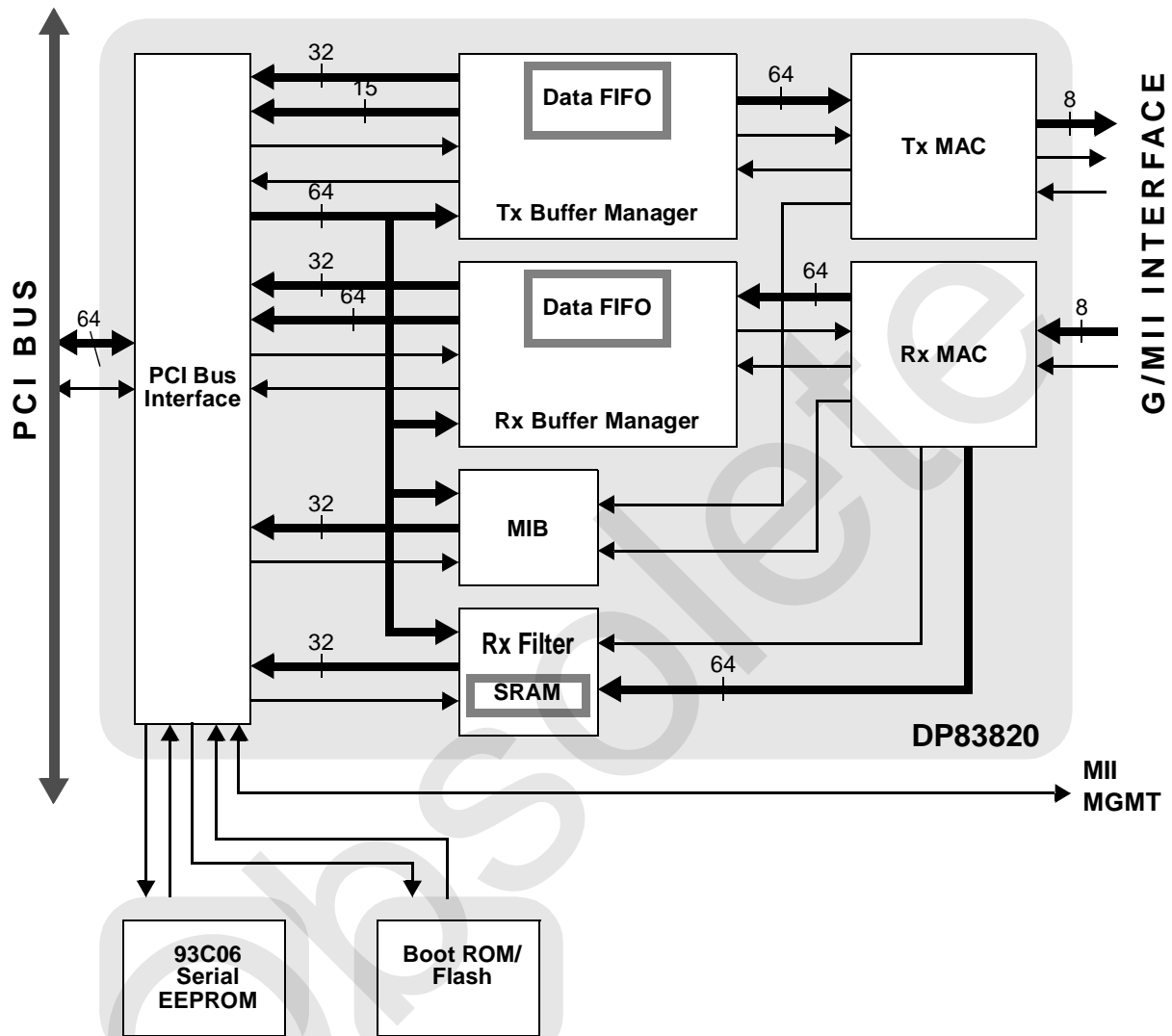


Figure 3-1 DP83820 Functional Block Diagram

3.1 DP83820

The DP83820 device is an enhanced version of the NSC MacPhyter MAC/BIU (Media Access Controller/Bus Interface Unit) which has been modified for 1000 Mb/s operation with additional buffering, higher bandwidth PCI bus implementation, and the Gigabit Media Independent Interface for 1000BASE-T phy support. The DP83820 supports an external 10/100/1000 physical layer device.

DP83820 contains the following major design elements:

- a PCI bus interface,
- an EEPROM interface, for access to an NMC93C06 EEPROM,

- a buffer management scheme that is simple, efficient and flexible,
- separate receive and transmit FIFOs and DMA controllers,
- a 10/100/1000 Mb/s Ethernet Media Access Control (MAC),
- a Physical Layer Interface (MII/GMII/TBI),
- Management Information Base (MIB) Statistics Registers,
- Receive Packet filtering logic.

This following section provides a functional overview of interfaces of the DP83820.

3.0 Functional Description (Continued)

3.2 PCI Bus Interface

The DP83820 implements the Peripheral Component Interconnect (PCI) bus interface as defined in PCI Local Bus Specification Version 2.2. When internal registers are being accessed the DP83820 acts as a PCI target (slave). When accessing host memory for descriptor or packet data transfer, the DP83820 acts as a PCI bus master.

All required pins and functions are implemented. The optional interface pin INTA for support of interrupt requests is implemented as well. The bus interface also supports 64-bit and 66Mhz operation in addition to the more common 32-bit and 33-Mhz capabilities.

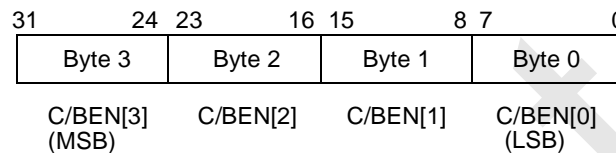
For more information, refer to the PCI Local Bus Specification version 2.2, December 18, 1998.

3.2.1 Byte Ordering

The DP83820 can be configured to order the bytes of data on the AD[31:0] bus to conform to Little Endian or Big Endian ordering through the use of the CFG:BEM bit. Byte ordering only affects bus mastered packet data transfers in 32-bit mode. Register information remains bit aligned (i.e. AD[31] maps to bit 31 in any register space, AD[0] maps to bit 0, etc.) when registers are accessed with 32-bit operations. Bus mastered transfers of buffer descriptor information also remain bit aligned.

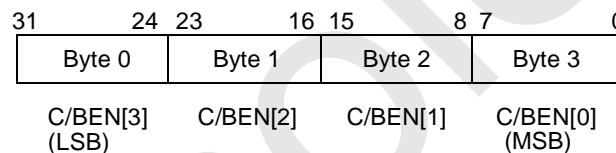
When configured for Little Endian (CFG:BEM=0), the byte orientation for receive and transmit data and descriptors in system memory is as follows:

Figure 3-2 Little Endian Byte Ordering



When configured for *big-endian* mode (CFG:BEM=1), the byte orientation for receive and transmit data and descriptors in system memory is as follows:

Figure 3-3 Big Endian Byte Ordering



3.2.2 Interrupt Control

Interrupts are performed by asynchronously asserting the INTAN pin. This pin is an open drain output. The source of the interrupt can be determined by reading the Interrupt Status Register (ISR) (See Section 4.2.6). One or more bits in the ISR will be set, denoting all currently pending interrupts. Reading of the ISR clears ALL bits. Masking of specific interrupts can be accomplished by using the Interrupt Mask Register (IMR) (See Section 4.2.7). Assertion of INTAN can be prevented by clearing the Interrupt Enable bit in the Interrupt Enable Register (See Section 4.2.8). This allows the system to defer interrupt processing as needed.

3.2.3 Latency Timer

The *Latency Timer* described in CFGLAT:LAT (See Section 4.1.4) defines the maximum number of bus clocks that the device will hold the bus. Once the device gains control of the bus and issues FRAMEN, the Latency Timer will begin counting down. If GNTN is deasserted before the DP83820 has finished with the bus, the device will maintain ownership of the bus until the timer reaches zero (or has finished the bus transfer). The timer is an 8-bit counter, with the lower 4 bits hard-coded to 1111b. This means that the timer value can only be incremented in units of 16 clocks.

3.2.4 64-Bit Data Operation

The DP83820 supports 64-bit operation as a bus master for transferring descriptor and packet data information. This mode can be enabled or disabled through configuration from EEPROM. As a target, the DP83820 only supports

32-bit mode of operation. At the rising edge of RSTN, the DP83820 samples the REQ64N pin to determine if the bus is 64-bit capable. If the bus is not 64-bit capable, the DP83820 will drive the 64-bit extension signals AD[63:32], CBEN[7:4], and PAR64 to a low level to prevent the floating inputs from causing significant current drain.

3.2.5 64-Bit Addressing

The DP83820 supports 64-bit addressing (Dual Address Cycle) as a bus master for transferring descriptor and packet data information. This mode can be enabled or disabled through configuration from EEPROM. The DP83820 also supports 64-bit addressing as a target.

3.3 Bus Operation

3.3.1 Target Read

A Target Read operation starts with the system generating FRAMEN, Address, and either an IO read (0010b) or Memory Read (0110b) command. See Figure 3-4. If the 32-bit address on the address bus matches the IO address range specified in CFGIOA:IOBASE (for I/O reads) or the memory address range specified in CFGMA:MEMBASE (for memory reads), the DP83820 will generate DEVSELN 2 clock cycles later (medium speed).

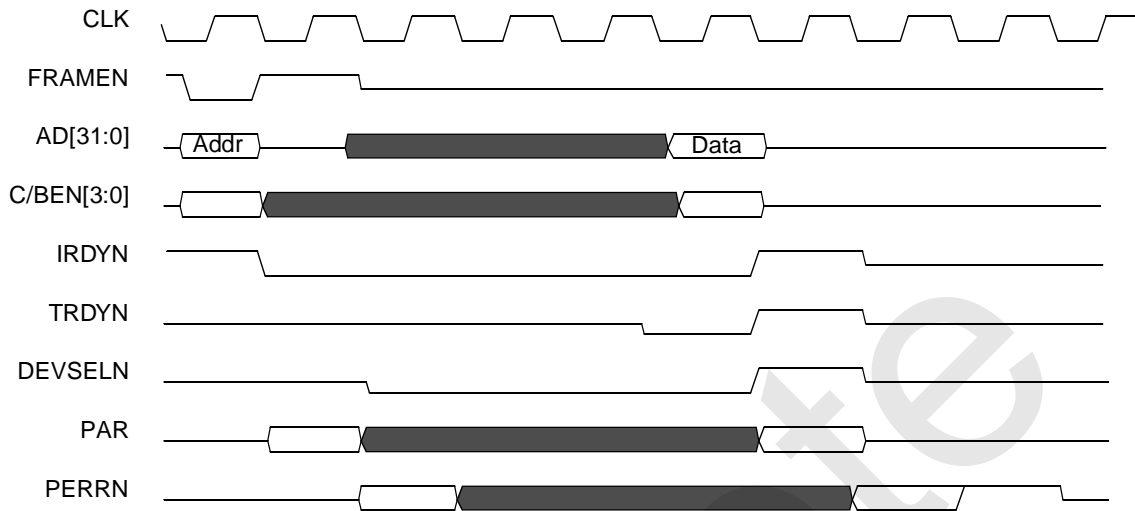
The system must tri-state the Address bus, and convert the C/BEN bus to byte enables, after the address cycle. On the 2nd cycle after the assertion of DEVSELN, all 32-bits of data and TRDYN will become valid. If IRDYN is asserted at that time, TRDYN will be forced HIGH on the next clock for 1 cycle, and then tri-stated.

3.0 Functional Description (Continued)

If FRAMEN is asserted beyond the assertion of IRDYN, the DP83820 will still make data available as described above, but will also issue a Disconnect. That is, it will assert the

STOPN signal with TRDYN. STOPN will remain asserted until FRAMEN is detected as deasserted.

Figure 3-4 Target Read Operation



3.3.2 Target Write

A Target Write operation starts with the system generating FRAMEN, Address, and Command (0011b or 0111b). See Figure 3-5. If the upper 24 bits on the address bus match CFGIOA:IOBASE (for I/O reads) or CFGMA:MEMBASE (for memory reads), the DP83820 will generate DEVSELN 2 clock cycles later.

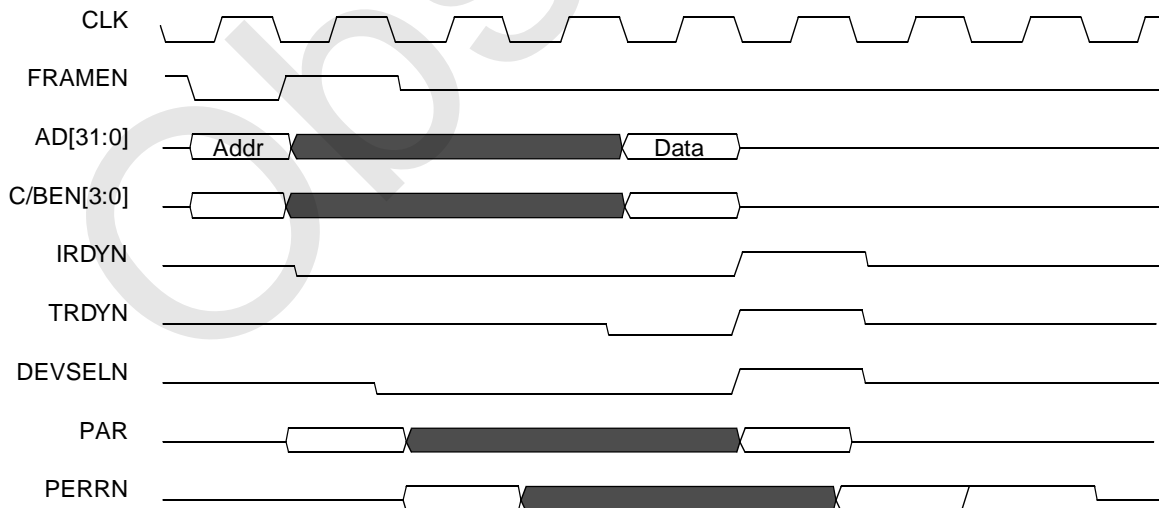
On the 2nd cycle after the assertion of DEVSELN, the device will monitor the IRDYN signal. If IRDYN is asserted

at that time, the DP83810 will assert TRDYN. On the next clock the 32-bit double word will be latched in, and TRDYN will be forced HIGH for 1 cycle and then tri-stated.

Note: Target write operations must be 32-bits wide.

If FRAMEN is asserted beyond the assertion of IRDYN, the DP83820 will still latch the first double word as described above, but will also issue a Disconnect. That is, it will assert the STOPN signal with TRDYN. STOPN will remain asserted until FRAMEN is detected as deasserted.

Figure 3-5 Target Write Operation



3.0 Functional Description (Continued)

3.3.3 Master Read

A Master Read operation starts with the DP83820 asserting REQN. See Figure 3-6. If GNTN is asserted within 2 clock cycles, FRAMEN, Address, and Command will be generated 2 clocks after REQN (Address and FRAMEN for 1 cycle only). If GNTN is asserted 3 cycles or later, FRAMEN, Address, and Command will be generated on the clock following GNTN.

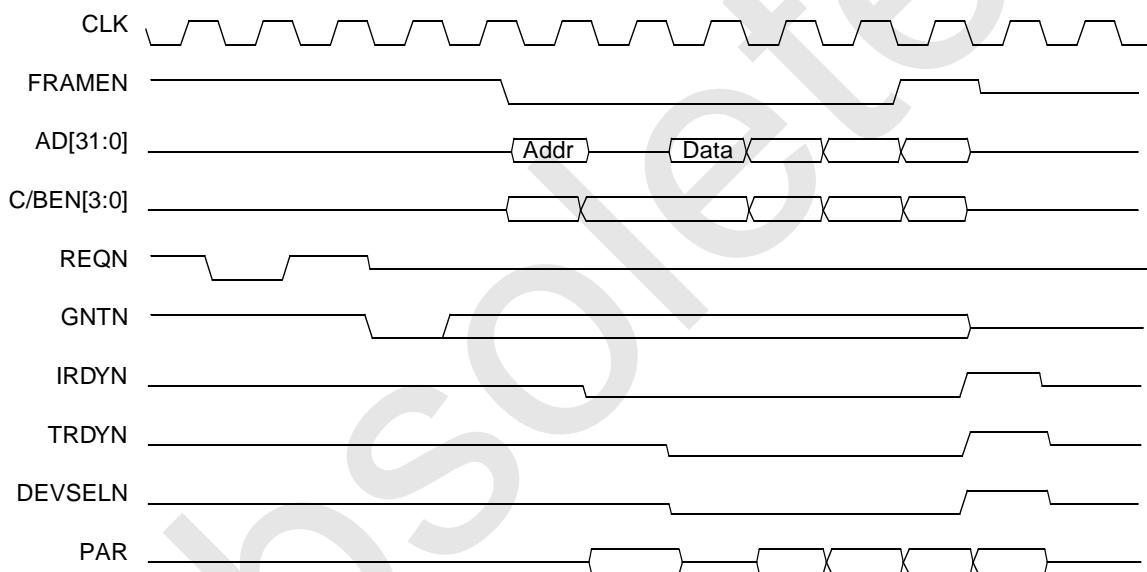
The device will wait for 8 cycles for the assertion of DEVSELN. After 8 clocks without DEVSELN, the device will issue a Master Abort by asserting FRAMEN HIGH for 1 cycle. IRDYN will be forced HIGH on the following cycle. Both signals will become tri-state on the cycle following their deassertion.

On the clock edge after the generation of Address and Command, the address bus will become tri-state, and the

C/BEN bus will contain valid byte enables. On the clock edge after FRAMEN was asserted, IRDYN will be asserted (and FRAMEN will be deasserted if this is to be a single read operation). On the clock where both TRDYN and DEVSELN are detected as asserted, data will be latched in (and the byte enables will change if necessary). This will continue until the cycle following the deassertion of FRAMEN.

On the clock where the second to last read cycle occurs, FRAMEN will be forced HIGH (it will be tri-stated 1 cycle later). On the next clock edge that the device detects TRDYN asserted, it will force IRDYN HIGH. It, too, will be tri-stated 1 cycle later. This will conclude the read operation. The DP83820 will never force a wait state during a read operation.

Figure 3-6 Master Read Operation



3.3.4 Master Write

A Master Write operation starts with the DP83820 asserting REQN. See Figure 3-7. If GNTN is asserted within 2 clock cycles, FRAMEN, Address, and Command will be generated 2 clocks after REQN (Address and FRAMEN for 1 cycle only). If GNTN is asserted 3 cycles or later, FRAMEN, Address, and Command will be generated on the clock following GNTN.

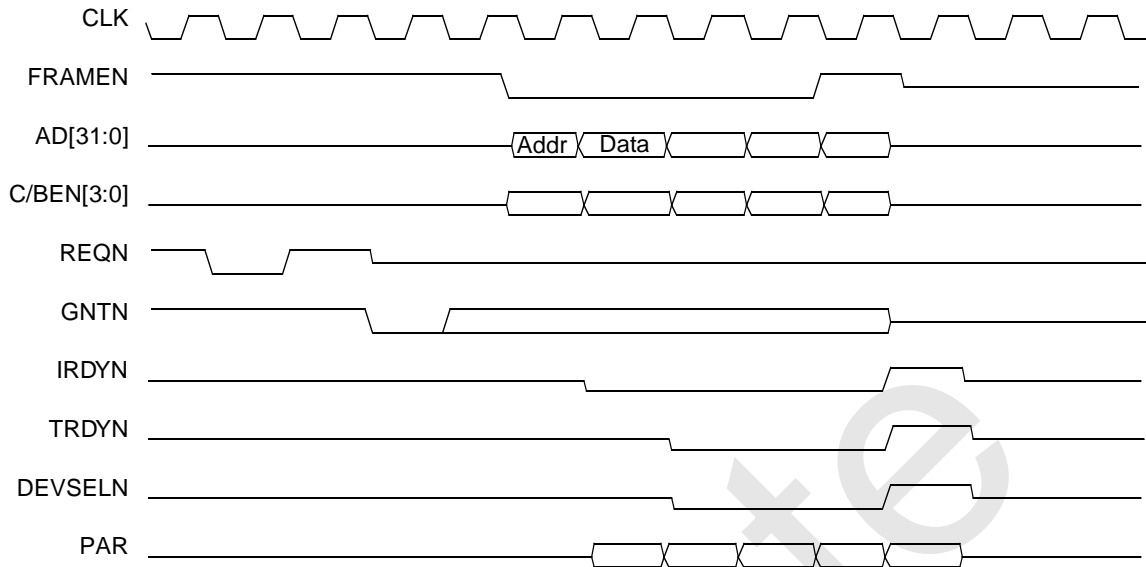
The device will wait for 8 cycles for the assertion of DEVSELN. After 8 clocks without DEVSELN, the device will issue a Master Abort by asserting FRAMEN HIGH for 1 cycle. IRDYN will be forced HIGH on the following cycle. Both signals will become tri-state on the cycle following their deassertion.

On the clock edge after the generation of Address and Command, the data bus will become valid, and the C/BEN bus will contain valid byte enables. On the clock edge after FRAMEN was asserted, IRDYN will be asserted (and FRAMEN will be deasserted if this is to be a single read operation). On the clock where both TRDYN and DEVSELN are detected as asserted, valid data for the next cycle will become available (and the byte enables will change if necessary). This will continue until the cycle following the deassertion of FRAMEN.

On the clock where the second to last write cycle occurs, FRAMEN will be forced HIGH (it will be tri-stated 1 cycle later). On the next clock edge that the device detects TRDYN asserted, it will force IRDYN HIGH. It, too, will be tri-stated 1 cycle later. This will conclude the write operation. The DP83820 will never force a wait state during a write operation.

3.0 Functional Description (Continued)

Figure 3-7 Master Write Operation



3.3.5 Configuration Access

Configuration register accesses are similar to Target reads and writes in that they are single data word transfers and are initiated by the system. For the system to initiate a Configuration access, it must also generate IDSELN as well as the correct Command (1010b or 1011b) during the Address phase. The DP83820 will respond as it does during Target operations.

Note: Configuration reads must be 32-bits wide, but writes may access individual bytes.

3.4 Packet Buffering

The DP83820 incorporates two independent FIFOs for transferring data to/from the system interface and from/to the network. The FIFOs, providing temporary storage of data, free the host system from the real-time demands of the network.

The way in which the FIFOs are emptied and filled is controlled by the FIFO threshold values in the TXCFG and RXCFG registers (See Sections 4.2.12 and 4.2.16). These values determine how full or empty the FIFOs must be before the device requests the bus. Additionally, there is a threshold value that determines how full the transmit FIFO must be before beginning transmission. Once the DP83820 requests the bus, it will attempt to empty or fill the FIFOs as allowed by the respective MXDMA settings in TXCFG and RXCFG.

3.4.1 Transmit Buffer Manager

The buffer management scheme used on the DP83820 allows quick, simple and efficient use of the frame buffer memory. The buffer management scheme uses separate buffers and descriptors for packet information. This allows effective transfers of data to the transmit buffer manager by simply transferring the descriptor information to the transmit queue. Refer to the Buffer Management section for complete information.

The Tx Buffer Manager DMA's packet data from PCI memory space and places it in the 8KB transmit FIFO, and pulls data from the FIFO to send to the Tx MAC. Multiple

packets may be present in the FIFO, allowing packets to be transmitted with minimum interframe gap. The way in which the FIFO is emptied and filled is controlled by the FIFO threshold values in the TXCFG register: FLTH (Tx Fill Threshold), and DRTH (Tx Drain Threshold). Additionally, once the DP83820 requests the bus, it will attempt to fill the FIFO as allowed by the MXDMA setting in the TXCFG register.

3.4.2 Transmit Priority Queueing

The Tx Buffer Manager process also supports priority queueing of transmit packets. It handles this by drawing from four separate descriptor lists to fill the internal FIFO. If packets are available in the higher priority queues, they will be loaded into the FIFO before those of lower priority.

3.4.3 Receive Buffer Manager

The Rx Buffer Manager uses the same buffer management scheme as used for transmits. Refer to the Buffer Management section for complete information.

The Rx Buffer Manager retrieves packet data from the Rx MAC and places it in the 32KB receive data FIFO, and pulls data from the FIFO for DMA to PCI memory space. The Rx Buffer Manager maintains a status FIFO, allowing up to 32 packets to reside in the FIFO at once. Similar to the transmit FIFO, the receive FIFO is controlled by the FIFO threshold value in RXCFG:DRTH (Rx Drain Threshold). This value determines the number of long words written into the FIFO from the MAC unit before a DMA request for system memory occurs. Once the DP83820 gets the bus, it will continue to transfer the long words from the FIFO until the data in the FIFO is less than one long word, or has reached the end of the packet, or the max DMA burst size is reached (RXCFG register:MXDMA).

3.4.4 Receive Priority Queueing

The Rx Buffer Manager process also supports priority queueing of receive packets. It handles this by placing packets on up to four separate descriptor lists when emptying the internal FIFO. The Rx Buffer Manager uses information in a VLAN tag to determine packet priority.

3.0 Functional Description (Continued)

3.4.5 Packet Recognition

The Receive packet filter and recognition logic allows software to control which packets are accepted based on destination address and packet type. Address recognition logic includes support for broadcast, multicast hash, and unicast addresses. The packet recognition logic includes support for WOL, Pause, and programmable pattern recognition.

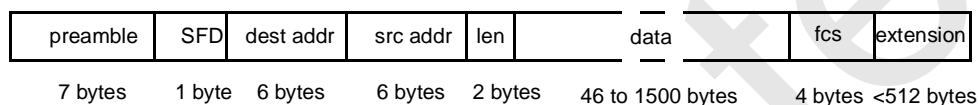
3.5 Ethernet Media Access Controller (MAC)

The Media Access Control (MAC) unit performs the control functions for the media access of transmitting and receiving packets. During transmission, the MAC unit handles building of frames and transmission of the frames over the interface to the physical layer device. During reception, data is received from the physical layer interface, the frame

is checked for valid reception, and the data is transferred to the receive FIFO. Control and status registers in the DP83820 govern the operation of the MAC unit.

The standard 802.3 Ethernet packet consists of the following fields: preamble, start of frame delimiter (SFD), destination address, source address, length, data, frame check sequence (FCS) and Extension (See Figure 3-8). All fields are of fixed length except for the data field and Extension. The Extension field is only used for 1000 Mb/s half-duplex operation. During reception, the preamble and SFD are stripped from the incoming packet. During transmission, the DP83820 generates and prepends the preamble and SFD. The FCS is normally appended by the DP83820, but software may disable FCS inclusion on a per-packet basis.

Figure 3-8 IEEE 802.3 Packet Structure



3.5.1 Full Duplex Operation

Full duplex operation is the simultaneous transmission and reception of packet data. In this mode of operation, receive activity (CRS) is ignored in the decision making process for transmission. During reception, collisions are also ignored.

To configure the DP83820 to operate in full duplex, set TXCFG:CSI and TXCFG:HBI=1, and RXCFG:RX_FD = 1.

3.5.2 Full Duplex Flow Control

The DP83820 supports full duplex flow control using the MAC Control Pause Frame as defined in the 802.3 specification. The packet recognition logic can detect Pause frames, and cause the transmit MAC to pause the correct number of slot times. In addition, the MAC can be programmed to send Pause frames based on Rx FIFO thresholds.

Flow Control operation is controlled by the Pause Control/Status Register.

3.5.3 1000 Mb/s Operation

The DP83820 includes additional features to support 1000 Mb/s speed of operation. In this mode, the physical layer interface is increased from 4-bit MII to 8-bit GMII (or 10-bit TBI). In addition, features such as carrier extension and frame bursting are required to meet the 802.3 specification for 1000 Mb/s half-duplex operation.

3.6 Transmit MAC

The Transmit MAC implements the transmit portion of 802.3 Media Access Control. The Tx MAC retrieves packet data from the Tx Buffer Manager and sends it out through the transmit physical layer interface. Additionally, the Tx MAC provides MIB control information for transmit packets. The TX MAC supports 4-bit MII, 8-bit GMII, and 10-bit TBI interfaces to physical layer devices

3.6.1 VLAN Tag Insertion

The Tx MAC has the capability to insert a 4-byte VLAN tag in the transmit packet. If Tx VLAN Tag insertion is enabled,

the MAC will insert the 4 bytes, as specified in the VTAG register, following the source and destination addresses of the packet. The VLAN tag insertion can be enabled on a global or per-packet basis.

3.6.2 Carrier Extension

For 1000 Mb/s half-duplex operation it is necessary for MAC to ensure that all valid carrier events exceed a slotTime of 4096 bit times. To accomplish this, any transmit event that is shorter than the slotTime will be extended using Carrier Extension. On the GMII interface, this is signaled to the Phy by TXER asserted with TXEN deasserted and a TXD value of 0x0F.

3.6.3 Frame Bursting

The Tx MAC supports burst mode operation for 1000 Mb/s half-duplex operation. This allows the device to transmit a burst of packets without releasing control of the physical medium. After a successful transmission, if additional packets are available, the MAC will transmit a burst of packets without allowing the medium to go idle. It does this by inserting carrier extension between the frames. The MAC will continue to burst frames as long as additional packets are available in the internal FIFO and a burstLimit of 65536 bit times has not been exceeded.

3.6.4 IP Checksum Generation

The Tx MAC supports task offloading of IP, TCP, and UDP checksum generation. It can generate the checksums and insert them into the packet. The checksum generation can be enabled on a global or per-packet basis.

3.7 Receive MAC

This block implements the receive portion of 802.3 Media Access Control. The Rx MAC retrieves packet data from the receive portion and sends it to the Rx Buffer Manager. Additionally, the Rx MAC provides MIB control information and packet address data for the Rx Filter. The RX MAC supports 4-bit MII, 8-bit GMII, and 10-bit TBI interfaces to physical layer devices.

3.0 Functional Description (Continued)

3.7.1 VLAN Tag Handling

The Rx MAC can detect packets containing a 4-byte VLAN tag, and remove the VLAN tag from the received packet. If RX VLAN Tag removal is enabled, then the 4 bytes following the source and destination addresses will be stripped out. The VLAN status can be returned in the Receive Descriptor Extended Status field.

3.7.2 Carrier Extension and Packet Bursting

The Receive MAC supports reception of packets with Carrier Extension and packets transmitted using Frame Bursting for 1000 Mb/s half-duplex operation. The first frame in a burst must be at least one slotTime in length, otherwise it will be considered to be a collision fragment.

3.7.3 IP Checksum Verification

The Rx MAC supports IP checksum verification. It can validate IP checksums as well as TCP and UDP checksums. Packets can be discarded based on detecting checksum errors.

3.8 Physical Layer Interface

The DP83820 implements a physical layer interface that can support all of the following:

- Media Independent Interface (MII)
- Gigabit Media Independent Interface (GMII)
- Ten-Bit Interface (TBI)

In addition, the DP83820 implements a Management interface as defined for MII and GMII.

3.8.1 Media Independent Interface (MII)

The DP83820 supports 10 Mb/s and 100 Mb/s physical layer devices through the Media Independent Interface (MII) as defined in IEEE 802.3 (clause 22). The MII consists of a transmit data interface (TXEN, TXER, TXD[3:0], and TXCLK), a receive data interface (RXDV, RXER, RXD[3:0], and RXCLK), 2 status signals (CRS and COL) and a management interface (MDC and MDIO). In

this mode of operation, both Transmit and Receive clocks are supplied by the Phy.

3.8.2 Gigabit Media Independent Interface (GMII)

The DP83820 can support 1000 Mb/s physical layer devices through the Gigabit Media Independent Interface (GMII) as defined in IEEE 802.3 (clause 35). The GMII is extended from the MII to use 8-bit data interfaces and to operate at higher frequency. The GMII consists of a transmit data interface (TXEN, TXER, TXD[7:0], and GTXCLK), a receive data interface (RXDV, RXER, RXD[7:0], and RXCLK), 2 status signals (CRS and COL) and a management interface (MDC and MDIO). Many of the signals are shared with the MII interface. One significant difference is the Transmit clock (GTXCLK) is supplied by the DP83820 instead of the Phy. The management interface (described later) is the same in both MII and GMII modes

3.8.3 Ten-Bit Interface (TBI)

The TBI provides a port for transmit and receive data for interfacing to devices that support the 1000Base-X portion of the 802.3 specification. This includes 1000Base-FX fiber devices. The port consists of data paths that are 10-bits wide in each direction as well as control signals. This interface shares pins with the MII and GMII interfaces.

3.8.4 MII/GMII Management Interface

The MII/GMII management interface utilizes a communication protocol similar to a serial EEPROM. Signaling occurs on two signals: clock (MDC) and data (MDIO). This protocol provides capability for addressing up to 32 individual Physical Media Dependent (PMD) devices which share the same serial interface, and for addressing up to 32 16-bit read/write registers within each PMD. The MII management protocol utilizes following frame format: start bits (SB), opcode (OP), PMD address (PA), register address (RA), line turnaround (LT) and data (See Figure 3-9).

Figure 3-9 MII Management Frame Format

| SB | OP | PA | RA | LT | Data |
|----|----|----|----|----|------|
| 2b | 2b | 5b | 5b | 2b | 16b |

Note: b = bits

- Start bits are defined as <01>.
- Opcode bits are defined as <01> for a Write access and <10> for a Read access.
- PMD address is the device address.
- Register address is address of the register within that device.
- Line turnaround bits will be <10> for Write accesses and will be <xx> for Read accesses. This allows time for the MII lines to “turn around”.
- Data is the 16 bits of data that will be written to or read from the PMD device.

A reset frame is also provided and defined as 32 consecutive 1s (FFFF FFFFh). After power up, all MII PMD devices must wait for a reset frame to be received prior to participating in MII management communication. Additionally, a reset frame may be issued at any time to allow all connected PMDs to re-synchronize to the data traffic.

The MII/EEPROM Access Register (MEAR) is used to provide access to the serial MII.

Refer to Section 4.2.3 for complete details of the MEAR.

3.0 Functional Description (Continued)

3.9 EEPROM Interface

The DP83820 supports the attachment of an external EEPROM. The EEPROM interface provides the ability for the DP83820 to read from and write data to an external serial EEPROM device. Values in the external EEPROM allow default fields in PCI configuration space and I/O space to be overridden following a hardware reset. The DP83820 will "autoload" values from the EEPROM to these fields in configuration space and I/O space and perform a checksum to verify that the data is valid. If the EEPROM is not present, the DP83820 initialization uses default values for the appropriate Configuration and Operational Registers. Software can read and write to the EEPROM using "bit-bang" accesses via the MII/EEPROM Access Register (MEAR).

3.10 Boot ROM Interface

The BIOS ROM interface allows the DP83820 to read from and write data to an external PROM/Flash device.

3.11 Power Management and Wake Functions

The DP83820 is compliant with the PCI Power Management Specification v1.1. The device can be programmed to any of the powered states (D0, D1, D2, D3hot) and enabled to assert its PMEN pin through the Configuration Register PMCSR. In addition, the device will enter the D3cold state when PCI power is dropped, regardless of the programmed power state. In either D3hot or D3cold, if PMEN assertion is enabled, the device will keep the receiver alive so that it may recognize wake packets and signal the system to wake up; if PMEN assertion is not enabled, the device will go to sleep and be unable to receive packets.

The DP83820 supports several types of wake events that will signal the power management logic to assert PMEN. These are detailed in the Wake On LAN section (4.2.18.1).

In order for the device to request a system wake, at least one wake event must be configured in the Wake Command and Status Register (WCSR). If PMEN assertion is enabled and the device enters the D3cold state with no wake events enabled, the device will go to sleep.

When the device is in a power management state other than D0 (the fully alive state), the only PCI bus activity it may initiate is the assertion of PMEN. This means any packets received will remain in the receive FIFO until the device is returned to the fully alive state. Upon waking up, the wake packet is available in the receive FIFO.

In any power state, enabling PMEN assertion adds additional packet filtering: only those packet types that are configured as wake packets in WCSR will be accepted. This prevents non-wake packets from filling the receive FIFO while the device is in a low power state and preventing a wake packet from being accepted. It is expected that while in the fully alive state, PMEN assertion will be disabled to eliminate the extra level of filtering.

3.12 Network Management Functions

The DP83820 allows compliance with several layer management standards to allow a node to monitor overall network performance. These standards are:

- RFC 1213 (MIB II),
- RFC 1643 (Ether-like MIB), and
- IEEE 802.2 Layer Management.

Many of the counters required by these standards are easily maintained in software during normal per-packet processing. Those counters that would either be difficult or impossible for software to maintain are provided for in hardware (See Section 4.2.27). The table below outlines each required counter, the relevant standard, and how the counter should be maintained.

Table 3-1 MIB Compliance

| Counter Name | Reference | Maintained by | Derivation |
|-----------------|--------------------|--|---|
| RXOctetsOK | RFC 1213, 802.3 LM | software, add cmdsts.SIZE on receive packets with cmdsts.OK bit set. | The byte count of each successfully received packet is added to this counter. The packet byte count includes the address, type, data, and FCS fields. |
| RXFramesOK | 802.3 LM | software, increment on receive packets with cmdsts.OK bit set. | This counter is incremented for each packet successfully received (this includes broadcast, multicast, and physical address packets). |
| RXBroadcastPkts | RFC 1213, 802.3 LM | software, increment on receive packets with cmdsts.OK set and cmdsts.DEST set to 11. | This counter is incremented for each broadcast packet successfully received. |
| RXMulticastPkts | RFC 1213, 802.3 LM | software, increment on receive packets with cmdsts.OK set and cmdsts.DEST set to 10. | This counter is incremented for each multicast packet successfully received. |
| RXErroredPkts | RFC 1213 | hardware, see MIB:RxErroredPkts. | This counter is incremented for each packet received with errors. This count includes packets which are automatically rejected from the FIFO due to both wire errors and FIFO overruns. |

3.0 Functional Description (Continued)

| | | | |
|-------------------|------------------------------|---|--|
| RXFCSErrors | RFC 1643, 802.3 LM | hardware, see MIB:RXFCSErrors. | This counter is incremented for each packet received with a Frame Check Sequence error (bad CRC). |
| RXMsdPktErrors | RFC 1213, RFC 1643, 802.3 LM | hardware, see MIB:RXMsdPktErrors. | This counter is incremented for each receive aborted due to data or status FIFO overruns (insufficient buffer space). |
| RXFAEErrors | RFC 1643, 802.3 LM | hardware, see MIB:RXFAEErrors. | This counter is incremented for each packet received with a Frame Alignment error. |
| RXSymbolErrors | 802.3 LM | hardware, see MIB:RXSymbolErrors | This counter is incremented for each packet received with one or more 100 Mb symbol errors detected. |
| RXFrameTooLong | RFC 1643, 802.3 LM | hardware, see MIB:RXFrameTooLong. | This counter is incremented for each packet received with greater than the 802.3 standard maximum length of 1518 bytes. |
| RXIRLErrors | 802.3 LM | hardware, see MIB:RXIRLErrors. | Packets received with In Range Length errors. This counter increments for packets received with a MAC length/type value between 64 and 1518 bytes, inclusive, that does not match the number of bytes received. This counter also increments for packets with a MAC length/type field of less than 64 bytes and more than 64 bytes received. |
| RXBadOpCodes | 802.3 LM | hardware, see MIB:RXBadOpCodes. | Packets received with a valid MAC control type and an opcode for a function that is not supported by the device |
| RXPauseFrames | 802.3 LM | hardware, see MIB:RXPauseFrames. | MAC control Pause frames received. |
| TXOctetsOK | RFC 1213, 802.3 LM | software, add sum of cmdsts.SIZE (+4) on transmit packets with cmdsts.OK bit set. | The byte count of each successfully transmitted packet is added to this counter. The packet byte count includes the address, type, data, and FCS fields. |
| TXFramesOK | 802.3 LM | software, increment on transmit packets with cmdsts.OK bit set. | This counter is incremented for each packet successfully transmitted. This count includes broadcast, multicast, and physical address packets. |
| TXDeferred | RFC 1643, 802.3 LM | software, increment on transmit packets with cmdsts.TD set. | This counter is incremented for each packet transmission which is deferred due to active line conditions (once per packet). |
| TxBroadcastPkts | RFC 1213, 802.3 LM | software, increment on transmit packets with cmdsts.OK set, and destination address set to ff-ff-ff-ff-ff | This counter is incremented for each broadcast packet successfully transmitted. |
| TXMulticastPkts | RFC 1213, 802.3 LM | software, increment on transmit packets with cmdsts.OK set, and LSB of first byte of destination address set. | This counter is incremented for each multicast packet successfully transmitted. |
| TXFrames1Coll | RFC 1643, 802.3 LM | software, increment on transmit packets with cmdsts.CCNT == 1 and cmdsts.OK set. | This counter is incremented for each packet successfully transmitted with 1 in-window collision. |
| TXFramesMultiColl | RFC 1643, 802.3 LM | software, increment on transmit packets with cmdsts.CCNT > 1 and cmdsts.OK set. | This counter is incremented for each packet successfully transmitted with 2-15 in-window collisions. |
| TXPauseFrames | 802.3 LM | hardware, see MIB:TXPauseFrames. | MAC control Pause frames transmitted. |

3.0 Functional Description (Continued)

| | | | |
|-----------------------|--------------------|--|---|
| TXPktsErrored | RFC 1213 | software, increment on receive packets with cmdsts.TXA set | This counter is incremented for each packet encountering errors during transmission. This count does include transmissions aborted manually and due to FIFO underruns, but does not include packets which experience less than 16 in-window collisions. |
| TXExcessiveCollisions | RFC 1643, 802.3 LM | software, increment on transmit packets with cmdsts.EC set. | This counter is incremented for each transmission aborted after experiencing 16 in-window collisions. |
| TXExcessiveDeferral | 802.3 LM | software, increment on transmit packets with cmdsts.ED set. | This counter is incremented for each transmission aborted due to a time-out of the excessive deferral timer (3.2ms). |
| TXOWC | RFC 1643, 802.3 LM | software, increment on transmit packets with cmdsts.OWC set. | This counter is incremented for each transmission which is aborted due to an out-of-window collision. |
| TXCSErrors | RFC 1643, 802.3 LM | software, increment on transmit packets with cmdsts.CRS set. | This counter is incremented for each transmission on which carrier is not detected after the start of transmission, or carrier sense is lost during transmission. |
| TXSQEErrors | RFC 1643 | hardware, see MIB:TxSQEErrors | This counter is incremented when the collision heartbeat pulse is not detected from by the PMD after a transmission. |

3.13 Buffer Management

The buffer management scheme used on the DP83820 allows quick, simple and efficient use of the frame buffer memory. Frames are saved in similar formats for both transmit and receive. The buffer management scheme also uses separate buffers and descriptors for packet information. This allows effective transfers of data from the receive buffer to the transmit buffer by simply transferring the descriptor from the receive queue to the transmit queue.

The format of the descriptors allows the packets to be saved in a number of configurations. A packet can be stored in memory with a single descriptor and a single packet fragment, or multiple descriptors with single fragments. This flexibility allows the user to configure the DP83820 to maximize efficiency. Architecture of the specific system's buffer memory, as well as the nature of network traffic, will determine the most suitable configuration of packet descriptors and fragments.

3.13.1 Overview

The buffer management design has the following goals:

- simplicity
- efficient use of the PCI bus (the overhead of the buffer management technique is minimal),
- low CPU utilization,
- flexibility.

Descriptors may be either per-packet or per-packet-fragment. Each descriptor may describe one packet fragment. Receive and transmit descriptors are symmetrical.

3.13.2 Descriptor Format

DP83820 uses a symmetrical format for transmit and receive descriptors. In bridging and switching applications this symmetry allows software to forward packets by simply moving the list of descriptors that describe a single received packet from the receive list of one MAC to the transmit list of another. Descriptors must be aligned on a 64-bit boundary.

3.0 Functional Description (Continued)

Table 3-2 DP83820 Descriptor Format

| offset | tag | description |
|----------------|--------|---|
| 0000h | link | 32- or 64-bit "link" field to the next descriptor in the linked list. Bits 2-0 must be 0, as descriptors must be aligned on 64-bit boundaries. |
| 0004h or 0008h | bufptr | 32- or 64-bit pointer to the first fragment or buffer. In transmit descriptors, the buffer can begin on any byte boundary. In receive descriptors, the buffer must be aligned on a 64-bit boundary. |
| 0008h or 0010h | cmdsts | 32-bit Command/Status Field (bit-encoded) |
| 000ch or 0014h | extsts | OPTIONAL 32-bit Extended Status Field. Contains VLAN and IP information. |

If 64-bit addressing is enabled, the *link* and *bufptr* fields are 64-bit fields. Otherwise, they are 32-bit fields. The DP83820 supports an optional extended status field which supports VLAN and IP functions. To enable the *extsts* field, software should set the EXTSTS_EN bit in the CFG register.

Some of the bit definitions in the *cmdsts* field are common to both receive and transmit descriptors:

Table 3-3 cmdsts Common Bit Definitions

| bit | tag | description | usage |
|-------|------------------|----------------------------|--|
| 31 | OWN | Descriptor Ownership | Set to 1 by the <i>data producer</i> of the descriptor to transfer ownership to the <i>data consumer</i> of the descriptor. Set to 0 by the <i>data consumer</i> of the descriptor to return ownership to the <i>data producer</i> of the descriptor. For transmit descriptors, the driver is the <i>data producer</i> , and the DP83820 is the <i>data consumer</i> . For receive descriptors, the DP83820 is the <i>data producer</i> , and the driver is the <i>data consumer</i> . |
| 30 | MORE | More descriptors | Set to 1 to indicate that this is NOT the last descriptor in a packet (there are MORE to follow). When 0, this descriptor is the last descriptor in a packet. Completion status bits are only valid when this bit is zero. |
| 29 | INTR | Interrupt | Set to 1 by software to request a "descriptor interrupt" when DP83820 transfers the ownership of this descriptor back to software. |
| 28 | SUPCRC INCCRC | Suppress CRC / Include CRC | In transmit descriptors, this indicates that CRC should not be appended by the MAC. On receives, this bit will be set based on the RXCFG:INCCRC bit. |
| 27 | OK | Packet OK | In the last descriptor in a packet, this bit indicates that the packet was either sent or received successfully. |
| 26-16 | --- | | The usage of these bits differ in receive and transmit descriptors. See below for details. |
| 15-0 | SIZE | Descriptor Byte Count | Set to the size in bytes of the data. |

Table 3-4 Transmit cmdsts Bit Definitions

| bit | tag | description | usage |
|-----|-----|-------------------------|--|
| 26 | TXA | Transmit Abort | Transmission of this packet was aborted. |
| 25 | TFU | Transmit FIFO Underrun | The transmit FIFO was exhausted during the transmission of this packet. |
| 24 | CRS | Carrier Sense Lost | Carrier was lost during the transmission of this packet. This condition is not reported if TXCFG:CSI is set. |
| 23 | TD | Transmit Deferred | Transmission of this packet was deferred. |
| 22 | ED | Excessive Deferral | The length of deferral during the transmission of this packet was excessive. |
| 21 | OWC | Out of Window Collision | The MAC encountered an "out of window" collision during the transmission of this packet. |

3.0 Functional Description (Continued)

| | | | |
|-------|------|----------------------|---|
| 20 | EC | Excessive Collisions | The number of collisions during the transmission of this packet was excessive, indicating transmission failure. If TXCFG register ECRETRY=0, this bit is set after 16 collisions. If TXCFG register ECRETRY=1, this bit is set after 4 Excessive Collision events (64 collisions). |
| 19-16 | CCNT | Collision Count | If TXCFG register ECRETRY=0, this field indicates the number of collisions encountered during the transmission of this packet. If TXCFG register ECRETRY=1, CCNT[3:2] = Excessive Collisions (0-3) CCNT[1] = Multiple Collisions CCNT[0] = Single Collision Note that Excessive Collisions indicate 16 attempts failed, while Multiple Collisions and Single Collision indicate collisions in addition to any excessive collisions. For example, a collision count of 33 includes 2 Excessive Collisions and will also set the Single Collision bit. |

Table 3-5 Receive cmdsts Bit Definitions

| bit | tag | description | usage |
|-------|------|--------------------------|--|
| 26 | RXA | Receive Aborted | Set to 1 by DP83820 when the receive was aborted. If RXO is set, then the receive was aborted due to an RX overrun. If RXO is clear, then a receive descriptor error occurred. SIZE will be set to the amount of data that was transferred to memory when the error was detected. |
| 25 | RXO | Receive Overrun | Set to 1 by DP83820 to indicate that a receive overrun condition occurred. RXA will also be set. |
| 24-23 | DEST | Destination Class | When the receive filter is enabled, these bits will indicate the destination address class as follows: 00 - Packet was rejected 01 - Destination matched the Receive Filter Node Address Register 10 - Destination is a multicast (but not broadcast) 11 - Destination is a broadcast address If the Receive Filter is enabled, 00 indicates that the packet was rejected. Normally packets that are rejected do not cause any bus activity, nor do they consume receive descriptors. However, this condition could occur if the packet is rejected by the Receive Filter later in the packet than the receive drain threshold (RXCFG:DRTH) |
| 22 | LONG | Too Long Packet Received | The size of the receive packet exceeded 1518 bytes (1522 bytes if VLAN tag included). |
| 21 | RUNT | Runt Packet Received | The size of the receive packet was smaller than 64 bytes (including CRC). |
| 20 | ISE | Invalid Symbol Error | (100 Mb only) An invalid symbol was encountered during the reception of this packet. |
| 19 | CRCE | CRC Error | The CRC appended to the end of this packet was invalid. |
| 18 | FAE | Frame Alignment Error | The packet did not contain an integral number of octets. |
| 17 | LBP | Loopback Packet | The packet is the result of a loopback transmission. |
| 16 | IRL | In-Range Length Error | The receive packet Length/Type field did not match the length of the data field for the packet. Only valid if the Length/Type field is a valid length (not a Type value). |

3.0 Functional Description (Continued)

Table 3-6 Transmit extsts Bit Definitions

| bit | tag | description | usage |
|-------|--------|------------------------------|---|
| 31-22 | | | unused |
| 21 | UDPPKT | UDP Packet | Indicates packet contains a UDP header and enables checksum generation for the UDP header if Checksumming is enabled on a per-packet basis. |
| 20 | | | unused |
| 19 | TCPPKT | TCP Packet | Indicates packet contains a TCP header and enables checksum generation for the TCP header if Checksumming is enabled on a per-packet basis. |
| 18 | | | unused |
| 17 | IPPKT | IP Packet | Indicates packet contains a IP header and enables checksum generation for the IP header if Checksumming is enabled on a per-packet basis. |
| 16 | VPKT | VLAN Packet | Insert VLAN tag. |
| 15-0 | VTCI | VLAN Tag Control Information | This is the VLAN TCI field to be inserted in the packet if the VPKT bit is set. |

Table 3-7 Receive extsts Bit Definitions

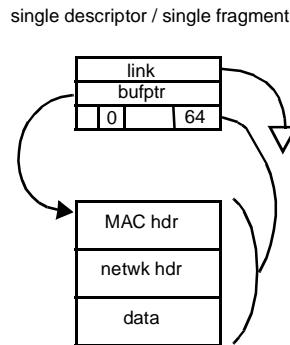
| bit | tag | description | usage |
|-------|--------|------------------------------|--|
| 31-23 | | | unused |
| 22 | UDPERR | UDP Checksum Error | Indicates a checksum error was detected in the UDP header. |
| 21 | UDPPKT | UDP Packet | Indicates an UDP header was detected for the packet. |
| 20 | TCPERR | TCP Checksum Error | Indicates a checksum error was detected in the TCP header. |
| 19 | TCPPKT | TCP Packet | Indicates an TCP header was detected for the packet. |
| 18 | IPERR | IP Checksum Error | Indicates a checksum error was detected in the IP header. |
| 17 | IPPKT | IP Packet | Indicates an IP header was detected for the packet. |
| 16 | VPKT | VLAN Packet | Packet contained a VLAN tag. This bit will be set if VLAN packet detection is enabled and the packet contained the correct type value. |
| 15-0 | VTCI | VLAN Tag Control Information | This is the VLAN TCI field to be extracted from the packet. It contains the user_priority, CFI, and VID fields. |

3.0 Functional Description (Continued)

3.13.2.1 Single Descriptor Packets

To represent a packet in a single descriptor, the MORE bit in the `cmdsts` field is set to 0.

Figure 3-10 Single Descriptor Packets

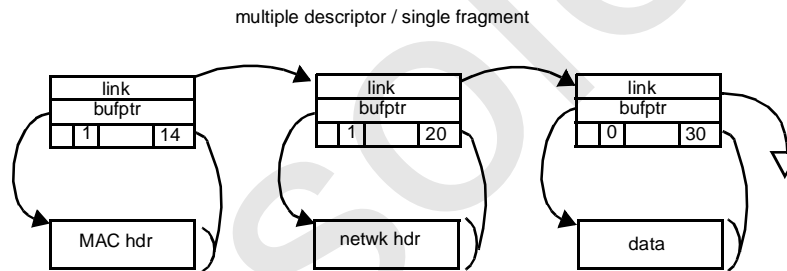


3.13.2.2 Multiple Descriptor Packets

A single packet may also cross descriptor boundaries. This is indicated by setting the MORE bit in all descriptors except the last one in the packet. Ethernet inter-networking applications (bridges, switches, routers, etc.) can optimize

memory utilization by using a single small buffer per receive descriptor, and allowing the DP83820 hardware to use the minimum number of buffers necessary to store an incoming packet.

Figure 3-11 Multiple Descriptor Packets



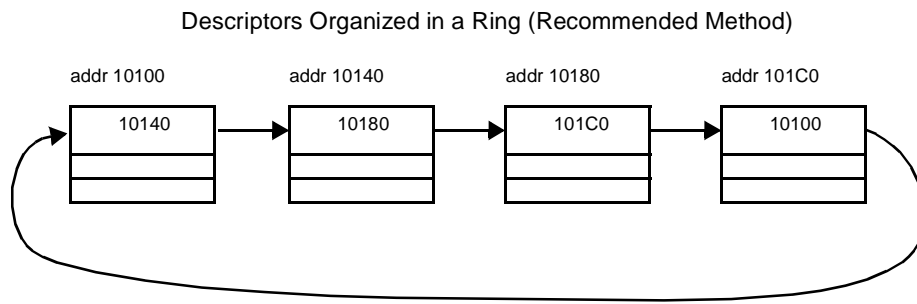
3.13.2.3 Descriptor Rings

The simplest and recommended organization of descriptors is in a fixed ring implementation. At initialization, the driver can set up a fixed list of descriptors complete with links connecting the descriptors in a ring. All descriptors will initially be owned by the producer of the data (the driver for transmit, the DP83820 for receive). The OWN bit is used by both driver and the DP83820 to

indicate data availability and to release descriptors back to the producer. When using a descriptor ring, the driver should never need to modify any fields of a descriptor it does not own. For transmit, the driver should never assign all descriptors to the device, reserving one descriptor to terminate the list, preventing the device from wrapping completely around the ring.

3.0 Functional Description (Continued)

Figure 3-12 Ring Descriptor Organization

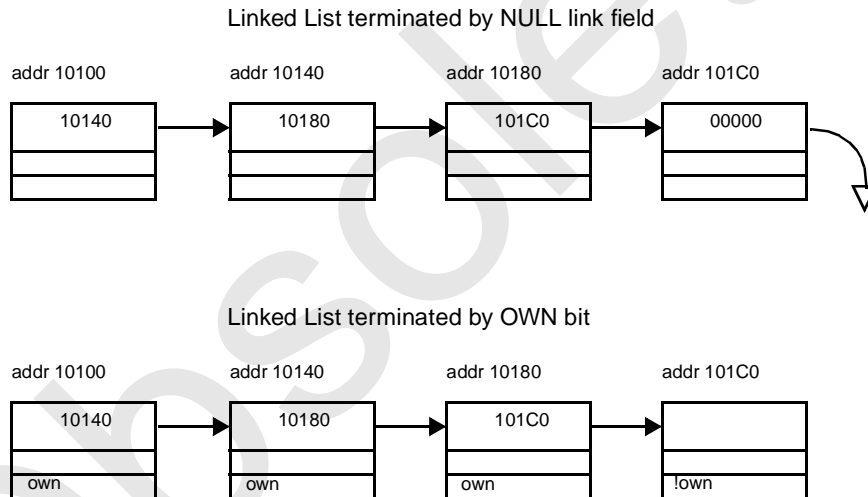


3.13.2.4 Descriptor Lists

Descriptors may also be organized in linked lists using the link field. The linked list may be terminated by either a NULL link field, or by using the descriptor OWN bit. A list of descriptors may represent any number of packets or packet

fragments. Care should be used when implementing a linked list terminated by a NULL link as there is a potential for driver software and the device to get out of sync. Before clearing a link field when freeing up descriptors, the driver should verify that the device has already traversed the link.

Figure 3-13 Linked List Descriptor Organization



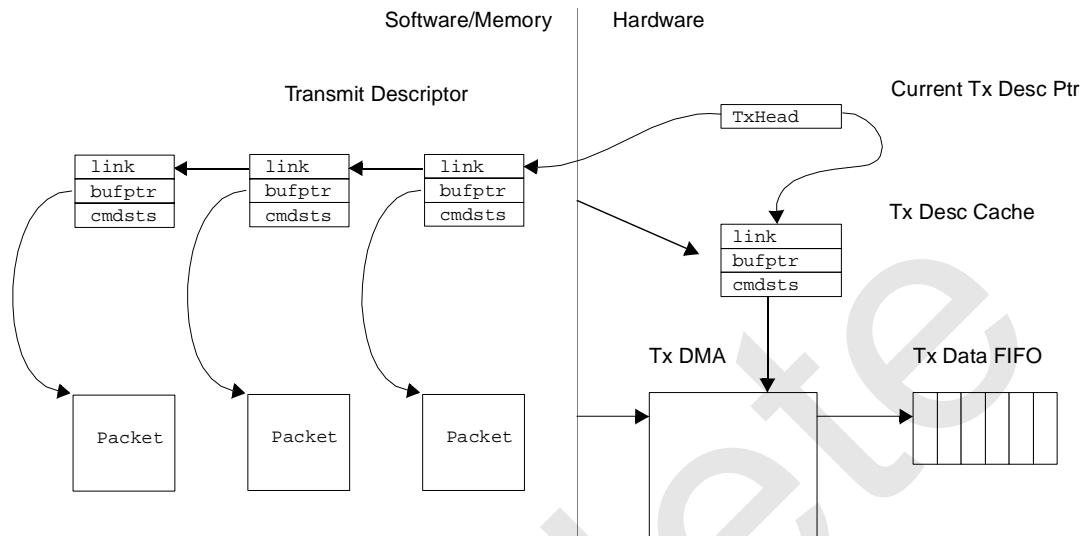
3.0 Functional Description (Continued)

3.13.3 Transmit Architecture

The Transmit architecture can support a single transmit queue, or can support multiple transmit queues for

handling priority traffic. The following figures illustrate the transmit architecture of the DP83820 10/100 Ethernet Controller with and without Priority Queueing.

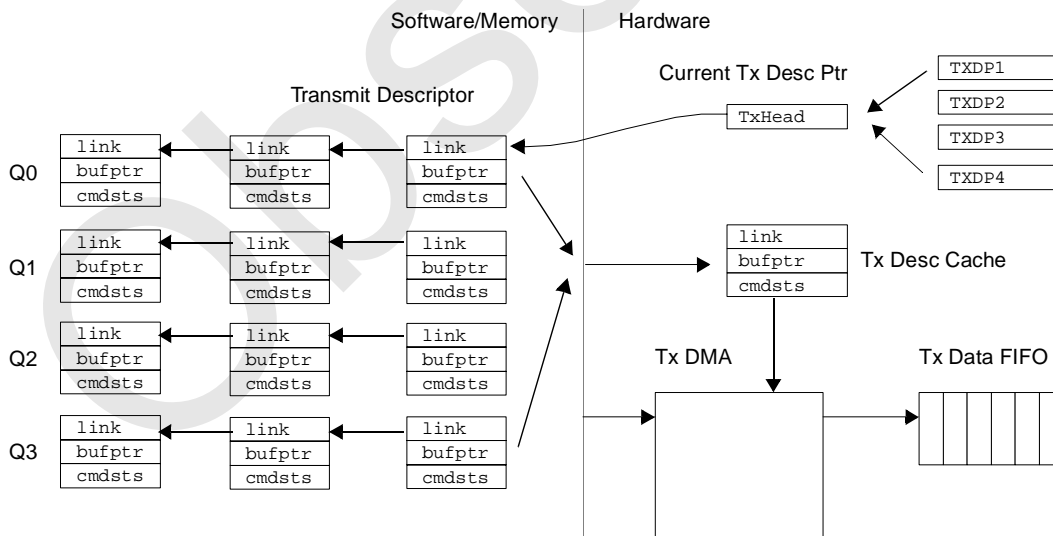
Figure 3-14 Transmit Architecture without Priority Queueing



Without Priority Queueing, the device will draw packets from a single Descriptor list. Only one descriptor pointer is required. When the CR:TXEN bit is set to 1 (regardless of the current state), and the DP83820 transmitter is idle, then

DP83820 will read the contents of the current transmit descriptor into the TxDescCache. The DP83820's TxDescCache can hold a single fragment pointer/count combination.

Figure 3-15 Transmit Architecture with Priority Queueing



With Priority Queueing, the device will draw packets from up to 4 Descriptor lists. The device has four descriptor pointers and associated control logic to keep track of when descriptors are available with valid packet information. In this case, pulsing CR:TXEN with CR:TXPRI[p] set will indicate to the DP83820 that a descriptor is available for

descriptor queue of priority 'p'. Based on the priority algorithm in use, the device will draw from the current highest priority descriptor that has packets available for transmission. There is no reordering of packets once they are queued within the internal FIFO.

3.0 Functional Description (Continued)

3.13.3.1 Transmit State Machine

The transmit state machine has the following states:

| | |
|-------------|---|
| txIdle | The transmit state machine is idle. |
| txDescRefr | Waiting for the "refresh" transfer of the link field of a completed descriptor from the PCI bus. |
| txDescRead | Waiting for the transfer of a complete descriptor from the PCI bus into the TxDescriptorCache. |
| txFifoBlock | Waiting for free space in the TxDataFIFO to reach TxFillThreshold. |
| txFragRead | Waiting for the transfer of a fragment (or portion of a fragment) from the PCI bus to the TxDataFIFO. |
| txDescWrite | Waiting for the completion of the write of the <code>cmdsts</code> field of an intermediate transmit descriptor (<code>cmdsts.MORE == 1</code>) to host memory. |
| txAdvance | (transitory state) Examine the link field of the current descriptor and advance to the next descriptor if link is not NULL. |

The transmit state machine manipulates the following internal data spaces:

| | |
|-------------|---|
| TXDP | A 32- or 64-bit register that points to the current transmit descriptor. If priority queueing is enabled, this points to the available transmit descriptor with the highest priority. |
| CTDD | Current Transmit Descriptor Done. An internal bit flag that is set when the current transmit descriptor has been completed, and ownership has been returned to the driver. It is cleared whenever TXDP is loaded with a new value (either by the state machine, or the driver). |
| TxDescCache | An internal data space equal to the size of the maximum transmit descriptor supported. |
| descCnt | Count of bytes remaining in the current descriptor. |
| fragPtr | Pointer to the next unread byte in the current fragment. |
| txFifoCnt | Current amount of data in the txDataFifo in bytes. |
| txFifoAvail | Current amount of free space in the txDataFifo in bytes (size of the txDataFifo - txFifoCnt). |

Inputs to the transmit state machine include the following events:

| | |
|-----------|--|
| CR:TXEN | Driver asserts the TXEN bit in the command register. If priority queueing is enabled, this corresponds to a specific priority queue. |
| XferDone | Completion of a PCI bus transfer request. |
| FifoAvail | TxFifoAvail is greater than TxFillThreshold. |

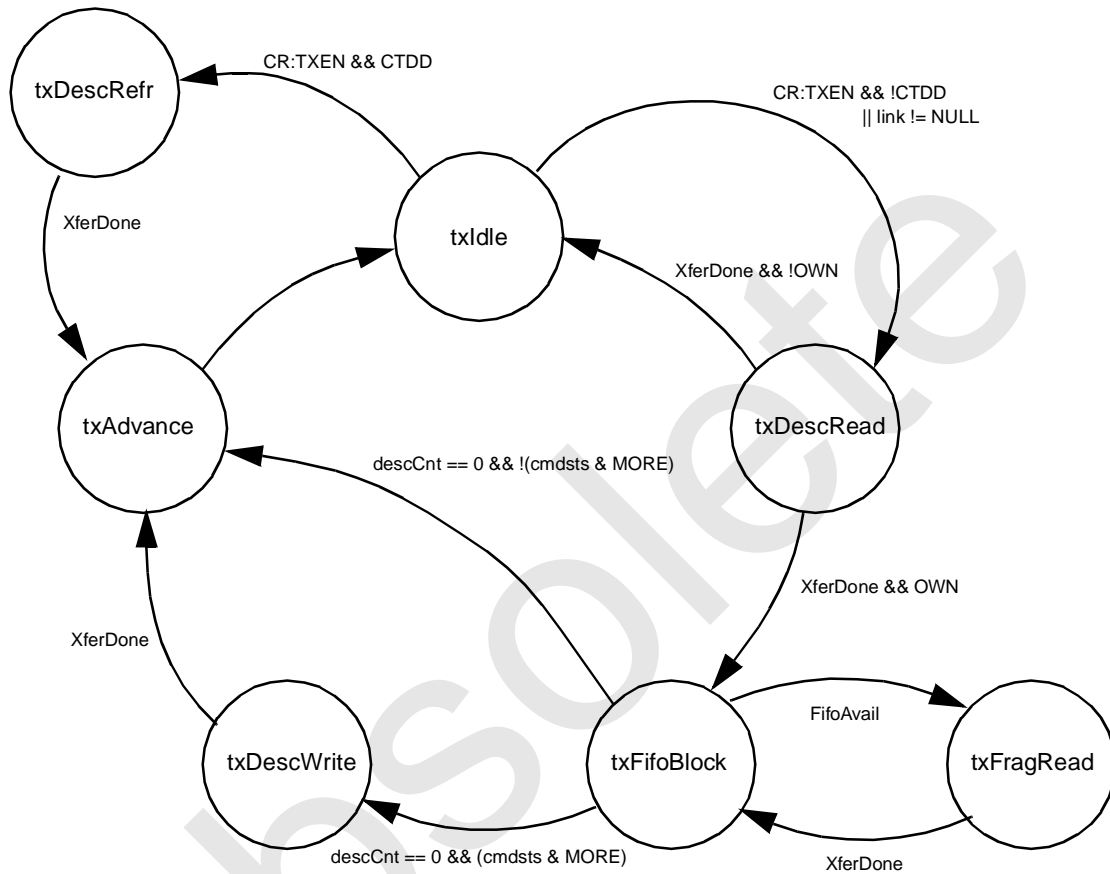
Table 3-8 Transmit State Tables

| state | event | next state | actions |
|-------------|-------------------------|-------------|---|
| txIdle | CR:TXEN && !CTDD | txDescRead | start a burst transfer at address TXDP and a length derived from TXCFG. |
| | CR:TXEN && CTDD | txDescRefr | start a burst transfer to refresh the link field of the current descriptor. |
| txDescRefr | XferDone | txAdvance | |
| txDescRead | XferDone && OWN | txFIFOblock | |
| | XferDone && !OWN | txIdle | set ISR:TXIDLE. |
| txFIFOblock | FifoAvail | txFragRead | start a burst transfer into the TxDataFIFO from fragPtr. The length will be the minimum of txFifoAvail and descCnt. Decrement descCnt accordingly. |
| | (descCnt == 0) && MORE | txDescWrite | start a burst transfer to write the status back to the descriptor, clearing the OWN bit. |
| | (descCnt == 0) && !MORE | txAdvance | write the value of TXDP to the txDataFIFO as a handle. |
| txFragRead | XferDone | txFIFOblock | |
| txDescWrite | XferDone | txAdvance | |

3.0 Functional Description (Continued)

| | | | |
|-----------|--------------|------------|--|
| txAdvance | link != NULL | txDescRead | TXDP <- txDescCache.link. Clear CTDD. Start a burst transfer at address TXDP with a length derived from TXCFG. |
| | link == NULL | txIdle | set CTDD, set ISR:TXIDLE. |

Figure 3-16 Transmit State Diagram



3.13.3.2 Transmit Data Flow without Priority Queueing

In the DP83820 transmit architecture without Priority Queueing, packet transmission involves the following steps:

1. The device driver receives packets from an upper layer.
2. An available DP83820 transmit descriptor is allocated. The fragment information is copied from the NOS specific data structure(s) to the next DP83820 transmit descriptor.
3. The driver adds this descriptor to its internal list of transmit descriptors awaiting transmission.
4. If the internal list was empty (this descriptor represents the only outstanding transmit packet), then the driver must set the TXDP register to the address of this descriptor, else the driver will append this descriptor to the end of the list.
5. The driver sets the TXEN bit in the CR register to insure that the transmit state machine is active.
6. If idle, the transmit state machine reads the descriptor into the TxDescriptorCache. If the OWN bit is not set, the transmit state machine returns to idle to wait for TXEN to be set again.
7. The state machine then moves through the fragment described within the descriptor, filling the TxDataFifo with data. The hardware handles all aspects of byte alignment; no alignment is assumed. Fragments may start and/or end on any byte address. The transmit state machine uses the fragment pointer and the SIZE field from the cmdsts field of the current descriptor to keep the TxDataFifo full. It also uses the MORE bit and the SIZE field from the cmdsts field of the current descriptor to know when packet boundaries occur.
8. When a packet has completed transmission (successful or unsuccessful), the state machine updates the cmdsts field of the current descriptor in main memory (by bus-mastering a single 32-bit word), relinquishing ownership, and indicating the packet completion status. If more than one descriptor

3.0 Functional Description (Continued)

was used to describe the packet, then completion status is updated only in the last descriptor. Intermediate descriptors only have the OWN bits modified.

9. If the link field of the descriptor is non-zero, the state machine advances to the next descriptor and continues. When reading the next descriptor, if the OWN bit is not set, the state machine will halt and wait for TXEN to be set again.
10. If the link field is NULL, the transmit state machine suspends, waiting for the TXEN bit in the CR register to be set. If the TXDP register is written to, the CTDD flag will be cleared. When the TXEN bit is set, the state machine will examine CTDD. If CTDD is set, the state machine will "refresh" the link field of the current descriptor. It will then follow the link field to any new descriptors that have been added to the end of the list. If CTDD is clear (implying that TXDP has been written to), the state machine will start by reading in the descriptor pointed to by TXDP.

3.13.3.3 Transmit Data Flow with Priority Queueing

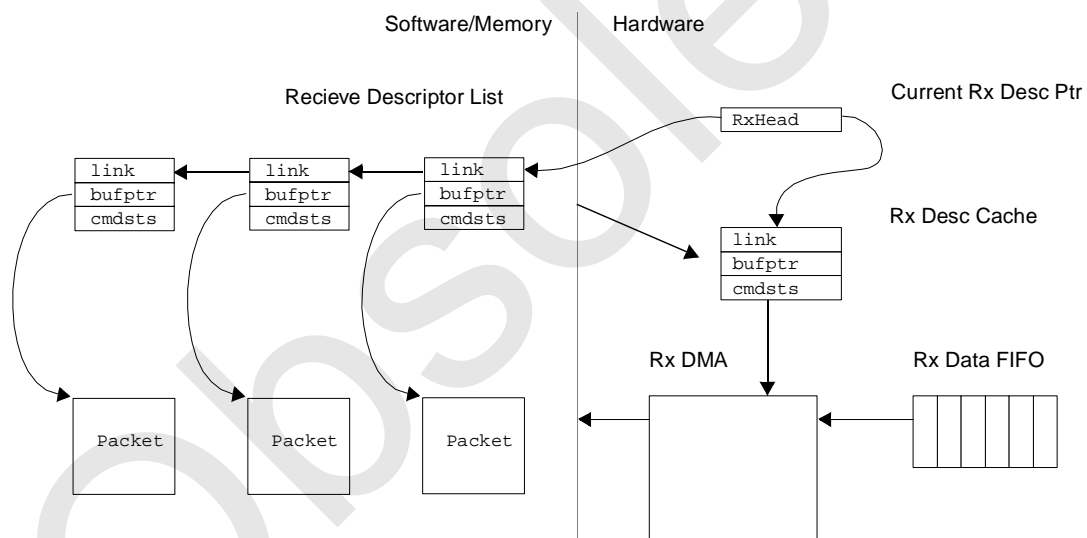
The transmit architecture with Priority Queueing is the same with a few minor differences:

- Driver keeps a separate list for each descriptor queue.
- When setting the TXEN bit, the driver must also set the appropriate TXPRI bit for the priority queue or queues to which descriptors are being appended.
- Upon completion of a packet, the transmit state machine first determines what the highest priority descriptor is available based on non-zero link fields and TXEN bits. It then follows the appropriate link or reads a new descriptor for the next packet to be transmitted.

3.13.4 Receive Architecture

The receive architecture is as "symmetrical" to the transmit architecture as possible. As is done in the transmitter, the receive architecture can support a single descriptor queue or multiple descriptor queues for handling priority traffic. When the amount of receive data in the RxDataFIFO is more than the RxDrainThreshold, or the RxDataFIFO contains a complete packet, then the state machine begins filling received buffers in host memory.

Figure 3-17 Receive Architecture without Priority Queueing

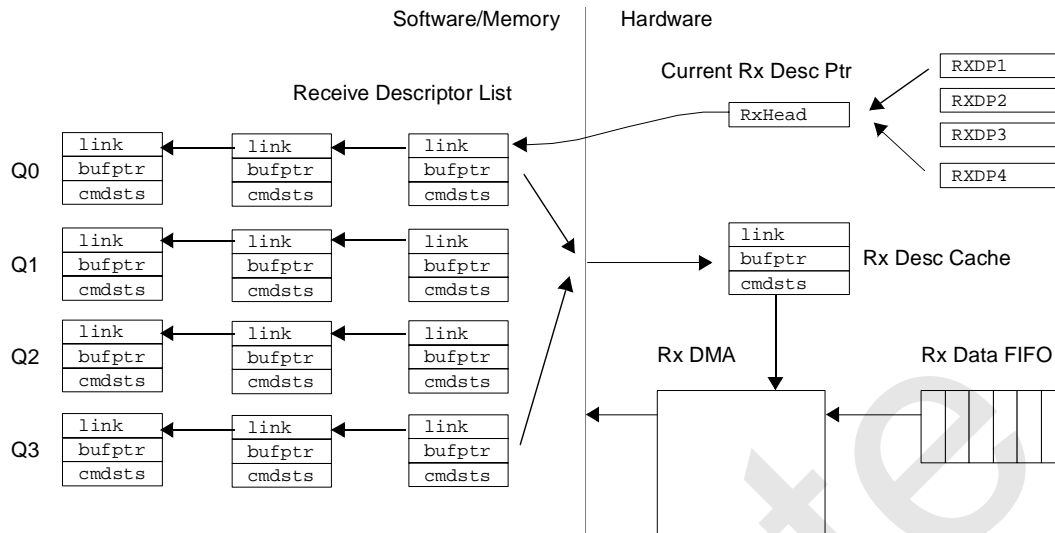


Without Priority Queueing, the device will transfer packets to a single Descriptor list. Only one descriptor pointer is required. The receive buffer manager prefetches receive descriptors to prepare for incoming packets. When the RXEN bit is set to 1 in the CR register (regardless of the current state), and the DP83820 receive state machine is idle, then DP83820 will read the contents of the descriptor

referenced by RXDP into the Rx Descriptor Cache. The Rx Descriptor Cache allows the DP83820 to read an entire descriptor in a single burst, and reduces the number of bus accesses required for fragment information to 1. The DP83820 Rx Descriptor Cache holds a single buffer pointer/count combination.

3.0 Functional Description (Continued)

Figure 3-18 Receive Architecture with Priority Queueing



With Priority Queueing, the device will transfer packets onto up to 4 Descriptor lists. The device has four descriptor pointers and associated control logic to keep track of when descriptors are available with valid packet information. The Receiver uses the user_priority field of a VLAN tag to determine the priority, based on the 802.1Q encodings based on the number of priority queues enabled. If the

packet has no VLAN tag, then a priority of 0 is assumed. There is no reordering of packets while in the Receive Data FIFO.

3.13.5 Receive State Machine

The receive state machine has the following states:

| | |
|-------------|---|
| rxIdle | The receive state machine is idle. |
| rxDescRefr | Waiting for the "refresh" transfer of the link field of a completed descriptor from the PCI bus. |
| rxDescRead | Waiting for the transfer of a descriptor from the PCI bus into the RxDescCache. |
| rxFifoBlock | Waiting for the amount of data in the RxDataFifo to reach the RxDrainThreshold or to represent a complete packet. |
| rxFragWrite | Waiting for the transfer of data from the RxDataFIFO via the PCI bus to host memory. |
| rxDescWrite | Waiting for the completion of the write of the cmdsts field of a receive descriptor. |

The receive state machine manipulates the following internal data spaces:

| | |
|-------------|--|
| RXDP | A 32- or 64-bit register that points to the current receive descriptor. |
| CRDD | An internal bit flag that is set when the current receive descriptor has been completed, and ownership has been returned to the driver. It is cleared whenever RXDP is loaded with a new value (either by the state machine, or the driver). |
| RxDescCache | An internal data space equal to the size of the maximum receive descriptor supported. |
| descCnt | Count of bytes available for storing receive data in all of the fragments described by the current descriptor. |
| fragPtr | Pointer to the next unwritten byte in the current fragment. |
| rxPktCnt | Number of packets in the rxDataFifo. Incremented by the MAC (the fill side of the FIFO). Decremented by the receive state machine as packets are processed. |
| rxPktBytes | Number of bytes in the current packet being drained from the rxDataFifo, that are in fact currently in the rxDataFifo (Note: for packets larger than the FIFO size, this number will never be greater than the FIFO size). |

Inputs to the receive state machine include the following events:

| | |
|----------|--|
| CR:RXEN | The RXEN bit in the Command Register has been set. |
| XferDone | completion of a PCI bus transfer request. |

3.0 Functional Description (Continued)

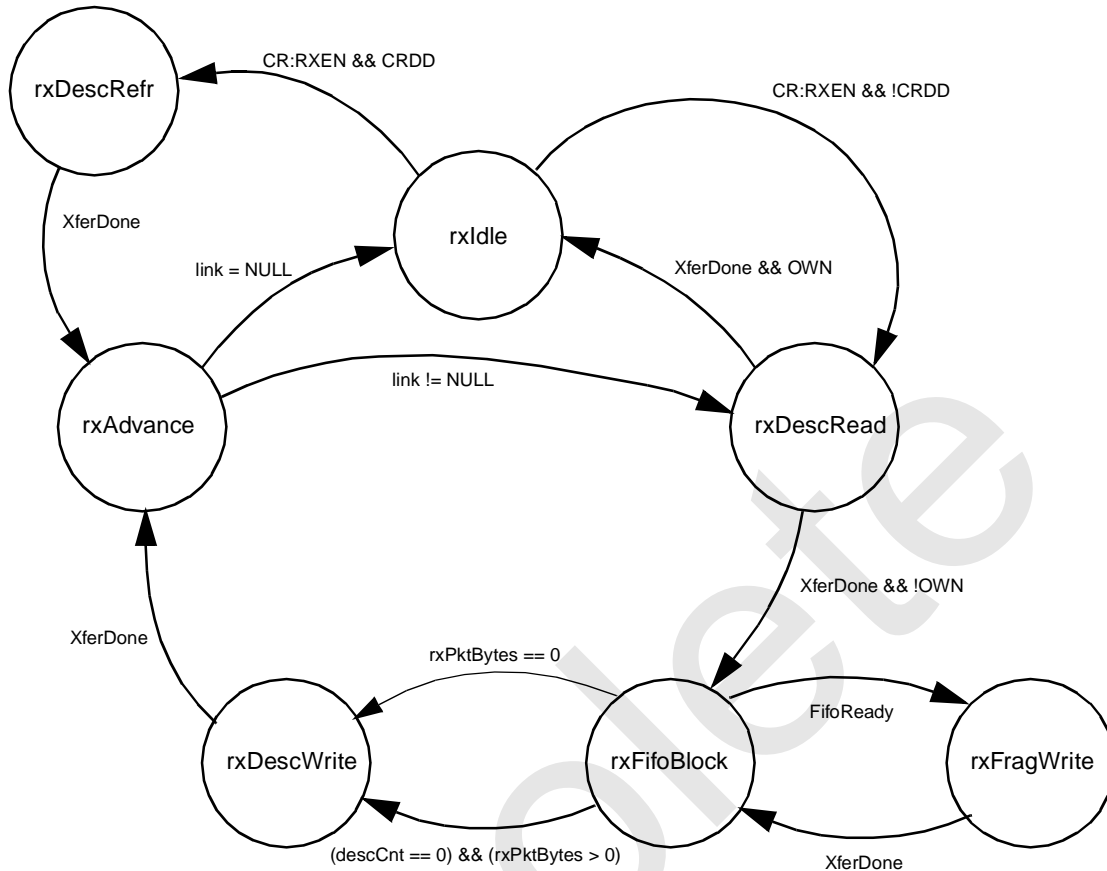
FifoReady (rxPktCnt > 0) or (rxPktBytes > rxDrainThreshold) ... in other words, if we have a complete packet in the FIFO (regardless of size), or the number of bytes that we do have is greater than the rxDrainThreshold, then we are ready to begin draining the rxDataFifo.

Table 3-9 Receive State Tables

| state | event | next state | actions |
|-------------|------------------------------------|-------------|---|
| rxIdle | CR:RXEN && !CRDD | rxDescRead | start a burst transfer at address RXDP and a length derived from RXCFG. |
| | CR:RXEN && CRDD | rxDescRefr | start a burst transfer to refresh the link field of the current descriptor. |
| rxDescRefr | XferDone | rxAdvance | |
| rxDescRead | XferDone && !OWN | rxFIFOblock | |
| | XferDone && OWN | rxIdle | set ISR:RXIDLE. |
| rxFIFOblock | FifoReady | rxFragWrite | start a burst transfer from the RxDataFIFO to host memory at fragPtr. The length will be the minimum of rxPktBytes and descCnt. Decrement descCnt accordingly. |
| | (descCnt == 0) && (rxPktBytes > 0) | rxDescWrite | start a burst transfer to write the status back to the descriptor, setting the OWN bit, and setting the MORE bit. We'll continue the packet in the next descriptor. |
| | rxPktBytes == 0 | rxDescWrite | start a transfer to write the cmdsts back to the descriptor, setting the OWN bit and clearing the MORE bit, and filling in the final receive status (CRC, FAE, SIZE, etc.). |
| rxFragWrite | XferDone | rxFIFOblock | |
| rxDescWrite | XferDone | rxAdvance | |
| rxAdvance | link != NULL | rxDescRead | RXDP <- rxDescCache.link. Clear CRDD. Start a burst transfer at address RXDP with a length derived from RXCFG:MAXF. |
| | link == NULL | rxIdle | set CRDD. Set ISR:RXIDLE. |

3.0 Functional Description (Continued)

Figure 3-19 Receive State Diagram



3.13.5.1 Receive Data Flow without Priority Queueing

With a bus mastering architecture, some number of buffers and descriptors for received packets must be pre-allocated when the DP83820 is initialized. The number allocated will directly affect the system's tolerance to interrupt latency. The more buffers that you pre-allocate, the longer the system will survive an incoming burst without losing receive packets, if receive descriptor processing is delayed or preempted. The following describes the Receive data flow without Priority Queueing:

1. Prior to packet reception, receive buffers must be described in a receive descriptor ring (or list, if preferred). In each descriptor, the driver assigns ownership to the hardware by clearing the OWN bit. Receive descriptors may describe a single buffer.
2. The address of the first descriptor in this list is then written to the RXDP register. As packets arrive, they are placed in available buffers. A single packet may occupy one or more receive descriptors, as required by the application. The device reads in the first descriptor into the RxDescCache.

3. As data arrives in the RxDataFIFO, the receive buffer management state machine places the data in the receive buffer described by the descriptor. This continues until either the end of packet is reached, or the descriptor byte count for this descriptor is reached.
4. If end of packet was reached, the status in the descriptor (in main memory) is updated by setting the OWN bit and clearing the MORE bit, by updating the receive status bits as indicated by the MAC, and by updating the SIZE field. The status bits in `cmd.st.s` are only valid in the last descriptor of a packet (with the MORE bit clear). Also for the last descriptor of a packet, the SIZE field will be updated to reflect the actual amount of data written to the buffer (which may be less the full buffer size allocated by the descriptor).

If the receive buffer management state machine runs out of descriptors while receiving a packet, data will buffer in the receive FIFO. If the FIFO overflows, the driver will be interrupted with an RxOVR error.

3.0 Functional Description (Continued)

3.13.5.2 Receive Data Flow with Priority Queueing

With Priority Queueing, it is still necessary to pre-allocate buffers and descriptors. Each priority queue must have a separate list of descriptors allocated. The receive data flow is similar to the above with the following exceptions:

— The Receive state machine waits until packet data is available, and the priority for the packet has been determined from a VLAN tag (assumes priority 0 if no tag is present). Using this information, the receive state machine will then process descriptors as detailed above.

— If no descriptors are available for the priority queue matching the packet priority, the state machine will wait for the system to append more descriptors to the descriptor list and pulse the CR:RXEN and CR:RXPRI controls.

— If no descriptors are available for the priority queue matching the packet priority, the state machine will wait for the system to append more descriptors to the descriptor list and pulse the CR:RXEN and CR:RXPRI controls.

Obsolete

4.0 Register Set

4.1 Configuration Registers

The DP83820 implements a PCI version 2.2 configuration register space. This allows a PCI BIOS to "soft" configure the DP83820. Software Reset has no effect on configuration registers. Hardware Reset returns all configuration registers to their hardware reset state. For all unused registers, writes are ignored, and reads return 0.

Table 4-1 Configuration Register Map

| offset | tag | description | access |
|---------|--------|--|--------|
| 00h | CFGID | Configuration Identification Register | RO |
| 04h | CFGCS | Configuration Command and Status Register | R/W |
| 08h | CFGRID | Configuration Revision ID Register | RO |
| 0Ch | CFGLAT | Configuration Latency Timer Register | RO |
| 10h | CFGIOA | Configuration IO Base Address Register | R/W |
| 14h | CFGMA | Configuration Memory Address Register | R/W |
| 18 | CFGMA1 | Configuration Memory Address High Dword Register | R/W |
| 1Ch-28h | | Reserved (reads return zero) | |
| 2Ch | CFGSID | Configuration Subsystem Identification Register | RO |
| 30h | CFGROM | Boot ROM configuration register | R/W |
| 34h | CAPPTR | Capabilities Pointer Register | RO |
| 38h | | Reserved (reads return zero) | |
| 3Ch | CFGINT | Configuration Interrupt Select Register | R/W |
| 40h | PMCAP | Power Management Capabilities Register | RO |
| 44h | PMCS | Power Management Control and Status Register | R/W |
| 48-FFh | | Reserved (reads return zero) | |

4.1.1 Configuration Identification Register

This register identifies the DP83820 Controller to PCI system software.

Tag: CFGID

Size: 32 bits

Hard Reset: 0022100B

Offset: 00h

Access: Read Only

Soft Reset: Unchanged

| bit | tag | description | usage |
|-------|-------|-------------|---|
| 31-16 | DEVID | Device ID | This field is read-only and is set to the device ID assigned by NSC to the DP83820, which is 0022h. |
| 15-0 | VENID | Vendor ID | This field is read-only and is set to a value of 100Bh which is National Semiconductor's PCI Vendor ID. |

4.0 Register Set (Continued)

4.1.2 Configuration Command and Status Register

The CFGCS register has two parts. The upper 16-bits (31-16) are devoted to device status. The lower 16-bits (15-0) are devoted to command and are used to configure and control the device.

Tag: CFGCS **Size:** 32 bits **Hard Reset:** 02900000h
Offset: 04h **Access:** Read Write **Soft Reset:** Unchanged

| bit | tag | description | usage |
|-------|---------|------------------------------------|---|
| 31-16 | STS | Status | Device Status Bits. A status bit is reset whenever the register is written, and the corresponding bit location is a 1. |
| 31 | DPERR | Detected Parity Error | Refer to the description in the PCI V2.2 specification. |
| 30 | SSERR | Signaled SERR | Refer to the description in the PCI V2.2 specification. |
| 29 | RMABT | Received Master Abort | Refer to the description in the PCI V2.2 specification. |
| 28 | RTABT | Received Target Abort | Refer to the description in the PCI V2.2 specification. |
| 27 | STABT | Sent Target Abort | Refer to the description in the PCI V2.2 specification. |
| 26-25 | DSTIM | DEVSELN Timing | This field will always be set to 01 indicating that DP83820 supports "medium" DEVSELN timing. |
| 24 | DPD | Data Parity Detected | Refer to the description in the PCI V2.2 specification. |
| 23 | FBB | Fast Back-to-Back Capable | DP83820 will set this bit to 1. |
| 22 | | | unused (reads return 0) |
| 21 | M66_CAP | 66MHz Capable | This field indicates the device is 66MHz capable. It will be loaded from EEPROM. |
| 20 | NCPEN | New Capabilities Enable | When set, this bit indicates that the Capabilities Pointer contains a valid value and new capabilities such as power management are supported. When clear, new capabilities (CAPPTR, PMCAP, PMCS) are disabled. The value in this register will either be loaded from the EEPROM or, if the EEPROM is disabled, from a strap option at reset. |
| 19-16 | | Unused | Unused (reads return 0) |
| 15-0 | CMD | Command | Device Command bits (see below). |
| 15-10 | | Unused | Unused (reads return 0) |
| 9 | FBBEN | Fast Back-to-Back Enable | Set to 1 by the PCI BIOS to enable the DP83820 to do Fast Back-to-Back transfers (FBB transfers as a master is not implemented in the current revision). |
| 8 | SERREN | SERRN Enable | When set, DP83820 will generate SERRN when an address parity error is detected. |
| 7 | | Unused | Unused (reads return 0) |
| 6 | PERRSP | Parity Error Response | When set, DP83820 will assert PERRN on the detection of a data parity error when acting as the target, and will sample PERRN when acting as the initiator. When reset, data parity errors are ignored. The action taken is specified by CFG: PESEL. |
| 5 | | Unused | Unused (reads return 0) |
| 4 | MWIEN | Memory Write and Invalidate Enable | When set, DP83820 may use the Memory Write and Invalidate command for qualifying transfers. If 0, Memory Write will always be used instead of MWI. The DP83820 further qualifies enabling the MWI command using the MWI_DIS bit in the CFG operational register. |
| 3 | | Unused | Unused (reads return 0) |
| 2 | BMEN | Bus Master Enable | When set, DP83820 is allowed to act as a PCI bus master. When reset, DP83820 is prohibited from acting as a PCI bus master. |
| 1 | MSEN | Memory Space Access | When set, DP83820 responds to memory space accesses. When reset, DP83820 ignores memory space accesses. |
| 0 | IOSEN | IO Space Access | When set, DP83820 responds to IO space accesses. When reset, DP83820 ignores IO space accesses. |

4.0 Register Set (Continued)

4.1.3 Configuration Revision ID Register

This register stores the silicon revision number, revision number of software interface specification and lets the configuration software know that it is an Ethernet controller in the class of network controllers.

Tag: CFGRID *Size:* 32 bits *Hard Reset:* 02000000h
Offset: 08h *Access:* Read Only *Soft Reset:* Unchanged

| bit | tag | description | usage |
|-------|--------|------------------|---|
| 31-24 | BASECL | Base Class | Returns 02h which specifies a network controller. |
| 23-16 | SUBCL | Sub Class | Returns 00h which specifies an Ethernet controller. |
| 15-8 | PROGIF | Programming IF | Returns 00h which specifies the first release of the DP83820. |
| 7-0 | REVID | Silicon Revision | Returns 00h which specifies the silicon revision. |

4.1.4 Configuration Latency Timer Register

This register gives status and controls such miscellaneous functions as BIST, Latency timer and Cache line size.

Tag: CFGLAT *Size:* 32 bits *Hard Reset:* 00000000h
Offset: 0Ch *Access:* Read Write *Soft Reset:* Unchanged

| bit | tag | description | usage |
|-------|---------|-----------------|--|
| 31 | BISTCAP | BIST Capable | Reads will always return 0. |
| 30 | BISTEN | BIST Enable | Reads will return a 0, writes are ignored. |
| 29-16 | | Reserved | Reads will return a 0, writes are ignored. |
| 15-8 | LAT | Latency Timer | Set by software to the number of PCI clocks that DP83820 may hold the PCI bus. |
| 7-0 | CLS | Cache Line Size | Set to the value of the system cache line size in dwords. Acceptable values are powers of 2 less than or equal to 128. All other values will be recognized as 0. |

DP83820 Bus Master Operations:

Based on cache line size, the DP83820 will use the following PCI commands for bus mastered transfers:

- 0110 - Mem Read Single dword read transfers
- 1110 - Mem Read Line Read More than 1 dword but not across a cacheline boundary
- 1100 - Mem Read Multiple Read transfers that cross a cacheline boundary
- 0111 - Mem Write Writes that do not exactly overwrite 1 or more cachelines
- 1111 - Mem Write Invalidate Writes that exactly overwrite 1 or more cachelines

4.0 Register Set (Continued)

4.1.5 Configuration IO Base Address Register

This register specifies the Base I/O address which is required to build an address map during configuration. It also specifies the number of bytes required as well as an indication that it can be mapped into I/O space.

Tag: CFGIOA Size: 32 bits Hard Reset: 00000001h
 Offset: 10h Access: Read Write Soft Reset: Unchanged

| bit | tag | description | usage |
|------|--------|--------------------|--|
| 31-8 | IOBASE | Base IO Address | This is set by software to the base IO address for the Operational Register Map. |
| 7-2 | IOSIZE | Size indication | Read back as 0. This allows the PCI bridge to determine that the DP83820 requires 256 bytes of IO space. |
| 1 | | Unused | Unused (reads return 0). |
| 0 | IOIND | IO Space Indicator | Read Only. Set to 1 by DP83820 to indicate that DP83820 is capable of being mapped into IO space. |

4.1.6 Configuration Memory Address Register

This register specifies the Base Memory address which is required to build an address map during configuration. It also specifies the number of bytes required as well as an indication that it can be mapped into memory space.

Tag: CFGMA Size: 32 bits Hard Reset: 00000000h
 Offset: 14h Access: Read Write Soft Reset: unchanged

| bit | tag | description | usage |
|-------|---------|------------------------|--|
| 31-12 | MEMBASE | Memory Base Address | This is set by software to the base address for the Operational Register Map. |
| 11-4 | MEMSIZE | Memory Size | These bits return 0, which indicates that the DP83820 requires 4096 bytes of Memory Space (the minimum recommended allocation). |
| 3 | MEMPF | Prefetchable | Read Only. Set to 0 by DP83820. |
| 2-1 | MEMLOC | Location Selection | Read Only. Set to 10 by DP83820 if target 64-bit addressing is enabled. Set to 00 if 64-bit addressing is not enabled. 64-bit addressing capability is loaded from EEPROM at power-up and is reflected in the CFG:T64ADDR bit in operational register space. |
| 0 | MEMIND | Memory Space Indicator | Read Only. Set to 0 by DP83820 to indicate that DP83820 is capable of being mapped into memory space. |

4.1.7 Configuration Memory Address High Dword Register

This register specifies the upper 32-bits of the Base Memory address which is required to build an address map during configuration.

Tag: CFGMA1 Size: 32 bits Hard Reset: 00000000h
 Offset: 18h Access: Read Write Soft Reset: unchanged

| bit | tag | description | usage |
|------|----------|--------------------------|--|
| 31-0 | MEMBASE1 | Memory Base High Address | This is set by software to the upper 32-bits of the base address for the Operational Register Map. If 64-bit addressing is disabled then this field will be read-only and always return 0. |

4.0 Register Set (Continued)

4.1.8 Configuration Subsystem Identification Register

The CFGSID allows system software to distinguish between different subsystems based on the same PCI silicon. The values in this register can be loaded from the EEPROM if configuration is enabled.

Tag: CFGSID *Size:* 32 bits *Hard Reset:* ?
Offset: 2Ch *Access:* Read Only *Soft Reset:* unchanged

| bit | tag | description | usage |
|-------|--------|---------------------|------------------------|
| 31-16 | SDEVID | Subsystem Device ID | Loaded from the EEPROM |
| 15-0 | SVENID | Subsystem Vendor ID | Loaded from the EEPROM |

4.1.9 Boot ROM Configuration Register

Tag: CFGROM *Size:* 32 bits *Hard Reset:* 00000000h
Offset: 30h *Access:* Read Write *Soft Reset:* unchanged

| bit | tag | description | usage |
|-------|---------|------------------|---|
| 31-16 | ROMBASE | ROM Base Address | Set to the base address for the boot ROM. |
| 15-11 | ROMSIZE | ROM Size | Read only. Set to 0 indicating a requirement for 64K bytes of Boot ROM space |
| 10-1 | | Unused | unused (reads return 0) |
| 0 | ROMEN | ROM Enable | This is used by the PCI BIOS to enable accesses to boot ROM. This allows the DP83820 to share the address decode logic between the boot ROM and itself. The BIOS will copy the contents of the boot ROM to system RAM before executing it. Set to 1 enables the address decode for boot ROM disabling access to operational target registers. |

4.1.10 Capabilities Pointer Register

This register stores the capabilities linked list offset into the PCI configuration space.

Tag: CAPPTR *Size:* 32 bits *Hard Reset:* 00000040h
Offset: 34h *Access:* Read Only *Soft Reset:* unchanged

| bit | tag | description | usage |
|------|-------|--------------------------|--|
| 31-8 | | | unused (reads return 0) |
| 7-0 | CLOFS | Capabilities List Offset | Offset into PCI configuration space for the location of the first item in the Capabilities Linked List, set to 40h to point to the PMCAP register. |

4.0 Register Set (Continued)

4.1.11 Configuration Interrupt Select Register

This register stores the interrupt line number as identified by the POST software that is connected to the interrupt controller as well as DP83820 desired settings for maximum latency and minimum grant. Max latency and Min. latency can be loaded from the EEPROM

Tag: CFGINT Size: 32 bits Hard Reset: 340b0100h
 Offset: 3Ch Access: Read Write Soft Reset: unchanged

| bit | tag | description | usage |
|-------|-------|-----------------|---|
| 31-24 | MXLAT | Maximum Latency | The DP83820 desired setting for Max Latency. The DP83820 will initialize this field to 52d (13 usec). The value in this register can be loaded from the EEPROM. |
| 23-16 | MNGNT | Minimum Grant | The DP83820 desired setting for Minimum Grant. The DP83820 will initialize this field to 11d (2.75 usec). The value in this register can be loaded from the EEPROM. |
| 15-8 | IPIN | Interrupt Pin | Read Only, always return 0000 0001 (INTA) |
| 7-0 | ILINE | Interrupt Line | Set to which line on the interrupt controller that the DP83820's interrupt pin is connected to. |

4.1.12 Power Management Capabilities Register

This register provides information on the capabilities of the functions related to power management. This register also contains a pointer to the next item in the capabilities list and the capability ID for Power Management. This register is only visible if CFGCS[4] is set.

Tag: PMCAP Size: 32 bits Hard Reset: FF820001
 Offset: 40h Access: Read Only Soft Reset: unchanged

| bit | tag | description | usage | | | | | | | | |
|--------------|------------------------------|--|--|------------|----------------|--------------|------------------------------|-------|-------|-------|------------------|
| 31-27 | PMES | PME Support | This 5 bit field indicates the power states in which DP83820 may assert PMEN. A 1 indicates PMEN is enabled for that state, a 0 indicates PMEN is inhibited in that state. XXXX1 - PMEN can be asserted from state D0 XXX1X - PMEN can be asserted from state D1 XX1XX - PMEN can be asserted from state D2 X1XXX - PMEN can be asserted from state D3hot 1XXXX - PMEN can be asserted from state D3cold The DP83820 will only report PME support for D3cold if auxiliary power is detected on the 3VAUX pin, in addition this value can be loaded from the EEPROM when in the D3cold state. | | | | | | | | |
| 26 | D2S | D2 Support | This bit is set to a 1 when the DP83820 supports the D2 state. | | | | | | | | |
| 25 | D1S | D1 Support | This bit is set to a 1 when the DP83820 supports the D1 state. | | | | | | | | |
| 24-22 | AUX_CURRENT | 3 bit field for aux current requirement. | Aux_Current - This 3 bit field reports the 3.3Vaux auxiliary current requirements for the PCI function. If PMEN generation from D3cold is not supported by the function(PMCAP[31]), this field returns a value of "000b" when read. <table border="0"> <tr> <td>Bit</td> <td>3.3Vaux</td> </tr> <tr> <td>8 7 6</td> <td>Max. Current Required</td> </tr> <tr> <td>1 1 0</td> <td>320mA</td> </tr> <tr> <td>0 0 0</td> <td>0 (self powered)</td> </tr> </table> | Bit | 3.3Vaux | 8 7 6 | Max. Current Required | 1 1 0 | 320mA | 0 0 0 | 0 (self powered) |
| Bit | 3.3Vaux | | | | | | | | | | |
| 8 7 6 | Max. Current Required | | | | | | | | | | |
| 1 1 0 | 320mA | | | | | | | | | | |
| 0 0 0 | 0 (self powered) | | | | | | | | | | |
| 21 | DSI | Device Specific Initialization | This bit is set to 1 to indicate to the system that initialization of the DP83820 device is required (beyond the standard PCI configuration header) before the generic class device driver is able to use it. A 1 indicates that DP83820 requires a DSI sequence following transition to the D0 uninitialized state. This bit can be loaded from the EEPROM. | | | | | | | | |

4.0 Register Set (Continued)

| | | | |
|-------|--------|--------------------------|--|
| 20 | | Reserved | Reserved (reads return 0) |
| 19 | PMEC | PME Clock | Returns 0 to indicate PCI clock not needed for PMEN. |
| 18-16 | PMV | Power Management Version | This bit field indicates compliance to a specific PM specification rev level. Currently set to 010b. |
| 15-8 | NLIPTR | Next List Item Pointer | Offset into PCI configuration space for the location of the next item in the Capabilities Linked List. Returns 00h as no other capabilities are offered. |
| 7-0 | CAPID | Capability ID | Always returns 01h for Power Management ID. |

4.1.13 Power Management Control and Status Register

This register contains PM control and status information.

Tag: PMCSR *Size:* 32 bits *Hard Reset:* 00000000h
Offset: 44h *Access:* Read Write *Soft Reset:* unchanged

| bit | tag | description | usage |
|-------|--------|---------------------------|---|
| 31-24 | | | reserved (reads return 0) |
| 23-16 | BSE | Bridge Support Extensions | unused (reads return 0) |
| 15 | PMESTS | PME Status | Sticky bit which represents the state of the PME logic, regardless of the state of the PMEEN bit. |
| 14-9 | DSCALE | Data Scale | reserved (reads return 0) |
| 8 | PMEEN | PME Enable | When set to 1, this bit enables the assertion of the PME function on the PMEN pin. When 0, the PMEN pin is forced to be inactive. This value can be loaded from the EEPROM. |
| 7-2 | | Reserved | unused (reads return 0) |
| 1-0 | PSTATE | Power State | This 2 bit field is used both to determine the current power state of DP83820, and to set a new power state. 00 - D0 01 - D1 10 - D2 11 - D3hot |

4.0 Register Set (Continued)

4.2 Operational Registers

The DP83820 provides the following set of operational registers mapped into PCI memory space or I/O space. Writes to reserved register locations are ignored. Reads to reserved register locations return undefined values. When mapped to I/O space, a 256 byte window allows access to all the Operational Registers (00-FCh). When mapped into PCI memory space, a 4096 byte window is enabled. In addition to access to Operational Registers, the PCI Configuration Registers can be read at addresses 200-2FCh. Other addresses provide aliased access to Operational Registers or read-only PCI Configuration Registers.

Table 4-2 Operational Register Map

| offset | tag | description | access |
|--------|---------|---|--------|
| 00h | CR | Command Register | R/W |
| 04h | CFG | Configuration Register | R/W |
| 08h | MEAR | EEPROM Access Register | R/W |
| 0Ch | PTSCR | PCI Test Control Register | R/W |
| 10h | ISR | Interrupt Status Register | RO |
| 14h | IMR | Interrupt Mask Register | R/W |
| 18h | IER | Interrupt Enable Register | R/W |
| 1Ch | IHR | Interrupt Holdoff Register | R/W |
| 20h | TXDP | Transmit Descriptor Pointer Register | R/W |
| 24h | TXDP_HI | Transmit Descriptor Pointer High Dword Register | R/W |
| 28h | TXCFG | Transmit Configuration Register | R/W |
| 2Ch | GPOR | General Purpose I/O Control Register | R/W |
| 30h | RXDP | Receive Descriptor Pointer Register | R/W |
| 34h | RXDP_HI | Receive Descriptor Pointer High Dword Register | R/W |
| 38h | RXCFG | Receive Configuration Register | R/W |
| 3Ch | PQCR | Priority Queueing Control Register | R/W |
| 40h | WCSR | Wake on LAN Control/Status Register | R/W |
| 44h | PCR | Pause Control/Status Register | R/W |
| 48h | RFCR | Receive Filter/Match Control Register | R/W |
| 4Ch | RFDR | Receive Filter/Match Data Register | R/W |
| 50h | BRAR | Boot ROM Address | R/W |
| 54h | BRDR | Boot ROM Data | R/W |
| 58h | SRR | Silicon Revision Register | RO |
| 5Ch | MIBC | Management Information Base Control Register | R/W |
| 60-88h | MIB | Management Information Base Data Registers | RO |
| 8C-9Ch | | Reserved | |
| A0h | TXDP1 | Transmit Descriptor Pointer Priority 1 Register | R/W |
| A4h | TXDP2 | Transmit Descriptor Pointer Priority 2 Register | R/W |
| A8h | TXDP3 | Transmit Descriptor Pointer Priority 3 Register | R/W |
| ACh | | Reserved | |
| B0h | RXDP1 | Receive Descriptor Pointer Priority 1 Register | R/W |
| B4h | RXDP2 | Receive Descriptor Pointer Priority 2 Register | R/W |
| B8h | RXDP3 | Receive Descriptor Pointer Priority 3 Register | R/W |
| BCh | VRCR | VLAN/IP Receive Control Register | R/W |
| C0h | VTCR | VLAN/IP Transmit Control Register | R/W |
| C4h | VDR | VLAN Data register | R/W |
| C8 | | Reserved | |

4.0 Register Set (Continued)

| | | | |
|----|-----------|-------------------------------------|--|
| 16 | MRM_DIS | Memory Read Multiple Disable | This bit can be used to prevent the DP83820 from using the Memory Read Multiple and Memory Read Line commands. This bit is loaded from EEPROM at power-up. R/W |
| 15 | MWI_DIS | Memory Write and Invalidate Disable | This bit can be used to prevent the DP83820 from using the MWI command. This allows additional control for driver software which may not have access to the MWIEN bit in Configuration space. This bit is loaded from EEPROM at power-up. R/W |
| 14 | T64ADDR | Target 64-bit Addressing Enable | This read-only bit indicates the device will accept 64-bit addressing as a target. This bit is loaded from EEPROM at power-up. RO |
| 13 | PCI64_DET | PCI 64-bit Bus Detected | This status bit indicates the PCI bus was detected to be 64-bit at reset time. RO |
| 12 | DATA64_EN | 64-bit Data Enable | Software can use this bit to enable 64-bit data transfers by the Transmit and Receive DMA engines. If 0, all bus master transfers will be 32-bit. This bit is loaded from EEPROM at power-up. This bit should be cleared by software if the PCI bus was not detected as 64-bit capable (PCI_64_DET = 0). R/W |
| 11 | M64ADDR | Master 64-bit Addressing Enable | Software can set this bit to enable the DMA controllers to use 64-bit addressing. When set, the link and bufptr fields in the Descriptors are assumed to be 64-bit fields. This bit does not affect the device operation as a target. This bit is loaded from EEPROM at power-up. R/W |
| 10 | PHY_RST | Reset Phy | Asserts reset to phy using the PHYRST_N pin. R/W |
| 9 | PHY_DIS | Disable Phy | Setting this bit can be used to disable an external phy by deasserting the RXEN pin. This can be used to cause a phy to tri-state its RX MII/GMII pins. R/W |
| 8 | EXTSTS_EN | Extended Status Enable | When set, the Extended Status field is enabled for Transmit and Receive Descriptors. This field contains data for supporting the VLAN and IP Checksum processing features. R/W |
| 7 | REQALG | PCI Bus Request Algorithm | Selects mode for making requests for the PCI bus. When set to 0 (default), DP83820 will use an aggressive Request scheme. When set to a 1 DP83820 will use a more conservative scheme. R/W |
| 6 | SB | Single Back-off | Setting this bit to 1, forces the transmitter back-off state machine to always back-off for a single 802.3 interframe gap time, instead of following the 802.3 random back-off algorithm. A 0 (default) allows normal transmitter back-off operation. R/W |
| 5 | POW | Program Out of Window Timer | This bit controls when the <i>Out of Window</i> collision timer begins counting its 512 bit slot time. A 0 causes the timer to start after the SFD is received. A 1 causes the timer to start after the first bit of the preamble is received. R/W |
| 4 | EXD | Excessive Deferral Abort | Setting this bit to 1 will cause the transmitter to abort transmission on an excessive deferral. R/W |
| 3 | PESEL | Parity Error Detection Action | This bit controls the assertion of SERR when a data parity error is detected while the DP83820 is acting as the bus master. When set, parity errors will not result in the assertion of SERR. When reset, parity errors will result in the assertion of SERR, indicating a system error. This bit should be set to a 1 by software if the driver can handle recovery from and reporting of data parity errors. R/W |
| 2 | BROM_DIS | Disable Boot ROM interface | When set to 1, this bit inhibits the operation of the Boot ROM interface logic. R/W |
| 1 | EXT_125 | External 125MHz reference Select | When set to a 1, the 125MHz transmit clock for 1000 Mb/s mode is sourced from the REF125 pin. When set to a 0, the clock is sourced by the internal clock generator. This bit is loaded from EEPROM at power-up. R/W |
| 0 | BEM | Big Endian Mode | When set, DP83820 will perform bus-mastered data transfers in "big endian" mode. Note that access to register space is unaffected by the setting of this bit. R/W |

4.0 Register Set (Continued)

4.2.3 MII/EEPROM Access Register

The MII/EEPROM Access Register provides an interface for software access to the serial management port of an external MII device or NMC9306 style EEPROM. The default values given assume that the MDIO and EEDO lines have pullup resistors to VDD.

Tag: MEAR Size: 32 bits Hard Reset: 00000012h
Offset: 0008h Access: Read Write Soft Reset: 00000012h

| bit | tag | description | usage |
|------|-------|--------------------------|--|
| 31-7 | | | unused |
| 6 | MDC | MII Management Clock | Read-Write. Controls the value of the MDC pin. When set, the MDC pin is 1, when clear the MDC pin is 0. |
| 5 | MDDIR | MII Management Direction | Read-Write. Controls the direction of MDIO pin. When set, MacPhyter3V drives the current state of the MDIO bit onto the MDIO pin. When clear, the MDIO bit reflects the current state of the MDIO pin. |
| 4 | MDIO | MII Management Data | Read-Write. Provides software access to the MDIO pin (see MDDIR). |
| 3 | EESEL | EEPROM Chip Select | Controls the value of the EESEL pin. When set, the EESEL pin is 1; when clear the EESEL pin is 0. R/W |
| 2 | EECLK | EEPROM Serial Clock | Controls the value of the EECLK pin. When set, the EECLK pin is 1; when clear the EECLK pin is 0. R/W |
| 1 | EEDO | EEPROM Data Out | Returns the current state of the EEDO pin. When set, the EEDO pin is 1; when clear the EEDO pin is 0. RO |
| 0 | EEDI | EEPROM Data In | Controls the value of the EEDI pin. R/W |

4.0 Register Set (Continued)

4.2.4 EEPROM Map

| EEPROM Address | Configuration/Operation Register Bits | Default Value (16 bits) |
|----------------|---|-------------------------|
| 0000h | CFGSID[31:16] | 0000h |
| 0001h | CFGSID[15:0] | 0000h |
| 0002h | CFGINT[31:16] | 340Bh |
| 0003h | 4'h0, 2'h0, CFGCS[21], CFGCS[20], 2'h0, PMCAP[31], PMCAP[21], 3'h0, PMCSR[8] | 0220h |
| 0004h | 6'h00, GPIOR[9:0] | 0000h |
| 0005h | 6'h00, CFG[24:21], CFG[16:14], CFG[12:11], CFG[1] | 0000h |
| 0006h | CR[2], WCSR[10:9], WCSR[4:0], 1'h0, RFCR[31:27], RFCR[22], RFCR[19] | 0000h |
| 0007h | SOPAS[47:32] | 0000h |
| 0008h | SOPAS[31:16] | 0000h |
| 0009h | SOPAS[15:0] | 0000h |
| 000Ah | PMATCH[47:32] | 0000h |
| 000Bh | PMATCH[31:16] | 0000h |
| 000Ch | PMATCH[15:0] | 0000h |
| 000Dh | checksum value | N/A |

Registers for SOPAS[47:0] and PMATCH[47:0] can be accessed directly via the combination of the RFCR (offset 0048h) and RFDR (offset 004Ch) registers as well as loaded from the EEPROM as noted here.

The lower 8 bits of the checksum value should be 55h. For the upper 8 bits, add the top 8 data bits to the lower 8 data bits for each address. Sum the resultant 8 bit values for all addresses and then add 55h. Take the 2's complement of the final sum. This 2's complement number should be the upper 8 bits of the checksum value in the last address.

As an example, consider an EEPROM with two addresses. EEPROM address 0000h contains the data 1234h. EEPROM address 0001h contains the data 5678h.

$$12h + 34h = 46h$$

$$56h + 78h = CEh$$

$$46h + CEh + 55h = 69h$$

The 2's complement of 69h is 97h so the checksum value entered into EEPROM address 0002h would be 9755h.

4.0 Register Set (Continued)

4.2.5 PCI Test Control Register

Tag: PTSCR

Size: 32 bits

Hard Reset: 00000000h

Offset: 000Ch

Access: Read Write

Soft Reset: 00000000h

| bit | tag | description | usage |
|-------|---------------|--------------------------|--|
| 31-16 | | Reserved | Reserved |
| 15 | | Reserved | Reserved. Must be written as a 0. |
| 14 | | Reserved | Reserved |
| 13 | RBIST_RST | SRAM BIST Reset | Setting this bit to 1 allows the SRAM BIST engine to be reset. R/W |
| 12-11 | | Reserved | Reserved |
| 10 | RBIST_EN | SRAM BIST Enable | Setting this bit to 1 starts the SRAM BIST engine. R/W |
| 9 | RBIST_DONE | SRAM BIST Done | This bit is set to 1 when the SRAM BIST completes each section. RO |
| 8 | RBIST_RX1FAIL | RX Status FIFO BIST Fail | This bit is set to 1 if the SRAM BIST detects a failure in RX Status FIFO SRAM. This bit is cleared only by resetting the BIST. RO |
| 7 | RBIST_RX0FAIL | RX Data FIFO BIST Fail | This bit is set to 1 if the SRAM BIST detects a failure in RX Data FIFO SRAM. This bit is cleared only by resetting the BIST. RO |
| 6 | | Reserved | Reserved |
| 5 | RBIST_TX0FAIL | TX Data FIFO BIST Fail | This bit is set to 1 if the SRAM BIST detects a failure in TX Data FIFO SRAM. This bit is cleared only by resetting the BIST. RO |
| 4 | RBIST_HFFAIL | Hash Filter BIST Fail | This bit is set to 1 if the SRAM BIST detects a failure in the hash filter SRAM. This bit is cleared only by resetting the BIST. RO |
| 3 | RBIST_RXFAIL | RX Filter BIST Fail | This bit is set to 1 if the SRAM BIST detects a failure in the RX filter SRAM. This bit is cleared only by resetting the BIST. RO |
| 2 | EELOAD_EN | Enable EEPROM Load | This bit is set to a 1 to manually initiate a load of configuration information from EEPROM. A 1 is returned while the configuration load from EEPROM is active. R/W |
| 1 | EEBIST_EN | Enable EEPROM BIST | This bit is set to a 1 to initiate EEPROM BIST, which verifies the EEPROM data and checksum without reloading configuration values to the device. A 1 is returned while the EEPROM BIST is active. R/W |
| 0 | EEBIST_FAIL | EE BIST Fail indication | This bit is set to a 1 upon completion of the EEPROM BIST (EEBIST_EN returns 0) if the BIST logic encountered an invalid checksum. RO |

4.2.6 Interrupt Status Register

This register indicates the source of an interrupt when the INTA pin goes active. Enabling the corresponding bits in the Interrupt Mask Register (IMR) allows bits in this register to produce an interrupt. When an interrupt is active, one or more bits in this register are set to a "1". The Interrupt Status Register reflects all current pending interrupts, regardless of the state of the corresponding mask bit in the IMR. Reading the ISR clears all interrupts. Writing to the ISR has no effect.

Tag: ISR

Size: 32 bits

Hard Reset: 00608000h

Offset: 0010h

Access: Read Only

Soft Reset: 00608000h

| bit | tag | description | usage |
|-----|---------|------------------------------------|--|
| 31 | | Reserved | Reserved |
| 30 | TXDESC3 | Tx Descriptor for Priority Queue 3 | This event is signaled after a transmit descriptor with the INTR bit set in the CMDSTS field has been updated. |
| 29 | TXDESC2 | Tx Descriptor for Priority Queue 2 | This event is signaled after a transmit descriptor with the INTR bit set in the CMDSTS field has been updated. |
| 28 | TXDESC1 | Tx Descriptor for Priority Queue 1 | This event is signaled after a transmit descriptor with the INTR bit set in the CMDSTS field has been updated. |
| 27 | TXDESC0 | Tx Descriptor for Priority Queue 0 | This event is signaled after a transmit descriptor with the INTR bit set in the CMDSTS field has been updated. |

4.0 Register Set (Continued)

| | | | |
|----|---------|------------------------------------|--|
| 26 | RXDESC3 | Rx Descriptor for Priority Queue 3 | This event is signaled after a receive descriptor with the INTR bit set in the CMDSTS field has been updated. |
| 25 | RXDESC2 | Rx Descriptor for Priority Queue 2 | This event is signaled after a receive descriptor with the INTR bit set in the CMDSTS field has been updated. |
| 24 | RXDESC1 | Rx Descriptor for Priority Queue 1 | This event is signaled after a receive descriptor with the INTR bit set in the CMDSTS field has been updated. |
| 23 | RXDESC0 | Rx Descriptor for Priority Queue 0 | This event is signaled after a receive descriptor with the INTR bit set in the CMDSTS field has been updated. |
| 22 | TXRCMP | Transmit Reset Complete | Indicates that a requested transmit reset operation is complete. |
| 21 | RXRCMP | Receive Reset Complete | Indicates that a requested receive reset operation is complete. |
| 20 | DPERR | Detected Parity Error | This bit is set whenever CFGCS:DPERR is set, but cleared (like all other ISR bits) when the ISR register is read. |
| 19 | SSERR | Signaled System Error | The DP83820 signaled a system error on the PCI bus. |
| 18 | RMABT | Received Master Abort | The DP83820 received a master abort generated as a result of target not responding. |
| 17 | RTABT | Received Target Abort | The DP83820 received a target abort on the PCI bus. |
| 16 | RXSOVR | Rx Status FIFO Overrun | Set when an overrun condition occurs on the Rx Status FIFO. |
| 15 | HIBINT | High Bits Interrupt Set | A logical OR of bits 22-16 |
| 14 | PHY | Phy interrupt | Set to 1 when interrupt is generated due to change in phy status. |
| 13 | PME | Power Management Event | Set when WOL conditioned detected |
| 12 | SWI | Software Interrupt | Set whenever the SWI bit in the CR register is set. |
| 11 | MIB | MIB Service | Set when one of the enabled management statistics has reached its interrupt threshold. |
| 10 | TXURN | Tx Underrun | Set when a transmit data FIFO underrun condition occurs. |
| 9 | TXIDLE | Tx Idle | This event is signaled when the transmit state machine enters the idle state from a non-idle state. This will happen whenever the state machine encounters an "end-of-list" condition (NULL link field or a descriptor with OWN clear). |
| 8 | TXERR | Tx Packet Error | This event is signaled after the last transmit descriptor in a failed transmission attempt has been updated with valid status. |
| 7 | TXDESC | Tx Descriptor | This event is signaled after a transmit descriptor with the INTR bit set in the CMDSTS field has been updated. If priority queueing is enabled, this bit will be set when any of the TXDESC0-3 bits are set. |
| 6 | TXOK | Tx Packet OK | This event is signaled after the last transmit descriptor in a successful transmission attempt has been updated with valid status |
| 5 | RXORN | Rx Overrun | Set when a receive data FIFO overrun condition occurs. |
| 4 | RXIDLE | Rx Idle | This event is signaled when the receive state machine enters the idle state from a running state. This will happen whenever the state machine encounters an "end-of-list" condition (NULL link field or a descriptor with OWN set). |
| 3 | RXEARLY | Rx Early Threshold | Indicates that the initial Rx Drain Threshold has been met by the incoming packet, and the transfer of the number of bytes specified by the DRTH field in the RXCFG register has been completed by the receive DMA engine. This interrupt condition will occur only once per packet. |
| 2 | RXERR | Rx Packet Error | This event is signaled after the last receive descriptor in a failed packet reception has been updated with valid status. |
| 1 | RXDESC | Rx Descriptor | This event is signaled after a receive descriptor with the INTR bit set in the CMDSTS field has been updated. If priority queueing is enabled, this bit will be set when any of the RXDESC0-3 bits are set. |
| 0 | RXOK | Rx OK | Set by the receive state machine following the update of the last receive descriptor in a good packet. |

4.0 Register Set (Continued)

4.2.7 Interrupt Mask Register

This register masks the interrupts that can be generated from the ISR. Writing a “1” to the bit enables the corresponding interrupt. During a hardware reset, all mask bits are cleared. Setting a mask bit allows the corresponding bit in the ISR to cause an interrupt. ISR bits are always set to 1, however, if the condition is present, regardless of the state of the corresponding mask bit.

Tag: IMR
Offset: 0014h

Size: 32 bits
Access: Read Write

Hard Reset: 00000000h
Soft Reset: 00000000h

| bit | tag | description | usage |
|-----|---------|------------------------------------|---|
| 31 | | | unused |
| 30 | TXDESC3 | Tx Descriptor for Priority Queue 3 | When this bit is 3, the corresponding bit in the ISR will not cause an interrupt. |
| 29 | TXDESC2 | Tx Descriptor for Priority Queue 2 | When this bit is 2, the corresponding bit in the ISR will not cause an interrupt. |
| 28 | TXDESC1 | Tx Descriptor for Priority Queue 1 | When this bit is 1, the corresponding bit in the ISR will not cause an interrupt. |
| 27 | TXDESC0 | Tx Descriptor for Priority Queue 0 | When this bit is 0, the corresponding bit in the ISR will not cause an interrupt. |
| 26 | RXDESC3 | Rx Descriptor for Priority Queue 3 | When this bit is 3, the corresponding bit in the ISR will not cause an interrupt. |
| 25 | RXDESC2 | Rx Descriptor for Priority Queue 2 | When this bit is 2, the corresponding bit in the ISR will not cause an interrupt. |
| 24 | RXDESC1 | Rx Descriptor for Priority Queue 1 | When this bit is 1, the corresponding bit in the ISR will not cause an interrupt. |
| 23 | RXDESC0 | Rx Descriptor for Priority Queue 0 | When this bit is 0, the corresponding bit in the ISR will not cause an interrupt. |
| 22 | TXRCMP | Transmit Reset Complete | When this bit is 0, the corresponding bit in the ISR will not cause an interrupt. |
| 21 | RXRCMP | Receive Reset Complete | When this bit is 0, the corresponding bit in the ISR will not cause an interrupt. |
| 20 | DPERR | Detected Parity Error | When this bit is 0, the corresponding bit in the ISR will not cause an interrupt. |
| 19 | SSERR | Signaled System Error | When this bit is 0, the corresponding bit in the ISR will not cause an interrupt. |
| 18 | RMABT | Received Master Abort | When this bit is 0, the corresponding bit in the ISR will not cause an interrupt. |
| 17 | RTABT | Received Target Abort | When this bit is 0, the corresponding bit in the ISR will not cause an interrupt. |
| 16 | RXSOVR | Rx Status FIFO Overrun | When this bit is 0, the corresponding bit in the ISR will not cause an interrupt. |
| 15 | HIBINT | High Bits Interrupt | When this bit is 0, the corresponding bit in the ISR will not cause an interrupt. |
| 14 | PHY | Phy interrupt | When this bit is 0, the corresponding bit in the ISR will not cause an interrupt. |
| 13 | PME | Power Management Event | When this bit is 0, the corresponding bit in the ISR will not cause an interrupt. |
| 12 | SWI | Software Interrupt | When this bit is 0, the corresponding bit in the ISR will not cause an interrupt. |
| 11 | MIB | MIB Service | When this bit is 0, the corresponding bit in the ISR will not cause an interrupt. |
| 10 | TXURN | Tx Underrun | When this bit is 0, the corresponding bit in the ISR will not cause an interrupt. |
| 9 | TXIDLE | Tx Idle | When this bit is 0, the corresponding bit in the ISR will not cause an interrupt. |
| 8 | TXERR | Tx Packet Error | When this bit is 0, the corresponding bit in the ISR will not cause an interrupt. |
| 7 | TXDESC | Tx Descriptor | When this bit is 0, the corresponding bit in the ISR will not cause an interrupt. |
| 6 | TXOK | Tx Packet OK | When this bit is 0, the corresponding bit in the ISR will not cause an interrupt. |
| 5 | RXORN | Rx Overrun | When this bit is 0, the corresponding bit in the ISR will not cause an interrupt. |
| 4 | RXIDLE | Rx Idle | When this bit is 0, the corresponding bit in the ISR will not cause an interrupt. |
| 3 | RXEARLY | Rx Early Threshold | When this bit is 0, the corresponding bit in the ISR will not cause an interrupt. |
| 2 | RXERR | Rx Packet Error | When this bit is 0, the corresponding bit in the ISR will not cause an interrupt. |
| 1 | RXDESC | Rx Descriptor | When this bit is 0, the corresponding bit in the ISR will not cause an interrupt. |
| 0 | RXOK | Rx OK | When this bit is 0, the corresponding bit in the ISR will not cause an interrupt. |

4.0 Register Set (Continued)

4.2.11 Transmit Descriptor Pointer High Dword Register

This register points to the upper 32-bits of the current Transmit Descriptor for 64-bit addressing. If Transmit Priority Queueing is enabled, this becomes the Descriptor pointer for all priority queues.

Tag: TXDP_HI Size: 32 bits Hard Reset: 00000000h
 Offset: 0024h Access: Read Write Soft Reset: 00000000h

| bit | tag | description | usage |
|------|---------|--|---|
| 31-0 | TXDP_HI | Transmit Descriptor Pointer High Dword | If 64-bit addressing is enabled, this will be used as the upper 32-bits of the current transmit descriptor pointer. |

4.2.12 Transmit Configuration Register

This register defines the Transmit Configuration for DP83820. It controls such functions as Loopback, Heartbeat, Auto Transmit Padding, programmable Interframe Gap, Fill & Drain Thresholds, and maximum DMA burst size.

Tag: TXCFG Size: 32 bits Hard Reset: 00000120
 Offset: 0028h Access: Read Write Soft Reset: 00000120

| bit | tag | description | usage |
|-------|---------|----------------------------------|--|
| 31 | CSI | Carrier Sense Ignore | Setting this bit to 1 causes the transmitter to ignore carrier sense activity, which inhibits reporting of CRS status to the transmit status register, and inhibits logging of TXCSErrors in the MIB counter block. When this bit is 0 (default), the transmitter will monitor the CRS signal during transmission and reflect valid status in the transmit status register and MIB counter block. This bit must be set to enable full-duplex operation. |
| 30 | HBI | HeartBeat Ignore | Setting this bit to 1 causes the transmitter to ignore the heartbeat (CD) pulse which follows the packet transmission and inhibits logging of TXSQEErrors in the MIB counter block. When this bit is set to 0 (default), the transmitter will monitor the heartbeat pulse and log TXSQEErrors to the MIB counter block. This bit must be set to enable full-duplex operation |
| 29 | MLB | MAC Loopback | Setting this bit to a 1 places the DP83820 MAC into a loopback state which routes all transmit traffic to the receiver, and disables the transmit and receive interfaces of the MII. A 0 in this bit allows normal MAC operation. The transmitter and receiver must be disabled before enabling the loopback mode. (Packets received during MLB mode will reflect loopback status in the receive descriptor's cmdsts.LBP field.) |
| 28 | ATP | Automatic Transmit Padding | Setting this bit to 1 causes the MAC to automatically pad small (runt) transmit packets to the Ethernet minimum size of 64 bytes. This allows driver software to transfer only actual packet data. Setting this bit to 0 disables the automatic padding function, forcing software to control runt padding. |
| 27-24 | | | unused |
| 23 | ECRETRY | Excessive Collision Retry Enable | This bit enables automatic retries of excessive collisions. If set, the transmitter will retry the packet up to 4 excessive collision counts, for a total of 64 attempts. If the packet still does not complete successfully, then the transmission will be aborted after the 64th attempt. If this bit is not set, then the transmission will be aborted after the 16th attempt. Note that setting this bit will change how collisions are reported in the status field of the transmit descriptor. |

4.0 Register Set (Continued)

| | | | |
|-------|----------|-------------------------------------|--|
| 22-20 | MXDMA | Max DMA Burst Size per Tx DMA Burst | This field sets the maximum size of transmit DMA data bursts according to the following table: 000 = 256 32-bit words (1024 bytes) 001 = 2 32-bit words (8 bytes) 010 = 4 32-bit words (16 bytes) 011 = 8 32-bit words (32 bytes) 100 = 16 32-bit words (64 bytes) 101 = 32 32-bit words (128 bytes) 110 = 64 32-bit words (256 bytes) 111 = 128 32-bit word (512 bytes) |
| 19 | BRST_DIS | 1000 Mb/s Burst Disable | This bit can disable transmit bursting for 1000 Mb/s half-duplex operation. The bit will have no affect 10/100 Mb/s or full-duplex modes. |
| 18-16 | | | unused |
| 15-8 | FLTH | Tx Fill Threshold | Specifies the fill threshold in units of 32 bytes. When the number of available bytes in the transmit FIFO reaches this level, the transmit bus master state machine will be allowed to request the PCI bus for transmit packet fragment reads. A value of 0 in this field will produce unexpected results and must not be used. |
| 7-0 | DRTH | Tx Drain Threshold | Specifies the drain threshold in units of 32 bytes. When the number of bytes in the FIFO reaches this level (or the FIFO contains at least one complete packet) the MAC transmit state machine will begin the transmission of a packet. NOTE: In order to prevent a deadlock condition from occurring, the transmit drain threshold should never be set higher than the (txFIFOsize - TXCFG:FLTH). A value of 0 in this field will prevent draining of the packet until the complete packet has been loaded into the FIFO. |

4.2.13 General Purpose I/O Control Register

This register allows configuration of the General Purpose I/O pins. Note that these pins are especially useful when interfacing to a Ten-Bit Interface Phy Device.

Tag: GPIOR Size: 32 bits Hard Reset: 00000000h
Offset: 002Ch Access: Read Write Soft Reset: 00000000h

| bit | tag | description | usage |
|-------|--------|-------------------------------------|--|
| 31-15 | | | unused |
| 14 | GP5_IN | General Purpose Pin 5 Input Value | Input value from the GP5 pin. When GP5_OE is a 1, this should reflect the value of GP5_OUT. RO |
| 13 | GP4_IN | General Purpose Pin 4 Input Value | Input value from the GP4 pin. When GP4_OE is a 1, this should reflect the value of GP4_OUT. RO |
| 12 | GP3_IN | General Purpose Pin 3 Input Value | Input value from the GP3 pin. When GP3_OE is a 1, this should reflect the value of GP3_OUT. RO |
| 11 | GP2_IN | General Purpose Pin 2 Input Value | Input value from the GP2 pin. When GP2_OE is a 1, this should reflect the value of GP2_OUT. RO |
| 10 | GP1_IN | General Purpose Pin 1 Input Value | Input value from the GP1 pin. When GP1_OE is a 1, this should reflect the value of GP1_OUT. RO |
| 9 | GP5_OE | General Purpose Pin 5 Output Enable | Enables the GP5 pin for use as an output. This bit is loaded from EEPROM at power-up. R/W |
| 8 | GP4_OE | General Purpose Pin 4 Output Enable | Enables the GP4 pin for use as an output. This bit is loaded from EEPROM at power-up. R/W |
| 7 | GP3_OE | General Purpose Pin 3 Output Enable | Enables the GP3 pin for use as an output. This bit is loaded from EEPROM at power-up. R/W |
| 6 | GP2_OE | General Purpose Pin 2 Output Enable | Enables the GP2 pin for use as an output. This bit is loaded from EEPROM at power-up. R/W |

4.0 Register Set (Continued)

4.2.16 Receive Configuration Register

This register is used to set the receive configuration for DP83820. Receive properties such as accepting error packets, runt packets, setting the receive drain threshold etc. are controlled here

Tag: RXCFG Size: 32 bits Hard Reset: 0000004h
 Offset: 0038h Access: Read Write Soft Reset: 0000004h

| bit | tag | description | usage |
|-------|----------|--|--|
| 31 | AEP | Accept Errored Packets | When set to 1, all packets with CRC or alignment errors will be accepted. When set to 0, all packets with CRC or alignment errors will be rejected if possible. Note that depending on the type of error, some packets may be received with errors, regardless of the setting of AEP. These errors will be indicated in the CMDSTS field of the last descriptor in the packet. |
| 30 | ARP | Accept Runt Packets | When set to 1, all packets under 64 bytes in length without errors are accepted. When this bit is 0, all packets less than 64 bytes in length will be rejected if possible. |
| 29 | STRIPCRC | Strip CRC | When set to a 1, the CRC will be stripped from the receive packet and the byte count adjusted appropriately. |
| 28 | RX_FD | Receive Full Duplex | When set to 1, data received simultaneously to a local transmission (such as during a PMD loopback or full duplex operation) will be accepted as valid received data. When set to 0 (default), all data received simultaneous to a local transmit will be rejected. This bit must be set to 1 for PMD loopback and full duplex operation. |
| 27 | ALP | Accept Long Packets | When set to 1, all packets > 1518 bytes in length and <= 65527 bytes will be treated as normal receive packets, and will not be tagged as long or error packets. All packets > 65527 bytes in length will be truncated at 65528 bytes and either rejected from the FIFO, or tagged as long packets. Care must be taken when accepting long packets to ensure that buffers provided are of adequate length. When ALP is set to 0, packets larger than 1518 bytes (CRC inclusive) will be truncated at 1514 bytes, and rejected if possible. |
| 26 | AIRL | Accept In-Range Length Errored Packets | When set to 1, packets with Length/Type fields that do not match the data length of the packet will be accepted. When set to 0, packets with Length/Type fields that do not match the data length of the packet will be rejected. In-Range Length checking only occurs if the Length/Type field is a valid length. |
| 25-23 | | | unused |
| 22-20 | MXDMA | Max DMA Burst Size per Rx DMA Burst | This field sets the maximum size of receive DMA data bursts according to the following table: 000 = 256 32-bit words (1024 bytes) 001 = 2 32-bit words (8 bytes) 010 = 4 32-bit words (16 bytes) 011 = 8 32-bit words (32 bytes) 100 = 16 32-bit words (64 bytes) 101 = 32 32-bit words (128 bytes) 110 = 64 32-bit words (256 bytes) 111 = 128 32-bit word (512 bytes) |
| 19-6 | | | unused |

4.0 Register Set (Continued)

| | | | |
|-------|-----------|---------------------------|--|
| 26 | | | unused - returns 0 |
| 25-24 | PS_STHI | RX Stat FIFO Hi Threshold | <p>Status FIFO Threshold for initiating a pause frame with a length field of 0. This allows termination of an active pause interval when the status FIFO has enough space available. The following encoding determines when a length 0 pause frame will be sent:</p> <p>00: Disabled 01: 2 or more packets available 10: 4 or more packets available 11: 8 or more packets available</p> <p>This value, if enabled, should always be equal to or greater than the low threshold (PS_STLO). When disabling the high threshold, the PS_FFHI field should also be set to disabled.</p> |
| 23-22 | PS_STLO | RX Stat FIFO Lo Threshold | <p>Status FIFO Threshold for initiating a pause frame. Upon reception of a valid packet, a pause frame will be transmitted if space remaining in the status FIFO is less than the threshold value. The encoding is as follows:</p> <p>00: Disabled 01: Less than 2 packets available 10: Less than 4 packets available 11: Less than 8 packets available</p> |
| 21-20 | PS_FFHI | RX Data FIFO Hi Threshold | <p>Data FIFO Threshold for initiating a pause frame with a length field of 0. This allows termination of an active pause interval when the data FIFO has enough space available. The following encoding determines when a length 0 pause frame will be sent:</p> <p>00: Disabled 01: 2K or more bytes available 10: 4K or more bytes available 11: 8K or more bytes available</p> <p>This value, if enabled, should always be equal to or greater than the low threshold (PS_FFLO). When disabling the high threshold, the PS_STHI field should also be set to disabled.</p> |
| 19-18 | PS_FFLO | RX Data FIFO Lo Threshold | <p>Data FIFO Threshold for initiating a pause frame. Upon reception of a valid packet, a pause frame will be transmitted if space remaining in the data FIFO is less than the threshold value. The encoding is as follows:</p> <p>00: Disabled 01: Less than 2K bytes available 10: Less than 4K bytes available 11: Less than 8K bytes available</p> |
| 17 | PS_TX | Transmit Pause Frame | This is a manual method of sending a pause frame. This bit will remain set until the pause frame is transmitted. R/W |
| 16 | | Reserved | Reserved. R/W |
| 15-0 | PAUSE_CNT | Pause Counter Value | Pause Length field which will be sent in a Transmit Pause frame. R/W |

4.0 Register Set (Continued)

4.2.20 Receive Filter/Match Control Register

The RFCR register is used to control and configure the DP83820 Receive Filter Control logic. The Receive Filter Control Logic is used to configure destination address filtering of incoming packets.

Tag: RFCR

Size: 32 bits

Hard Reset: 00000000h

Offset: 0048h

Access: Read Write

Soft Reset: 00000000h

| bit | tag | description | usage |
|-------|------|-------------------------|---|
| 31 | RFEN | Rx Filter Enable | When this bit is set to 1, the Rx Filter is enabled to qualify incoming packets. When set to a 0, receive packet filtering is disabled (i.e. all receive packets are rejected). This bit must be 0 for the other bits in this register to be configured. |
| 30 | AAB | Accept All Broadcast | When set to a 1, this bit causes all broadcast address packets to be accepted. When set to 0, no broadcast address packets will be accepted. |
| 29 | AAM | Accept All Multicast | When set to a 1, this bit causes all multicast address packets to be accepted. When set to 0, multicast destination addresses must have the appropriate bit set in the multicast hash table mask in order for the packet to be accepted. |
| 28 | AAU | Accept All Unicast | When set to a 1, this bit causes all unicast address packets to be accepted. When set to 0, the destination address must match the node address value specified through some other means in order for the packet to be accepted. |
| 27 | APM | Accept on Perfect Match | When set to 1, this bit allows the perfect match register to be used to compare against the DA for packet acceptance. When this bit is 0, the perfect match register contents will not be used for DA comparison. |
| 26-23 | APAT | Accept on Pattern Match | When one or more of these bits is set to 1, a packet will be accepted if the first n bytes (n is the value defined in the associated pattern count register) match the associated pattern buffer memory contents. When a bit is set to 0, the associated pattern buffer will not be used for packet acceptance. |
| 22 | AARP | Accept ARP Packets | When set to 1, this bit allows all ARP packets (packets with a TYPE/LEN field set to 0806h) to be accepted, regardless of the DA value. When set to 0, ARP packets are treated as normal packets and must meet other DA match criteria for acceptance. |
| 21 | MHEN | Multicast Hash Enable | When set to 1, this bit allows hash table comparison for multicast addresses, i.e. a hash table hit for a multicast addressed packet will be accepted. When set to 0, multicast hash hits will not be used for packet acceptance. |
| 20 | UHEN | Unicast Hash Enable | When set to 1, this bit allows hash table comparison for unicast addresses, i.e. a hash table hit for a unicast addressed packet will be accepted. When set to 0, unicast hash hits will not be used for packet acceptance. |
| 19 | ULM | U/L bit Mask | When set to 1, this bit will cause the U/L bit (2nd MSb) of the DA to be ignored during comparison with the perfect match register. |
| 18-10 | | | unused - returns 0 |

4.0 Register Set (Continued)

| | | | |
|-----|--------|--|---|
| 9-0 | RFADDR | Receive Filter Extended Register Address | <p>Selects which internal receive filter register is accessible via RFDR:</p> <p>Perfect Match Register (PMATCH)</p> <p>000h - PMATCH octets 1-0</p> <p>002h - PMATCH octets 3-2</p> <p>004h - PMATCH octets 5-4</p> <p>Pattern Count Registers (PCOUNT)</p> <p>006h - PCOUNT1, PCOUNT0</p> <p>008h - PCOUNT3, PCOUNT2</p> <p>SecureOn Password Register (SOPAS)</p> <p>00Ah - SOPAS octets 1-0</p> <p>00Ch - SOPAS octets 3-2</p> <p>00Eh - SOPAS octets 5-4</p> <p>Filter Memory</p> <p>100h-3FEh - Rx filter memory (Hash table/pattern buffers)</p> |
|-----|--------|--|---|

4.2.21 Receive Filter/Match Data Register

The RFDR register is used for reading from and writing to the internal receive filter registers, the pattern buffer memory, and the hash table memory.

Tag: RFDR

Size: 32 bits

Hard Reset: 00000000h

Offset: 004Ch

Access: Read Write

Soft Reset: 00000000h

| bit | tag | description | usage |
|-------|--------|---------------------|---|
| 31-18 | | | unused |
| 17-16 | BMASK | Byte mask | Used as byte mask values for pattern match template data. |
| 15-0 | RFDATA | Receive Filter Data | |

4.0 Register Set (Continued)

4.2.22 Receive Filter Logic

The Receive Filter Logic supports a variety of techniques for qualifying incoming packets. The most basic filtering options include Accept All Broadcast, Accept All Multicast and Accept All Unicast packets. These options are enabled by setting the corresponding bit in the Receive Filter Control Register, RFCR. Accept on Perfect Match, Accept on Pattern Match, Accept on Multicast Hash and Accept on Unicast Hash are more robust in their filtering capabilities, but require additional programming of the Receive Filter registers and the internal filter RAM.

4.2.22.1 Accept on Perfect Match

When enabled, the Perfect Match Register is used to compare against the DA for packet acceptance. The Perfect Match Register is a 6-byte register accessed indirectly through the RFCR. The address of the internal receive filter register to be accessed is programmed through bits 9:0 of the RFCR. The Receive Filter Data Register, RFDR, is used for reading/writing the actual data.

RX Filter Address: 000h - Perfect match octets 1-0
002h - Perfect Match octets 3-2
004h - Perfect Match octets 5-4

Octet 0 of the Perfect Match Register corresponds to the first octet of the packet as it appears on the wire. Octet 5 corresponds to the last octet of the DA as it appears on the wire.

The following steps are required to program the RFCR to accept packets on a perfect match of the DA.

Example: Destination Address of 08-00-17-07-28-55

```
iow | $RFCR (0000)perfect match register, octets 1-0
iow | $RFDR (0008)write address, octets 1-0
iow | $RFCR (0002)perfect match register, octets 3-2
iow | $RFDR (0717)write address, octets 3-2
iow | $RFCR (0004)perfect match register, octets 5-4
iow | $RFDR (5528)write address, octets 5-4
iow | $RFDR (0606)
```

(\$RFEN|\$APM) -enable filtering, perfect match

4.2.22.2 Accept on Pattern Match

The Receive Filter Logic provides access to 4 separate internal RAM-based pattern buffers to be used as additional perfect match address registers. All pattern buffers are 128 bytes deep, allowing perfect match on the first 128 bytes of a packet.

When one or more of the Pattern Match enable bits are set in the RFCR, a packet will be accepted if it matches the associated pattern buffer. As indicated above, the pattern buffers are 128 bytes deep organized as 64 words, where a word is 18 bits. Bits 17 and 18 of a respective word are mask bits for byte 0 and byte 1 of the 16-bit data word (bits 15:0). An incoming packet is compared to each enabled pattern buffer on a byte by byte basis for a specified count. Masking a pattern byte results in a byte match regardless of its value (a don't care). A count value must be programmed for each pattern buffer to be used for comparison. The minimum valid count is 1 byte and the maximum valid count is 128 for all pattern buffers. The pattern count registers are internal receive filter registers accessed through the RFCR and the RFDR. The Receive Filter memory is also accessed through the RFCR and the RFDR. A memory map of the internal pattern RAM is shown in Figure 4-1.

4.0 Register Set (Continued)

| | Byte1 Mask Bit | Byte0 Mask Bit | | |
|--------------------|----------------------|----------------------|----------|----------|
| Pattern 3 Word 3Fh | | | byte 127 | byte126 |
| ⋮ | | | ⋮ | ⋮ |
| Pattern 3 Word 0 | | | byte 1 | byte 0 |
| Pattern 2 Word 3Fh | | | byte 127 | byte126 |
| ⋮ | | | ⋮ | ⋮ |
| Pattern 2 Word 0 | | | byte 1 | byte 0 |
| Pattern 1 Word 3Fh | | | byte 127 | byte 126 |
| ⋮ | | | ⋮ | ⋮ |
| Pattern 1 Word 0 | | | byte 1 | byte 0 |
| Pattern 0 Word 3Fh | | | byte 127 | byte 126 |
| ⋮ | | | ⋮ | ⋮ |
| Pattern 0 Word 1 | | | byte 3 | byte 2 |
| Pattern 0 Word 0 | | | byte 1 | byte 0 |
| Bit# | 17 | 16 | 15 | 0 |

Figure 4-1 Pattern Buffer Memory -100h words (word=18bits)

Example: Pattern match on the following destination addresses:

| | |
|-------------------|---------------|
| 02-00-03-01-04-02 | match 6 bytes |
| 12-10-13-11-14-12 | match 4 bytes |
| 22-20-23-21-24-22 | match 6 bytes |
| 32-30-33-31-34-32 | match 4 bytes |

write data pattern into buffer 2

```
iowrite I RFCR (300)
iowrite I RFDR (2022)
iowrite I RFCR (302)
iowrite I RFDR (2123)
iowrite I RFCR (304)
iowrite I RFDR (2224)
```

RFDR = (IO base + 48h)
RFDR = (IO base + 4Ch))

write counts

```
iowrite I RFCR (0006) # pattern count registers 1, 0
iowrite I RFDR (0406) # count 1 = 4, count 0= 6
iowrite I RFCR (0008) # pattern count registers 3, 2
iowrite I RFDR (0406) # count 3 = 4, count 2 = 6
```

write data pattern into buffer 3

```
iowrite I RFCR (380)
iowrite I RFDR (3032)
iowrite I RFCR (382)
iowrite I RFDR (3133)
iowrite I RFCR (384)
iowrite I RFDR (3234)
```

write data pattern into buffer 0

```
iowrite I RFCR (200)
iowrite I RFDR (0002)
iowrite I RFCR (202)
iowrite I RFDR (0103)
iowrite I RFCR (204)
iowrite I RFDR (0204)
```

#enable receive filter on all patterns

```
iowrite I RFCR (RFEN | APAT3 | APAT2 | APAT1 | APAT0)
```

write data pattern into buffer 1

```
iowrite I RFCR (280)
iowrite I RFDR (1012)
iowrite I RFCR (282)
iowrite I RFDR (1113)
iowrite I RFCR (284)
iowrite I RFDR (1214)
```

Example of how to mask out a byte in a pattern:

write data pattern into buffer 0

```
iowrite I RFCR (200)
iowrite I RFDR (10002) #mask byte 0 (value = 02)
iowrite I RFCR (202)
iowrite I RFDR (20103) #mask byte 1 (value = 01)
iowrite I RFCR (204)
iowrite I RFDR (30204) #mask byte 0 and 1
```

4.0 Register Set (Continued)

4.2.2.3 Accept on Multicast or Unicast Hash

Multicast and Unicast addresses may be further qualified by use of the receive filter hash functions. An internal 2048 bit (256 byte) RAM-based hash table is used to perform imperfect filtering of multicast or unicast packets. By enabling either Multicast Hashing or Unicast Hashing in the RFCR, the receive filter logic will use the 11 most significant bits of the destination addresses' CRC as an index into the Hash Table memory. The upper 7 bits represent the word address and the lower 4 bits select the bit within the word. If the corresponding bit is set, then the packet is accepted, otherwise the packet is rejected. The hash table memory is accessed through the RFCR and the RFDR. Refer to Figure 4-2 for a memory map. Below is example code for setting a bit in the hash table.

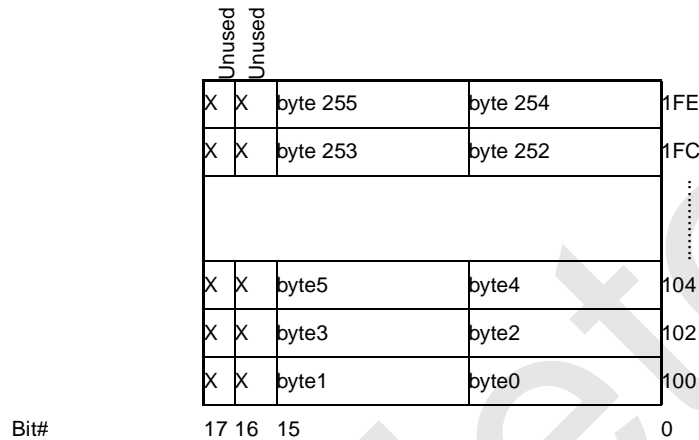


Figure 4-2 Hash Table Memory - 100h bytes addressed on word boundaries

Given a CRC of F9E80000:

RFCR = (IO base + 48h)

RFDR = (IO base + 4Ch)

Bits 31-25 select which 16-bit word

Word = 7C

#Lower 24-21 bits select which bit in 16-bit word

Bit =F

Select bit to set/clear

hash_bit = (0001<<bit)

#Write word address into RFCR

iowrite I RFCR (100 + word)

Read indexed word from table

ioread I RFDR

set hash_word =(result | hash_bit) #OR in the hash bit to set

#Write to the hash table

iowrite I RFDR (hash_word)

Enable multicast and/or unicast hash

iowrite I RFCR (RFEN | MHEN | UHEN)

4.0 Register Set (Continued)

4.2.23 Boot ROM Address Register

The BRAR is used to setup the address for an access to an external ROM/FLASH device.

Tag: BRAR *Size:* 32 bits *Hard Reset:* FFFFFFFFh
Offset: 0050h *Access:* Read Write *Soft Reset:* FFFFFFFFh

| bit | tag | description | usage |
|-------|---------|------------------|---|
| 31 | AUTOINC | Auto-Increment | When set, the contents of ADDR will auto increment with every 32-bit access to the BRDR register. |
| 30-16 | | | unused |
| 15-0 | ADDR | Boot ROM Address | 16-bit address used to access the external Boot ROM. |

4.2.24 Boot ROM Data Register

The BRDR is used to read and write ROM/FLASH data from the data from/to an external ROM/FLASH device.

Tag: BRDR *Size:* 32 bits *Hard Reset:* undefined
Offset: 0054h *Access:* Read Write *Soft Reset:* undefined

| bit | tag | description | usage |
|------|------|---------------|---|
| 31-0 | DATA | Boot ROM Data | Access port to external Boot ROM. Software can use BRAR and BRDR to read (and write if FLASH memory is used) the external Boot ROM. All accesses must be 32-bits wide and aligned on 32-bit boundaries. |

4.2.25 Silicon Revision Register

Tag: SRR *Size:* 32 bits *Hard Reset:* as defined
Offset: 0058h *Access:* Read Only *Soft Reset:* unchanged

| bit | tag | description | usage |
|-------|-----|----------------|--|
| 31-16 | | | unused (reads return 0) |
| 15-0 | REV | Revision Level | Silicon Revision for the DP83820. Rev B 0103h |

4.0 Register Set (Continued)

4.2.26 Management Information Base Control Register

The MIBC register is used to control access to the statistics block and the warning bits and to control the collection of management information statistics.

Tag: MIBC

Size: 32 bits

Hard Reset: 00000002h

Offset: 005ch

Access: Read Write

Soft Reset: 00000002h

| bit | tag | description | usage |
|------|------|------------------------|---|
| 31-4 | | | unused |
| 3 | MIBS | MIB Counter Strobe | Writing a 1 to this bit location causes the counters in all enabled blocks to increment by 1, providing a single-step test function. The MIBS bit is always read back as 0. This bit is used for test purposes only and should be set to 0 for normal counter operation. |
| 2 | ACLR | Clear all counters | When set to a 1, this bit forces all counters to be reset to 0. This bit is always read back as 0. |
| 1 | FRZ | Freeze all counters | When set to a 1, this bit forces count values to be frozen such that a read of the statistic block will represent management statistics at a given instant in time. When set to 0, the counters will increment normally and may be read individually while counting. |
| 0 | WRN | Warning Test Indicator | This field is read only. This bit is set to 1 when all statistic counters have reached their respective overflow warning condition. WRN will be cleared after one or more of the statistic counters have been cleared. |

4.0 Register Set (Continued)

4.2.27 Management Information Base Registers

The counters provide a set of statistics compliant with the following management specifications: MIB II, Ether-like MIB, and IEEE MIB. The values provided are accessed through the various registers as shown below. All MIB counters are cleared to 0 when read.

Due to cost and space limitations, the counter bit widths provided in the DP83820 MIB are less than the bit widths called for in the above specifications. It is assumed that management agent software will maintain a set of fully compliant statistic values ("software" counters), utilizing the hardware counters to reduce the frequency at which these

"software" counters must be updated. Sizes for specific hardware statistic counters were chosen such that the count values will not roll over in less than 30 ms if incremented at the theoretical maximum rates described in the above specifications. However, given that the theoretical maximum counter rates do not represent realistic network traffic and events, the actual rollover rates for the hardware counters are more likely to be on the order of several seconds. The hardware counters are updated automatically by the MAC on the occurrence of each event.

Table 4-3 MIB Registers

| offset | tag | size | warning (MS bits) | description |
|--------|----------------|------|-------------------|--|
| 0060h | RXErroredPkts | 16 | 8 | Packets received with errors. This counter is incremented for each packet received with errors. This count includes packets which are automatically rejected from the FIFO due to both wire errors and FIFO overruns. |
| 0064h | RXFCSErrors | 16 | 8 | Packets received with frame check sequence errors. This counter is incremented for each packet received with a Frame Check Sequence error (bad CRC). Note: For the MII interface, an FCS error is defined as a resulting invalid CRC after CRS goes invalid and an even number of bytes have been received. |
| 0068h | RXMsdPktErrors | 16 | 8 | Packets missed due to FIFO overruns. This counter is incremented for each receive aborted due to data or status FIFO overruns (insufficient buffer space). |
| 006Ch | RXFAErrors | 16 | 8 | Packets received with frame alignment errors. This counter is incremented for each packet received with a Frame Check Sequence error (bad CRC). Note: For the MII interface, an FAE error is defined as a resulting invalid CRC on the last full octet, and an odd number of nibbles have been received (Dribble nibble condition with a bad CRC). |
| 0070h | RXSymbolErrors | 16 | 8 | Packets received with one or more symbol errors. This counter is incremented for each packet received with one or more symbol errors detected. Note: For the MII interface, a symbol error is indicated by the RX_ER signal becoming active for one or more clocks while the RX_DV signal is active (during valid data reception). |
| 0074h | RXFrameTooLong | 16 | 8 | Packets received with length greater than 1518 bytes (too long packets). This counter is incremented for each packet received with greater than the 802.3 standard maximum length of 1518 bytes. |
| 0078h | RXIRLErrors | 16 | 8 | Packets received with In Range Length errors. This counter increments for packets received with a MAC length/type value between 64 and 1518 bytes, inclusive, that does not match the number of bytes received. This counter also increments for packets with a MAC length/type field of less than 64 bytes and more than 64 bytes received. |
| 007Ch | RXBadOpCodes | 16 | 8 | Packets received with a valid MAC control type and an opcode for a function that is not supported by the device. |
| 0080h | RXPauseFrames | 16 | 8 | MAC control Pause frames received. |
| 0084h | TXPauseFrames | 16 | 8 | MAC control Pause frames transmitted. |
| 0088h | TXSQEErrors | 8 | 4 | Loss of collision heartbeat during transmission. This counter is incremented when the collision heartbeat pulse is not detected by the PMD after a transmission. |

4.0 Register Set (Continued)

4.2.28 Transmit Descriptor Pointer 1 Register

This register points to the Transmit Descriptor for Priority Queue 1.

Tag: TXDP1 *Size:* 32 bits *Hard Reset:* 00000000h
Offset: 00A0h *Access:* Read Write *Soft Reset:* 00000000h

| bit | tag | description | usage |
|------|-------|-------------------------------|--|
| 31-3 | TXDP1 | Transmit Descriptor Pointer 1 | The current value of the transmit descriptor pointer for Priority Queue 1. When initializing the queue for a new transmission, software must set TXDP to the address of a completed transmit descriptor. While the transmit state machine is active, TXDP will follow the state machine as it advances through a linked list of active descriptors. If the link field of the current transmit descriptor is NULL (signifying the end of the list), TXDP will not advance, but will remain on the current descriptor. Any subsequent writes to the TXE bit of the CR register will cause the transmit state machine to reread the link field of the current descriptor to check for new descriptors that may have been appended to the end of the list. Transmit descriptors must be aligned on an even 64-bit boundary in host memory (A2-A0 must be 0). |
| 2-0 | | | unused |

4.2.29 Transmit Descriptor Pointer 2 Register

This register points to the Transmit Descriptor for Priority Queue 2.

Tag: TXDP2 *Size:* 32 bits *Hard Reset:* 00000000h
Offset: 00A4h *Access:* Read Write *Soft Reset:* 00000000h

| bit | tag | description | usage |
|------|-------|-------------------------------|--|
| 31-3 | TXDP2 | Transmit Descriptor Pointer 2 | The current value of the transmit descriptor pointer for Priority Queue 2. When initializing the queue for a new transmission, software must set TXDP to the address of a completed transmit descriptor. While the transmit state machine is active, TXDP will follow the state machine as it advances through a linked list of active descriptors. If the link field of the current transmit descriptor is NULL (signifying the end of the list), TXDP will not advance, but will remain on the current descriptor. Any subsequent writes to the TXE bit of the CR register will cause the transmit state machine to reread the link field of the current descriptor to check for new descriptors that may have been appended to the end of the list. Transmit descriptors must be aligned on an even 64-bit boundary in host memory (A2-A0 must be 0). |
| 2-0 | | | unused |

4.2.30 Transmit Descriptor Pointer 3 Register

This register points to the Transmit Descriptor for Priority Queue 3.

Tag: TXDP3 *Size:* 32 bits *Hard Reset:* 00000000h
Offset: 00A8h *Access:* Read Write *Soft Reset:* 00000000h

| bit | tag | description | usage |
|------|-------|-------------------------------|--|
| 31-3 | TXDP3 | Transmit Descriptor Pointer 3 | The current value of the transmit descriptor pointer for Priority Queue 3. When initializing the queue for a new transmission, software must set TXDP to the address of a completed transmit descriptor. While the transmit state machine is active, TXDP will follow the state machine as it advances through a linked list of active descriptors. If the link field of the current transmit descriptor is NULL (signifying the end of the list), TXDP will not advance, but will remain on the current descriptor. Any subsequent writes to the TXE bit of the CR register will cause the transmit state machine to reread the link field of the current descriptor to check for new descriptors that may have been appended to the end of the list. Transmit descriptors must be aligned on an even 64-bit boundary in host memory (A2-A0 must be 0). |
| 2-0 | | | unused |

4.0 Register Set (Continued)

4.2.31 Receive Descriptor Pointer 1 Register

This register points to the Receive Descriptor for Priority Queue 1.

Tag: RXDP1 *Size:* 32 bits *Hard Reset:* 00000000h
Offset: 00B0h *Access:* Read Write *Soft Reset:* 00000000h

| bit | tag | description | usage |
|------|-------|------------------------------|---|
| 31-3 | RXDP1 | Receive Descriptor Pointer 1 | The current value of the receive descriptor pointer for Priority Queue 1. Packets will be stored in Priority Queue 1 based on the number of priority queues enabled and the priority field in the VLAN tag. When the receive state machine is idle, software must set RXDP1 to the address of an available receive descriptor, and then enable the queue by writing to the RXE bit in the CR with the RXPRI[1] bit set. While the receive state machine is active, RXDP1 will follow the state machine as it advances through a linked list of available descriptors. If the link field of the current receive descriptor is NULL (signifying the end of the list), RXDP1 will not advance, but will remain on the current descriptor. Any subsequent writes to the RXE bit of the CR register will cause the receive state machine to reread the link field of the current descriptor to check for new descriptors that may have been appended to the end of the list. Software should not write to this register unless the receive state machine is idle. Receive descriptors must be aligned on 64-bit boundaries (A2-A0 must be zero). |
| 2-0 | | | unused |

4.2.32 Receive Descriptor Pointer 2 Register

This register points to the Receive Descriptor for Priority Queue 2.

Tag: RXDP2 *Size:* 32 bits *Hard Reset:* 00000000h
Offset: 00B4h *Access:* Read Write *Soft Reset:* 00000000h

| bit | tag | description | usage |
|------|-------|------------------------------|---|
| 31-3 | RXDP2 | Receive Descriptor Pointer 2 | The current value of the receive descriptor pointer for Priority Queue 2. Packets will be stored in Priority Queue 2 based on the number of priority queues enabled and the priority field in the VLAN tag. When the receive state machine is idle, software must set RXDP2 to the address of an available receive descriptor, and then enable the queue by writing to the RXE bit in the CR with the RXPRI[2] bit set. While the receive state machine is active, RXDP2 will follow the state machine as it advances through a linked list of available descriptors. If the link field of the current receive descriptor is NULL (signifying the end of the list), RXDP2 will not advance, but will remain on the current descriptor. Any subsequent writes to the RXE bit of the CR register will cause the receive state machine to reread the link field of the current descriptor to check for new descriptors that may have been appended to the end of the list. Software should not write to this register unless the receive state machine is idle. Receive descriptors must be aligned on 64-bit boundaries (A2-A0 must be zero). |
| 2-0 | | | unused |

4.0 Register Set (Continued)

4.2.33 Receive Descriptor Pointer 3 Register

This register points to the Receive Descriptor for Priority Queue 3 (highest priority).

Tag: RXDP3 Size: 32 bits Hard Reset: 00000000h
 Offset: 00B8h Access: Read Write Soft Reset: 00000000h

| bit | tag | description | usage |
|------|-------|------------------------------|---|
| 31-3 | RXDP3 | Receive Descriptor Pointer 3 | The current value of the receive descriptor pointer for Priority Queue 3. Packets will be stored in Priority Queue 3 based on the number of priority queues enabled and the priority field in the VLAN tag. When the receive state machine is idle, software must set RXDP3 to the address of an available receive descriptor, and then enable the queue by writing to the RXE bit in the CR with the RXPRI[3] bit set. While the receive state machine is active, RXDP3 will follow the state machine as it advances through a linked list of available descriptors. If the link field of the current receive descriptor is NULL (signifying the end of the list), RXDP3 will not advance, but will remain on the current descriptor. Any subsequent writes to the RXE bit of the CR register will cause the receive state machine to reread the link field of the current descriptor to check for new descriptors that may have been appended to the end of the list. Software should not write to this register unless the receive state machine is idle. Receive descriptors must be aligned on 64-bit boundaries (A2-A0 must be zero). |
| 2-0 | | | unused |

4.2.34 VLAN/IP Receive Control Register

This register allows enabling of the various VLAN tag handling features and IP Checksum offload features.

Tag: VRCR Size: 32 bits Hard Reset: 00000000h
 Offset: 00BCh Access: Read Write Soft Reset: 00000000h

| bit | tag | description | usage |
|------|-------|----------------------------|---|
| 31-8 | | | unused |
| 7 | RUDPE | Reject UDP Checksum Errors | When set to 1, all packets with UDP headers that have errors in the UDP checksum field will be rejected. If IPEN is 0, this bit will be ignored. |
| 6 | RTCPE | Reject TCP Checksum Errors | When set to 1, all packets with TCP headers that have errors in the TCP checksum field will be rejected. If IPEN is 0, this bit will be ignored. |
| 5 | RIPE | Reject IP Checksum Errors | When set to 1, all packets with IP headers that have errors in the IP checksum field will be rejected. If IPEN is 0, this bit will be ignored. |
| 4 | IPEN | IP Checksum Enable | When set to a 1, the receiver will detect IP, TCP, and UDP headers, and validate the checksum fields. |
| 3 | DUTF | Discard Untagged Frames | Receiver will discard any frames without a 1 VLAN tag. |
| 2 | DVTF | Discard VLAN Tagged Frames | Receiver will discard any frames with a VLAN tag. |
| 1 | VTREN | VLAN Tag Removal Enable | Enables stripping of the VLAN tag upon detection. If VTDEN is not set, then this bit will have no effect. |
| 0 | VTDEN | VLAN Tag Detection Enable | Enable detection of VLAN packets based on VLAN type field as configured in the VLAN Data Register. VLAN status, including user_priority, CFI and VID fields, will be posted in the EXTSTS field of the receive packet descriptor. |

4.0 Register Set (Continued)

4.2.35 VLAN/IP Transmit Control Register

This register allows enabling of the various VLAN tag handling features and IP checksum offload features.

Tag: VTCR
Offset: 00C0h

Size: 32 bits
Access: Read Write

Hard Reset: 00000000h
Soft Reset: 00000000h

| bit | tag | description | usage |
|------|-------|--------------------------------|---|
| 31-4 | | | unused |
| 3 | PPCHK | Per-Packet Checksum Generation | Enables IP/TCP/UDP Checksum generation on a per-packet basis. Uses individual enable controls from the EXTSTS field of the packet descriptor. |
| 2 | GCHK | Global Checksum Generation | Enables IP/TCP/UDP Checksum generation on all transmit packets. |
| 1 | VPPTI | VLAN Per-Packet Tag Insertion | Insert VLAN tag in on a per-packet basis. Uses the VLAN Data Register VLAN type field for the VLAN type. Uses the user_priority, CFI and VLAN ID fields from the EXTSTS field of the packet descriptor. |
| 0 | VGTI | VLAN Global Tag Insertion | Insert VLAN tag in all transmit packets. Uses the VLAN Data Register data for the 32-bit VLAN tag to be inserted. |

4.2.36 VLAN Data Register

This register contains data for VLAN tag insertion and detection.

Tag: VDR
Offset: 00C4h

Size: 32 bits
Access: Read Write

Hard Reset: 00000081h
Soft Reset: 00000081h

| bit | tag | description | usage |
|-------|-------|------------------------------------|---|
| 31-16 | VTCL | VLAN Tag Control Information Field | This field is the 2-octet VLAN TCI field. It is used by the transmitter during Global Tag Insertion. It contains the VLAN user_priority, CFI and VID (VLAN Identifier) fields. It is not used by the receiver. |
| 15-0 | VTYPE | VLAN Type Field | This field is the 2-octet VLAN type field. By default this contains the 802.1Q Tag Type of 81-00. In order to represent the order the bytes will be shifted on the wire, it actually contains a value of 0081h. This field is used by the transmitter for Global Tag Insertion and by the receiver for Tag detection. |

4.2.37 Clockrun Control/Status Register

Tag: CCSR
Offset: 00CCh

Size: 32 bits
Access: Read Write

Hard Reset: 00000000h
Soft Reset: 00000000h

| bit | tag | description | usage |
|-------|-----------|---------------|---|
| 31-16 | | | unused (reads return 0) |
| 15 | PMESTS | PME Status | Sticky bit which represents the state of the PME/CLKRUN logic, regardless of the state of the PMEEN bit. Mirrored from PCI configuration register PMCSR. Writing a 1 to this bit clears it. |
| 14-9 | | | unused (reads return 0) |
| 8 | PMEEN | PME Enable | When set to 1, this bit enabled the assertion of the PMEN pin. When 0, the PMEN pin is forced to be inactive. This value can be loaded from the EEPROM. Mirrored from PCI configuration register PMCSR. |
| 7-1 | | | unused (reads return 0) |
| 0 | CLKRUN_EN | CLKRUN Enable | When set to 1, this bit enables the CLKRUN logic and allows the assertion of the CLKRUN_N pin. When 0, the CLKRUN function is disabled. |

4.0 Register Set (Continued)

4.2.38 TBI Control Register

This register is used to enable and/or restart TBI auto-negotiation. It is also used to enable PCS loopback of TBI data.

Tag: TBICR *Size:* 32 bits *Hard Reset:* 00000000h
Offset: 00E0h *Access:* Read Write *Soft Reset:* 00000000h

| bit | tag | description | usage |
|-------|---------------|---|--|
| 15 | | | unused - Returns 0 |
| 14 | MR_LOOPBACK | TBI PCS Loopback Enable | When set to a 1, indicates to TBI that the interfacing PHY device is in loopback mode (i.e. signal detect not necessarily required). |
| 13 | | | unused - returns 0 |
| 12 | MR_AN_ENABLE | TBI Auto-Negotiation Enable | When set to a 1, enables the Auto-negotiation function for the TBI interface. R/W |
| 11-10 | | | unused - returns 0 |
| 9 | MR_RESTART_AN | Restarts the TBI Auto-negotiation Process | When set to a 1, TBI Auto-Negotiation is restarted. This bit allows management control of renegotiation. |
| 8-0 | | | unused - returns 0 |

4.2.39 TBI Status Register

This register indicates the link status and the auto-negotiation status for the TBI interface.

Tag: TBISR *Size:* 32 bits *Hard Reset:* 00000000h
Offset: 00E4h *Access:* Read Only *Soft Reset:* 00000000h

| bit | tag | description | usage |
|------|----------------|---|---|
| 15-6 | | | unused - returns 0 |
| 5 | MR_LINK_STATUS | Link Status of the TBI Interface | Read-only bit, when set to a 1, indicates that the TBI interface is ready to transmit and receive data. |
| 4-3 | | | unused - returns 0 |
| 2 | MR_AN_COMPLETE | TBI Auto-negotiation Completed Successfully | Read-only bit, when set to a 1, indicates that the TBI interface has successfully completed auto-negotiation. |
| 1-0 | | | unused - returns 0 |

4.2.40 TBI Auto-Negotiation Advertisement Register

This register is configured before auto-negotiation begins and contains the advertised ability of the local device.

Tag: TANAR *Size:* 32 bits *Hard Reset:* 00000000h
Offset: 00E8h *Access:* Read Write *Soft Reset:* 00000000h

| bit | tag | description | usage |
|-----|-----|-----------------------------|--|
| 15 | NP | Next Page Exchange Required | When set to a 1, this bit indicates that next page transmission is requested. Subsequent next pages may set the NP bit to a 0 to indicate next page transmission is completed. A device may implement next page ability and choose not to engage in a next page exchange by clearing this bit. |
| 14 | | | mr_adv_ability |

4.0 Register Set (Continued)

| | | | |
|-------|----------|---------------------------|--|
| 13-12 | RF2, RF1 | Remote Fault | Read-only bits indicating that a fault or error condition has occurred. The default value is 00. 00 - No error, Link OK 10 - Offline 01 - Link Failure 11 - Auto-Negotiation Error |
| 11-9 | | | unused - returns 0 |
| 8-7 | PS2, PS1 | Pause Capability Encoding | PS1 indicates that the device is capable of providing symmetric PAUSE functions. PS2 indicates that asymmetric PAUSE operation is supported. The value of PS1 when PS2 is set indicates the direction PAUSE frames are supported for flow across the link. Asymmetric PAUSE configuration results in independent enabling of the PAUSE receive and PAUSE transmit functions for PAUSE configuration resolution. 00 - No PAUSE 10 - Asymmetric PAUSE Toward Link Partner 01 - Symmetric PAUSE 11 - Both Symmetric PAUSE and Asymmetric PAUSE Toward Local Device. |
| 6 | HALF_DUP | Half Duplex | When set to 1, advertises half duplex capability. |
| 5 | FULL_DUP | Full Duplex | When set to 1, advertises full duplex capability. |
| 4-0 | | | unused - returns 0 |

4.2.41 TBI Auto-Negotiation Link Partner Ability Register

This register contains the advertised ability of the link partner. The bit definitions are a direct representation of the link partner's base page. The value of this register is valid after successful completion of auto-negotiation or when a new base page has been received as indicated by bit 6 of the Auto-Negotiation Expansion Register.

Tag: TANLPAR *Size:* 32 bits *Hard Reset:* 00000000h
Offset: 00ECh *Access:* Read Only *Soft Reset:* 00000000h

| bit | tag | description | usage |
|-------|----------|-----------------------------|---|
| 15 | NP | Next Page Exchange Required | Read-only. Indicates that the link partner has a next page to transmit. |
| 14 | | | ignore on read - internal use only |
| 13-12 | RF2,RF1 | | Read-only. Indicates remote fault status of the link partner. |
| 11-9 | | | unused - returns 0 |
| 8-7 | PS2, PS1 | | Read-only. Indicates the PAUSE capability of the link partner. |
| 6 | HALF_DUP | | Read-only. Link partner is half duplex capable. |
| 5 | FULL_DUP | | Read-only. Link partner is full duplex capable |
| 4-0 | | | unused - returns 0 |

4.0 Register Set (Continued)

4.2.42 TBI Auto-Negotiation Expansion Register

This register is a read-only register indicating if a new base page from the link partner has been received and if the local device is next page able. Writes to this register have no effect.

Tag: TANER *Size:* 32 bits *Hard Reset:* 00000000h
Offset: 00F0h *Access:* Read Only *Soft Reset:* 00000000h

| bit | tag | description | usage |
|------|----------------|--|--|
| 15-3 | | | unused - returns 0 |
| 2 | Next Page Able | Local Device supports the Next Page Function | When set to a 1, this bit indicates that the local device supports the Next Page function. |
| 1 | Page Received | New Page Received from Link Partner | When set to a 1, this bit indicates that a new page has been received from the link partner and stored in the applicable auto-negotiation link partner ability register or next page register. |
| 0 | | | unused - returns 0 |

4.2.43 TBI Extended Status Register

This is a read-only register indicating all modes of operation for the local device. Writes to this register have no effect.

Tag: TESR *Size:* 32 bits *Hard Reset:* 0000C000h
Offset: 00F4h *Access:* Read Only *Soft Reset:* 0000C000h

| bit | tag | description | usage |
|------|------------------------|--------------------------------|--|
| 15 | 1000BASE-X Full Duplex | Full Duplex 1000BASE-X Capable | Read-only bit, set to a 1, indicating that the local device is able to perform full duplex, 1000BASE-X operations. |
| 14 | 1000BASE-X Half Duplex | Half Duplex 1000BASE-X Capable | Read-only bit, set to a 1, indicating that the local device is able to perform half duplex, 1000BASE-X operations. |
| 13-0 | | | unused - returns 0 |

5.0 DC and AC Specifications

Absolute Maximum Ratings

| | |
|--|----------------------------|
| Supply Voltage (V_{DD}) | -0.5 V to 3.6 V |
| 3.3 V PCI signaling, 5.0 V tolerant | |
| DC Input Voltage (V_{IN}) | -0.5 V to 7.0 V |
| DC Output Voltage (V_{OUT}) | -0.5 V to $V_{DD} + 0.5$ V |
| Storage Temperature Range (T_{STG}) | -65 °C to 150 °C |
| Power Dissipation (P_D) | 743 mW |
| Body Temp. (T_B) (Soldering, 10 sec) | 220 °C |
| ESD Rating | 2.0 KV |
| ($R_{ZAP} = 1.5k\Omega$, $C_{ZAP} = 120$ pF) | |
| TPTD+/- ESD Rating | 1.6 KV |
| θ_{ja} (@0 cfm, 1 Watt) | 44.5 °C/W |
| θ_{jc} (@1 Watt) | 9.5 °C/W |

Recommended Operating Conditions

| | |
|------------------------------------|-----------------------|
| Supply voltage (V_{DD}) - IO | 3.3 Volts \pm 0.3V |
| Supply voltage (V_{DD}) - Core | 1.8 Volts \pm 0.15V |
| Ambient Temperature (T_A) | 0 to 70 °C |

Note: Absolute maximum ratings are values beyond which operation is not recommended or guaranteed. Extended exposure beyond these limits may affect device reliability. They are not meant to imply that the device should be operated at these limits.

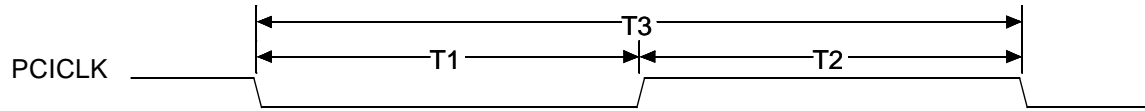
5.1 DC Specifications

| Symbol | Parameter | Conditions | Min | Typ | Max | Units |
|-----------|-----------------------------------|------------------------------------|------|-----|-----|---------|
| V_{OH} | Minimum High Level Output Voltage | $I_{OH} = -6$ mA -4 mA for PMEN | 2.4 | | | V |
| V_{OL} | Maximum Low Level Output Voltage | $I_{OL} = 6$ mA 4 mA for PMEN | | | 0.4 | V |
| V_{IH} | Minimum High Level Input Voltage | | 2.0 | | | V |
| V_{IL} | Maximum Low Level Input Voltage | | -0.5 | | 0.8 | V |
| V_{IH} | Minimum High Level Input Voltage | GMII pins | 1.7 | | | V |
| V_{IL} | Maximum Low Level Input Voltage | GMII pins | | | 0.9 | V |
| I_{IN} | Input Current | $V_{IN} = V_{DD}$ or GND | -10 | | 10 | μ A |
| I_{OZ} | TRI-STATE Output Leakage Current | $V_{OUT} = V_{DD}$ or GND | -10 | | 10 | μ A |
| I_{DD} | 3.3 V Operating Supply Current | $I_{OUT} = 0$ mA, FREQ = F_{MAX} | | 150 | | mA |
| | 3.3 V WOL standby | | | 40 | | mA |
| | 3.3 V Sleep mode | | | 30 | | mA |
| I_{DD} | 1.8 V Operating Supply Current | $I_{OUT} = 0$ mA, FREQ = F_{MAX} | | 75 | | mA |
| | 1.8 V WOL standby | | | 40 | | mA |
| | 1.8 V Sleep mode | | | 10 | 50 | mA |
| C_{IN} | CMOS Input Capacitance | | | 8 | | pF |
| C_{OUT} | CMOS Output Capacitance | | | 8 | | pF |

5.0 DC and AC Specifications (Continued)

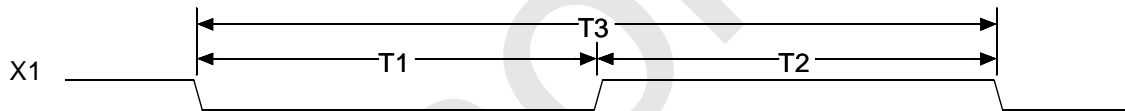
5.2 AC Specifications

5.2.1 PCI Clock Timing



| Number | Parameter | Min | Typ | Units |
|---------|-------------------|-----|----------|-------|
| 5.2.1.1 | PCICLK Low Time | 6 | | ns |
| 5.2.1.2 | PCICLK High Time | 6 | | ns |
| 5.2.1.3 | PCICLK Cycle Time | 15 | ∞ | ns |

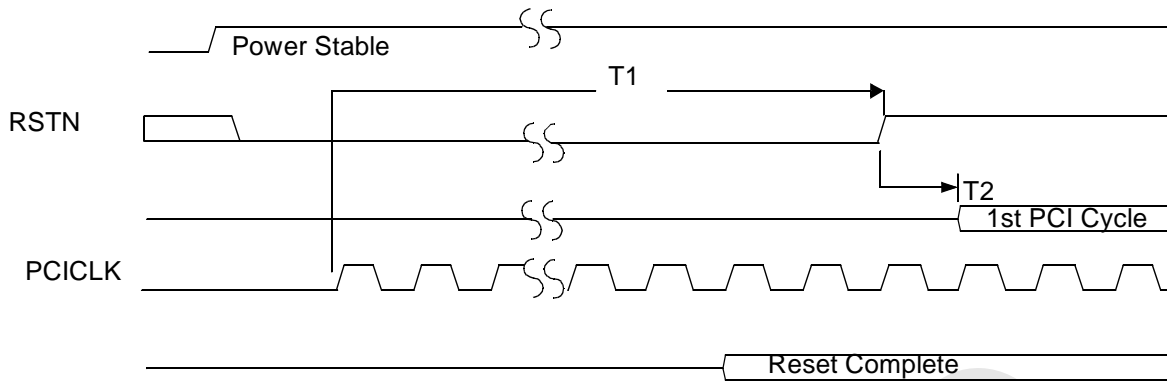
5.2.2 X1 Clock Timing



| Number | Parameter | Min | Typ | Units |
|---------|---------------|-----|-----|-------|
| 5.2.2.1 | X1 Low Time | 16 | | ns |
| 5.2.2.2 | X1 High Time | 16 | | ns |
| 5.2.2.3 | X1 Cycle Time | 40 | 40 | ns |

5.0 DC and AC Specifications (Continued)

5.2.3 Power On Reset (PCI Active)



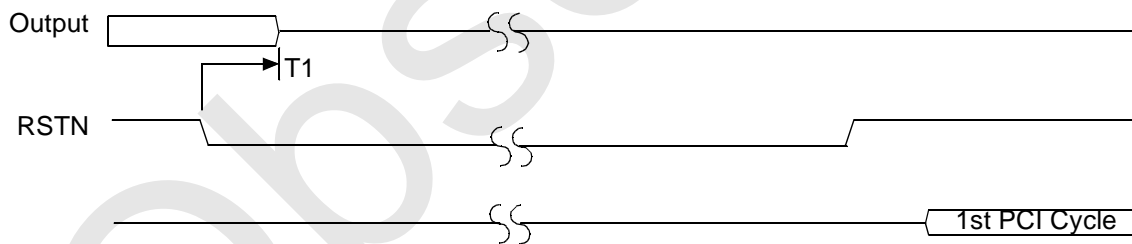
| Number | Parameter | Min | Typ | Units |
|---------|---|------|-----|-------|
| 5.2.3.1 | RSTN Active Duration from PCICLK stable | 1 | | ms |
| 5.2.3.2 | Reset Disable to 1st PCI Cycle | | | |
| | EE Enabled | 2000 | | us |
| | EE Disabled | 1 | | us |

Note 1: Minimum reset complete time is a function of the PCI, transmit, and receive clock frequencies.

Note 2: Minimum access after reset is dependent on PCI clock frequency. Accesses to DP83820 during this period will be ignored.

Note 3: EE is disabled for non power on reset.

5.2.4 Non Power On Reset

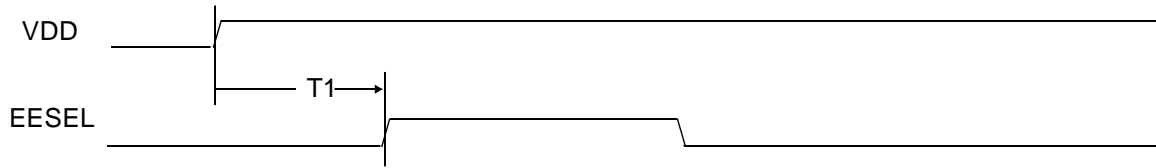


| Number | Parameter | Min | Typ | Units |
|---------|----------------------|-----|-----|-------|
| 5.2.4.1 | RSTN to Output Float | | 40 | ns |

Note 4: Minimum reset complete time is a function of the PCI, transmit, and receive clock frequencies.

5.0 DC and AC Specifications (Continued)

5.2.5 POR PCI Inactive



| Number | Parameter | Min | Typ | Units |
|---------|---|-----|-----|-------|
| 5.2.5.1 | VDD stable to EE access VDD indicates the digital supply (AUX power plane, except PCI bus power.) Guaranteed by design. | | 60 | us |

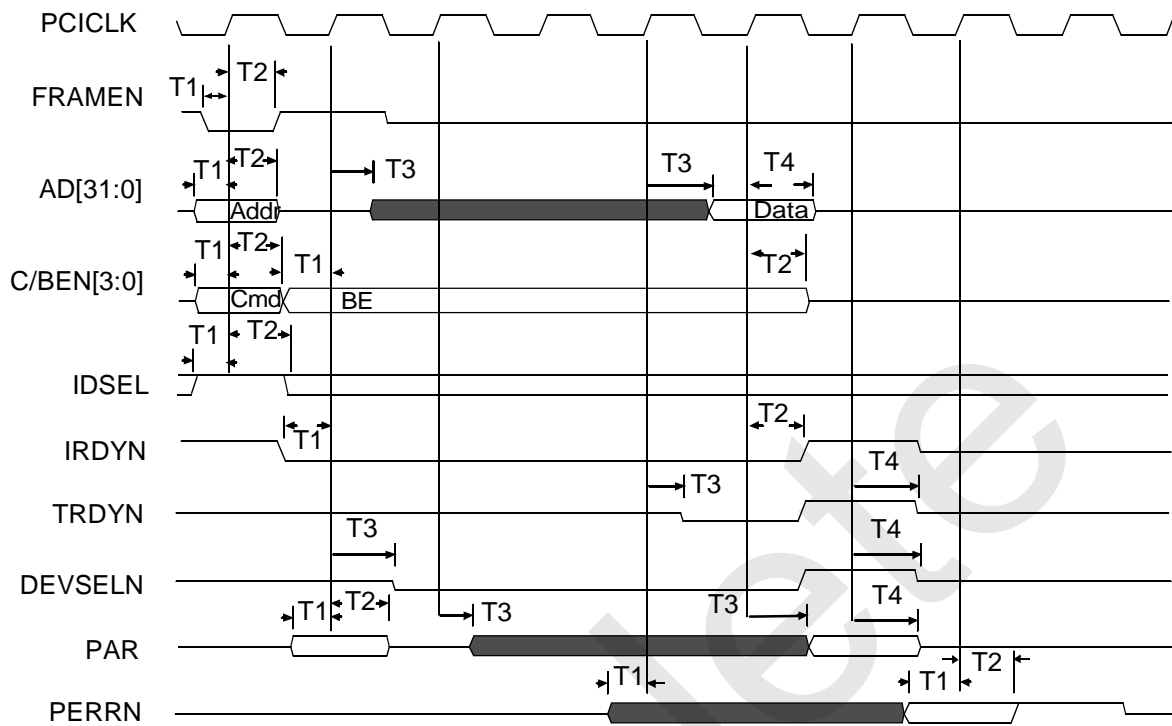
5.2.6 PCI Bus Cycles

The following table parameters apply to **ALL** the PCI Bus Cycle Timing Diagrams contained in this section.

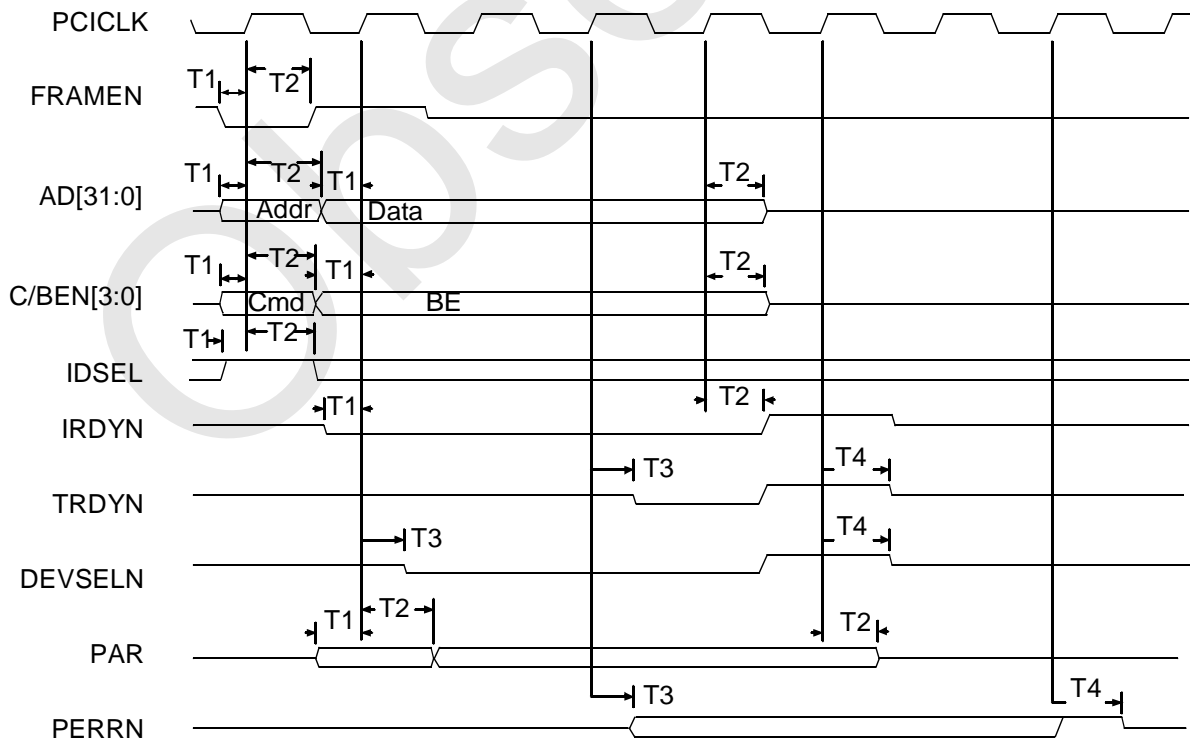
| Number | Parameter | Min | Typ | Units |
|---------|--|-----|-----|-------|
| 5.2.6.1 | Input Setup Time | 3 | | ns |
| 5.2.6.2 | Input Hold Time | 0 | | ns |
| 5.2.6.3 | Output Valid Delay | 2 | 6 | ns |
| 5.2.6.4 | Output Float Delay (t_{off} time) | | 14 | ns |
| 5.2.6.5 | Input Setup Time for GNTN - point to point | 5 | | ns |

5.0 DC and AC Specifications (Continued)

PCI Configuration Read

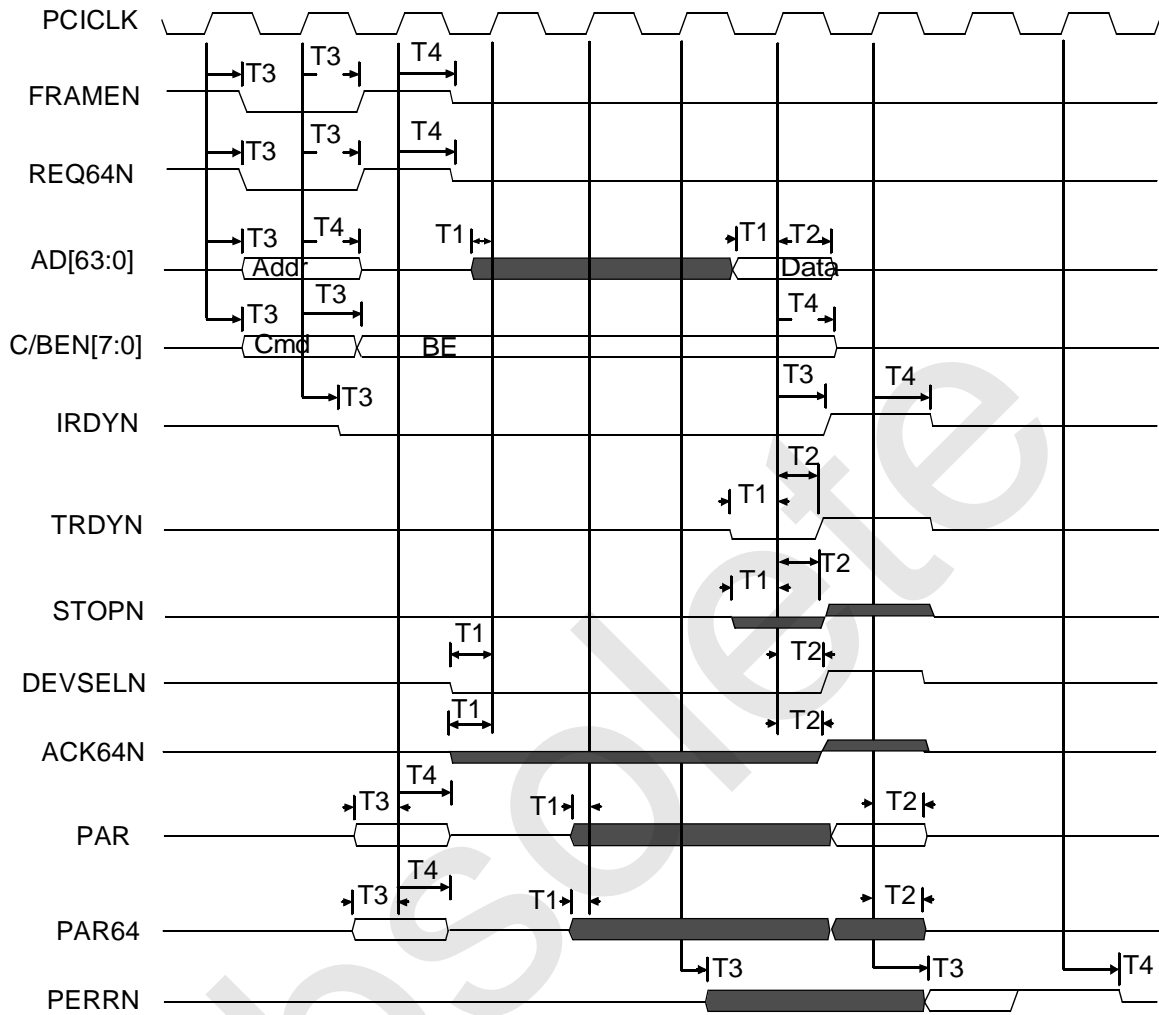


PCI Configuration Write



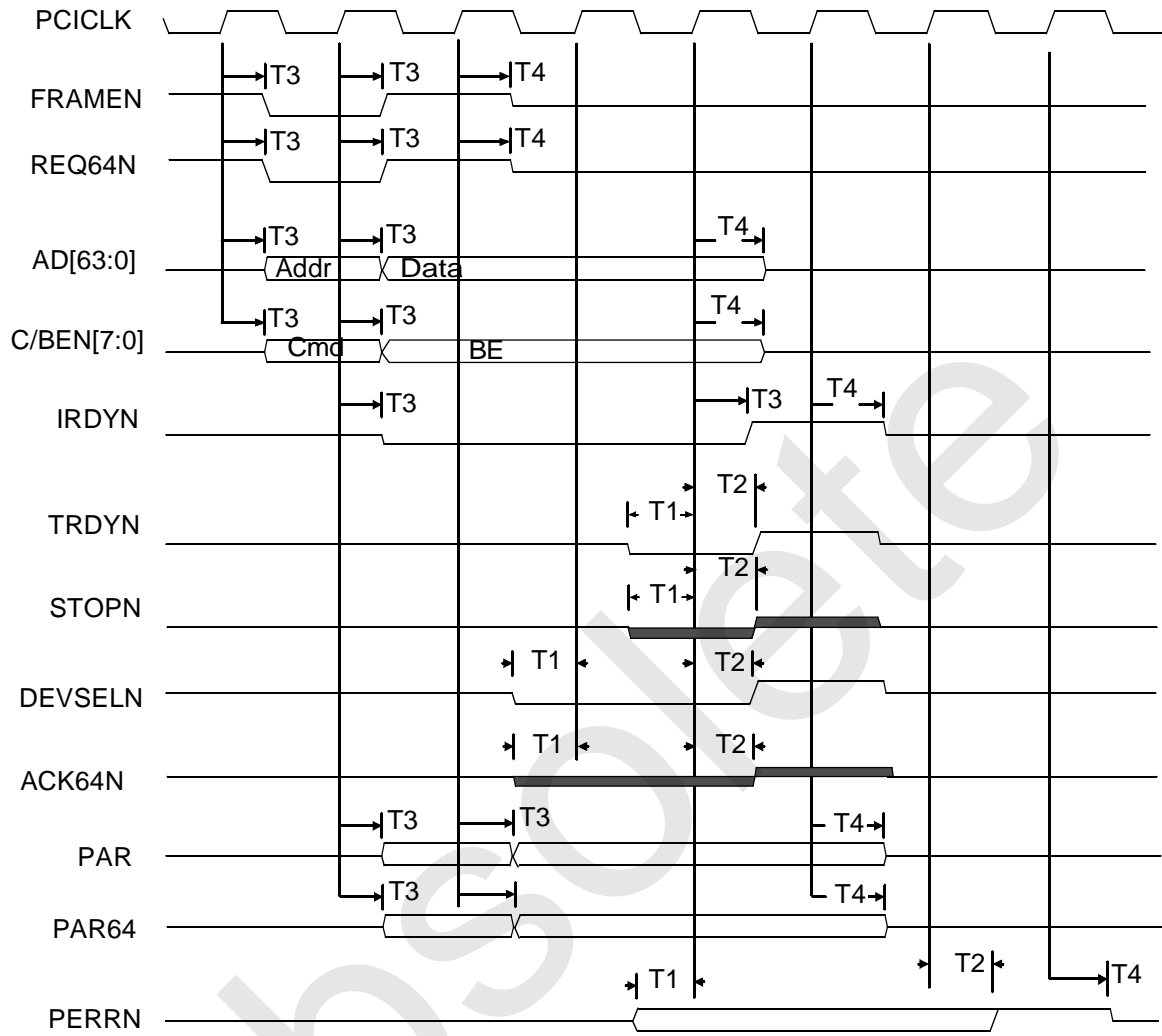
5.0 DC and AC Specifications (Continued)

PCI Bus Master Read



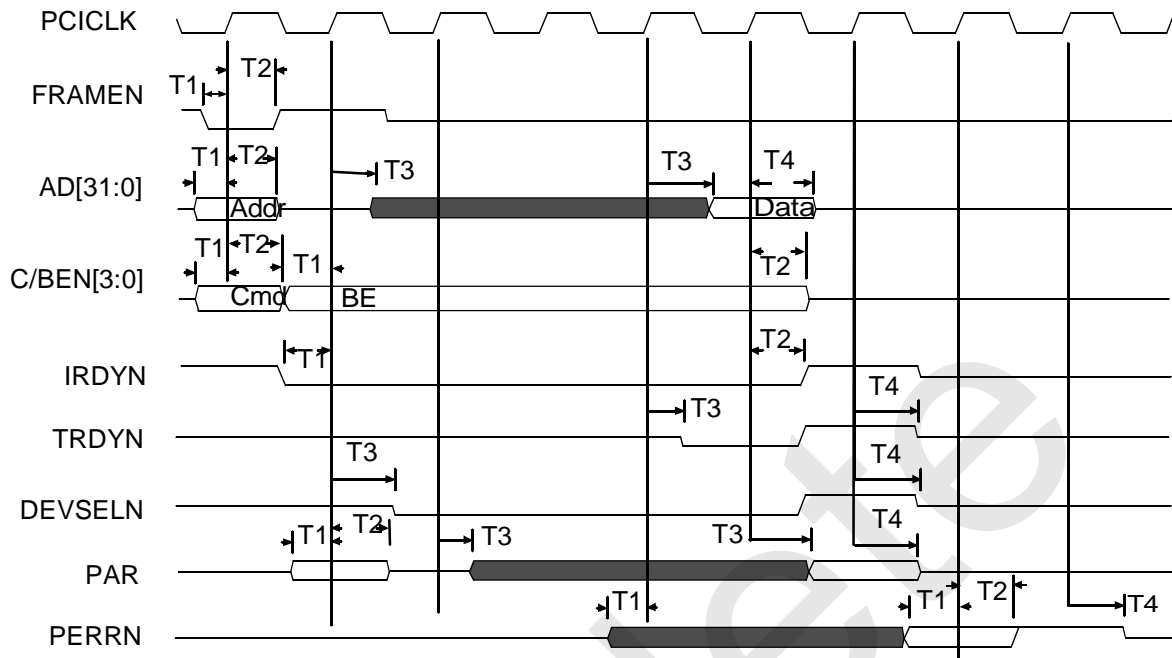
5.0 DC and AC Specifications (Continued)

PCI Bus Master Write

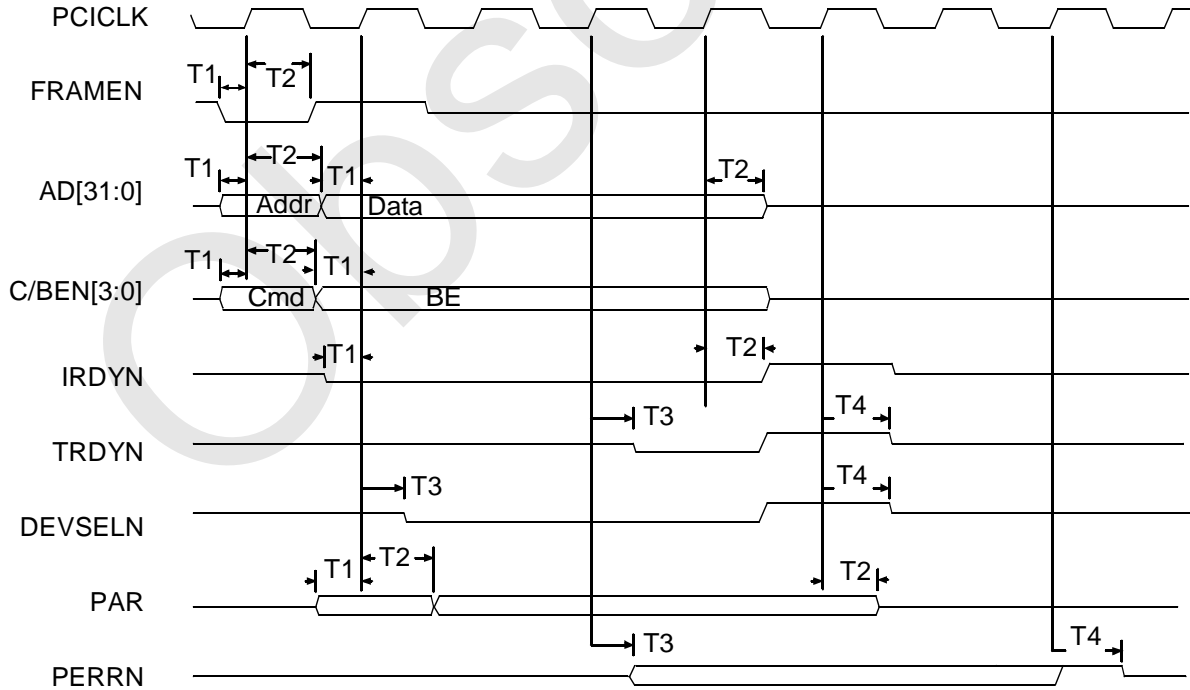


5.0 DC and AC Specifications (Continued)

PCI Target Read

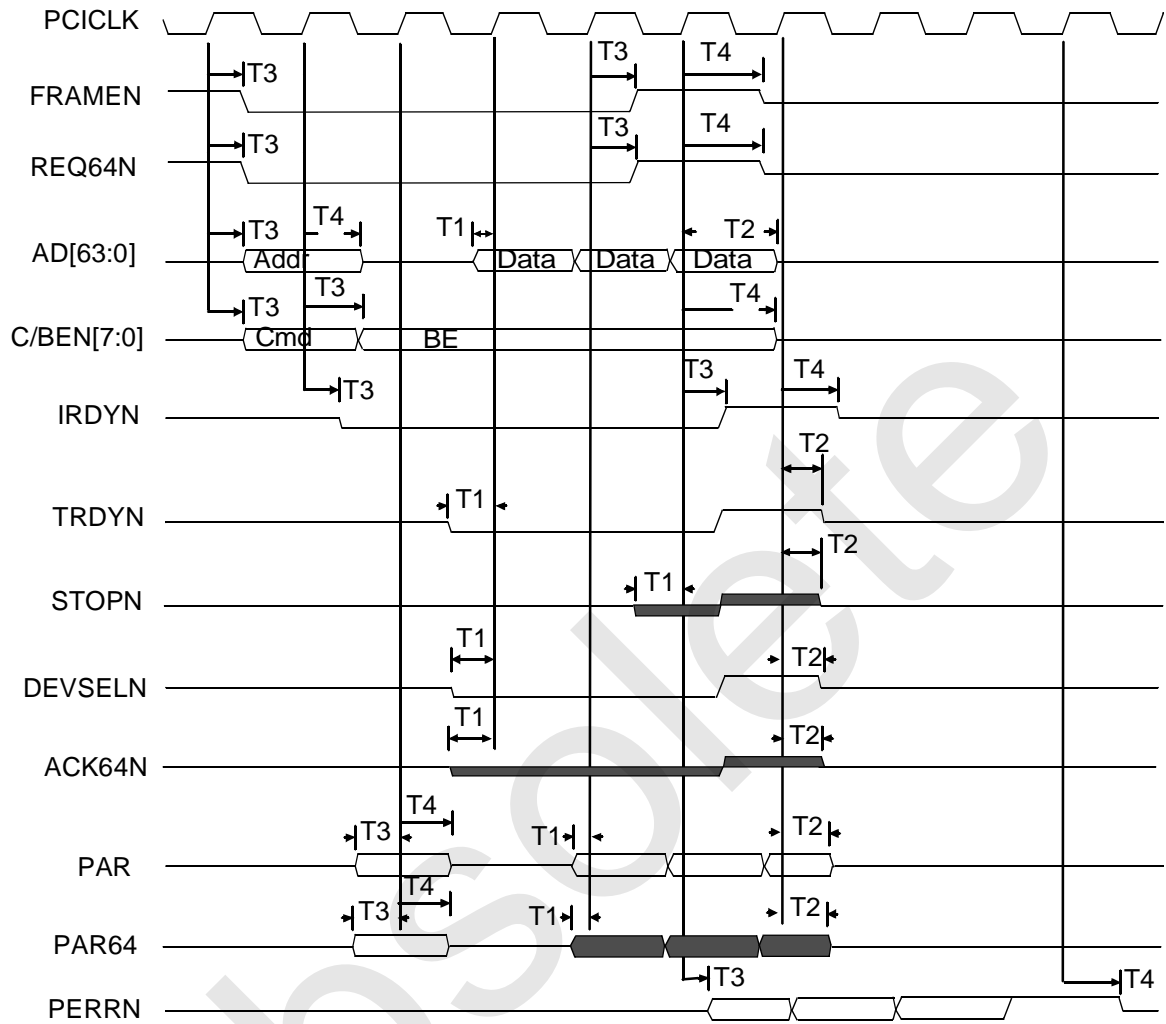


PCI Target Write



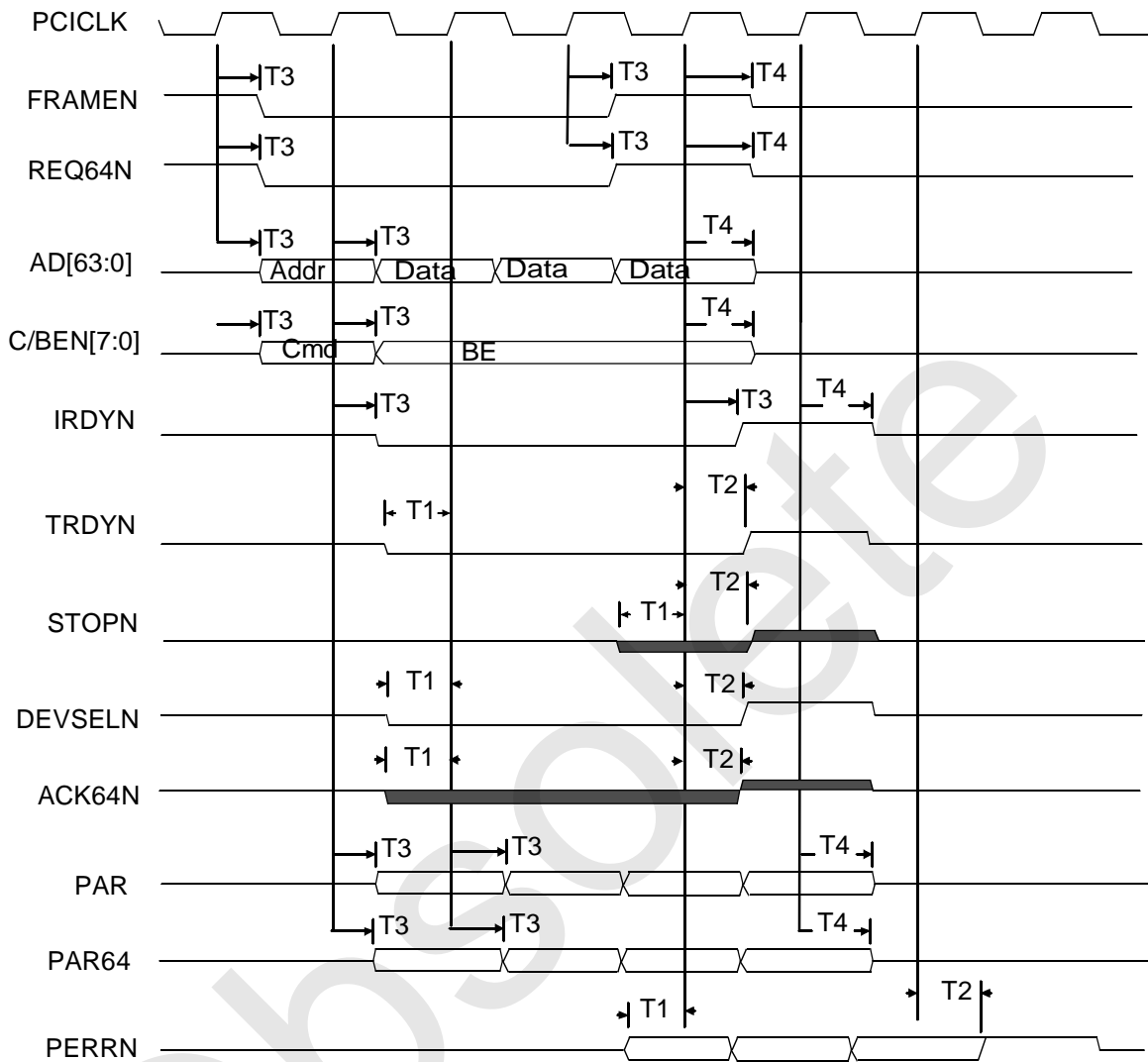
5.0 DC and AC Specifications (Continued)

PCI Bus Master Burst Read

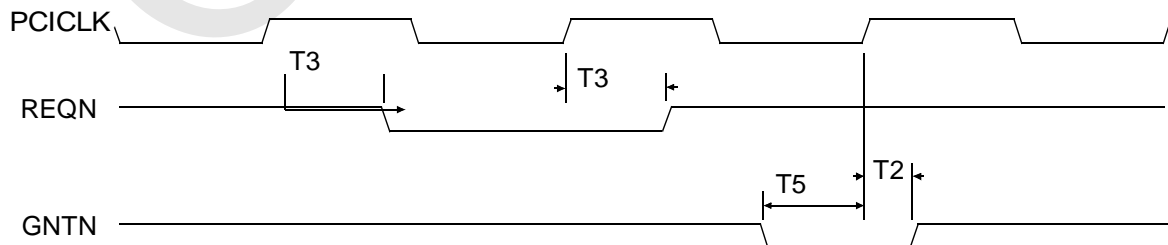


5.0 DC and AC Specifications (Continued)

PCI Bus Master Burst Write

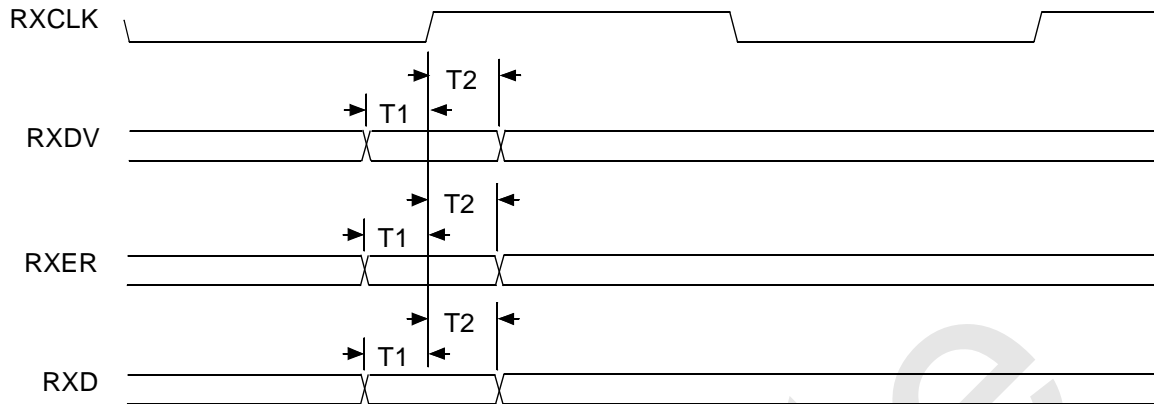


PCI Bus Arbitration



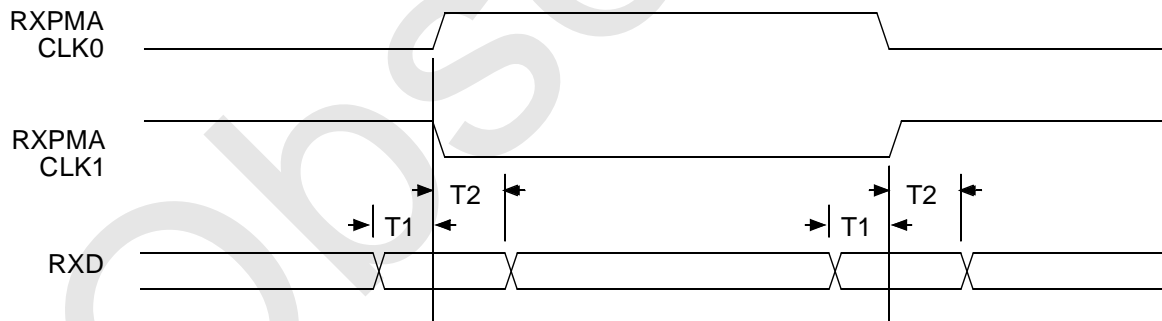
5.0 DC and AC Specifications (Continued)

5.2.7 RX MII/GMII Interface



| Number | Parameter | Min | Typ | Units |
|---------|--|-----|-----|-------|
| 5.2.7.1 | RXDV/RXER/RXD to RXCLK Setup Requirement | 2.0 | | ns |
| 5.2.7.2 | RXDV/RXER/RXD to RXCLK Hold Requirement | 0.0 | | ns |

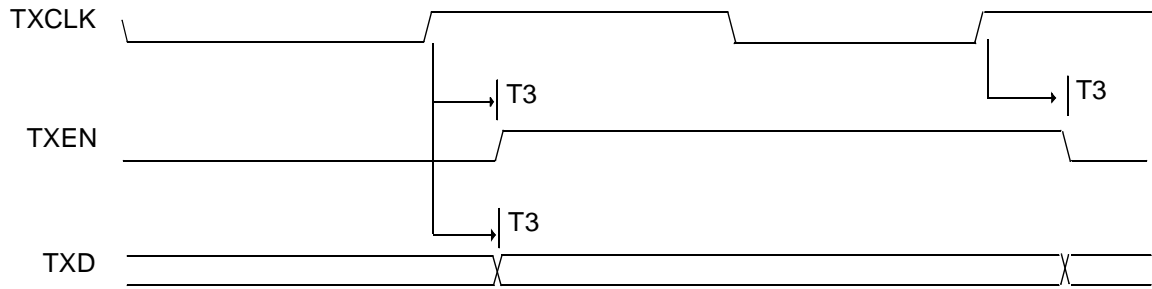
5.2.8 RX TBI Interface



| Number | Parameter | Min | Typ | Units |
|---------|--|-----|-----|-------|
| 5.2.8.1 | RXD[9:0] to RXPMACLK0 or RXPMACLK1 Setup Requirement | 2.5 | | ns |
| 5.2.8.2 | RXD[9:0] to RXPMACLK0 or RXPMACLK1 Hold Requirement | 1.5 | | ns |

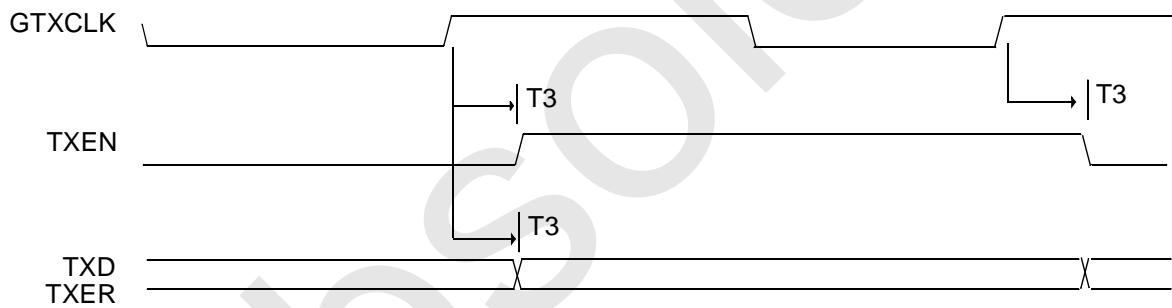
5.0 DC and AC Specifications (Continued)

5.2.9 TX MII Interface



| Number | Parameter | Min | Typ | Units |
|---------|----------------------------------|-----|-----|-------|
| 5.2.9.1 | TXEN/TXD Output Delay from TXCLK | 2 | 12 | ns |

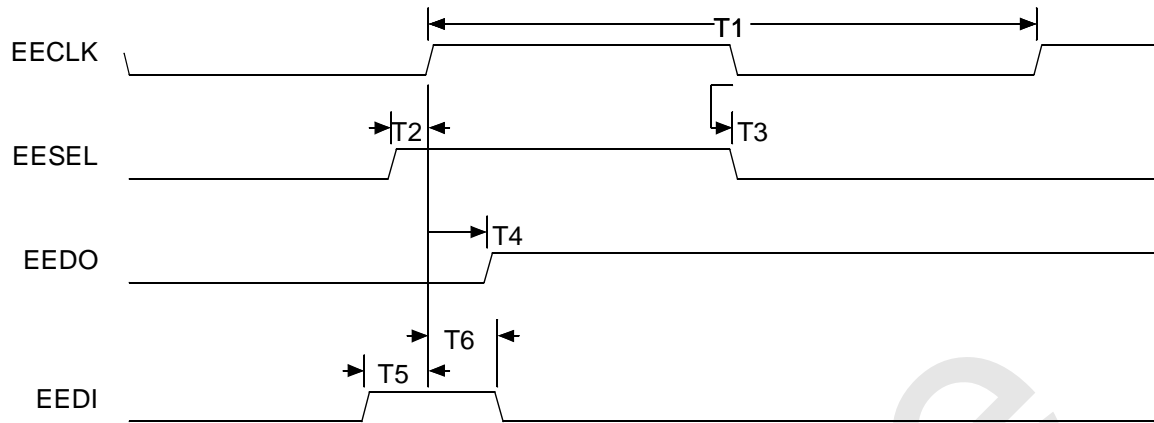
5.2.10 TX GMII/TBI Interface



| Number | Parameter | Min | Typ | Units |
|----------|--|-----|-----|-------|
| 5.2.10.1 | TXEN/TXER/TXD Output Delay from GTXCLK | 0.5 | 5.5 | ns |

5.0 DC and AC Specifications (Continued)

5.2.11 EEPROM Auto-Load

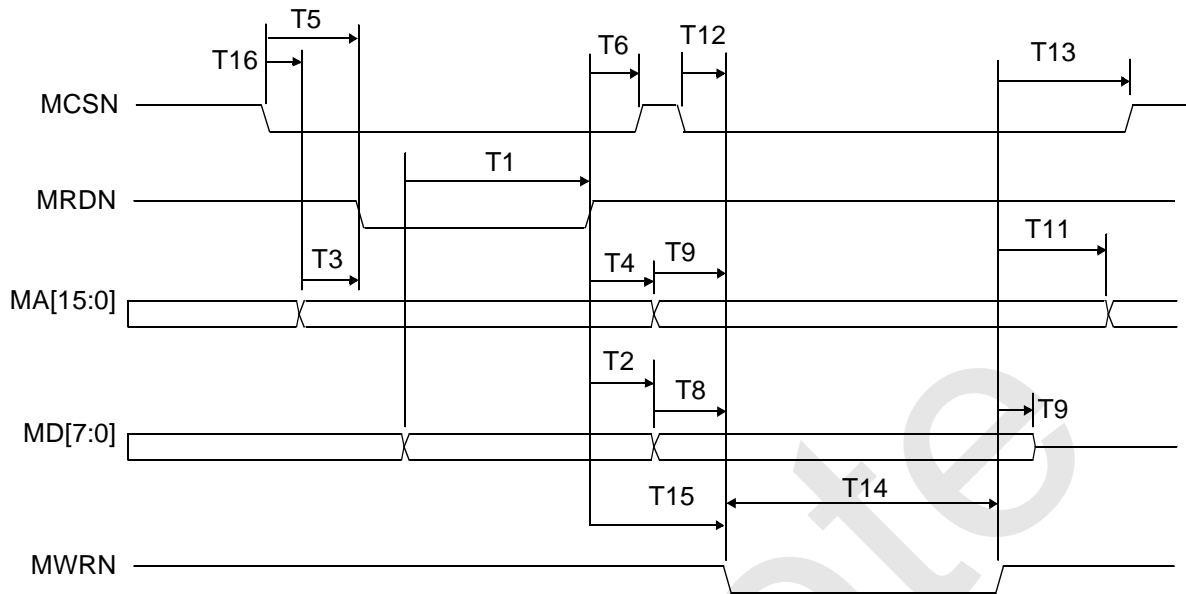


Refer to NM93C06 data sheet

| Number | Parameter | Min | Typ | Units |
|----------|----------------------------|-----|------|-------|
| 5.2.11.1 | EECLK Cycle Time | 5 | | us |
| 5.2.11.2 | EECLK Delay from EESEL | 1 | | us |
| 5.2.11.3 | EECLK Low to EESEL Invalid | 2 | | us |
| 5.2.11.4 | EECLK to EEDO Valid | | 3500 | us |
| 5.2.11.5 | EEDI Setup Time to EECLK | 2 | | us |
| 5.2.11.6 | EEDI Hold Time from EECLK | 3 | | us |
| 5.2.11.7 | EE Config load duration | | 2000 | us |

5.0 DC and AC Specifications (Continued)

5.2.12 Boot PROM/FLASH

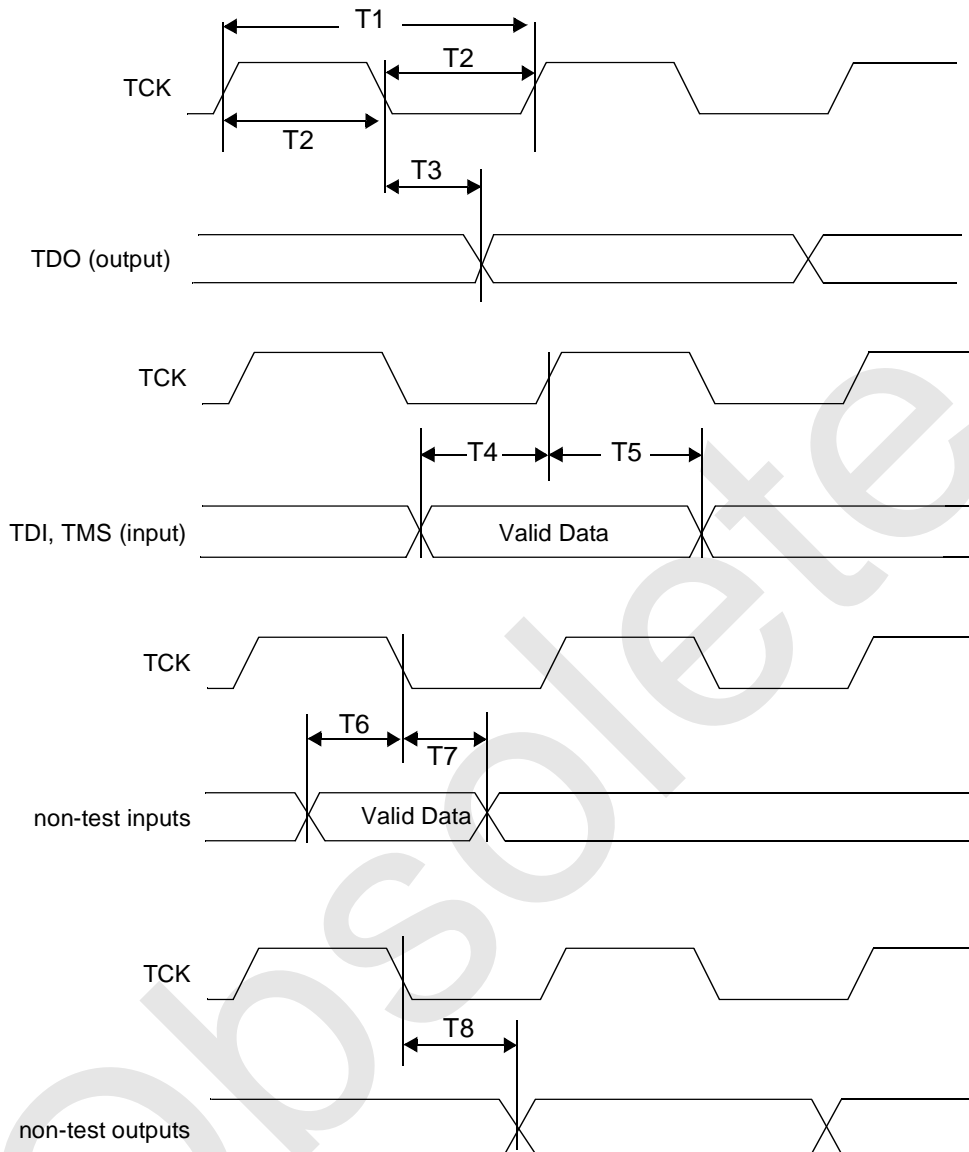


| Number | Parameter | Min | Typ | Units |
|-----------|-----------------------------------|-----|-----|-------|
| 5.2.12.1 | Data Valid to MRDN Invalid | 15 | | ns |
| 5.2.12.2 | Data Invalid from MRDN Invalid | 0 | | ns |
| 5.2.12.3 | Address Valid to MRDN Valid | | 15 | ns |
| 5.2.12.4 | Address Invalid from MRDN Invalid | | 0 | ns |
| 5.2.12.5 | MCSN Valid to MRDN Valid | | 15 | ns |
| 5.2.12.6 | MRDN Invalid to MCSN Invalid | | 0 | ns |
| 5.2.12.7 | MRDN Pulse Width | | 165 | ns |
| 5.2.12.8 | Data Valid to MWRN Valid | | 15 | ns |
| 5.2.12.9 | Data Invalid from MWRN Invalid | | 30 | ns |
| 5.2.12.10 | Address Valid to MWRN Valid | | 15 | ns |
| 5.2.12.11 | Address Invalid from MWRN Invalid | | 15 | ns |
| 5.2.12.12 | MCSN Valid to MWRN Valid | | 15 | ns |
| 5.2.12.13 | MWRN Invalid to MCSN Invalid | | 15 | ns |
| 5.2.12.14 | MWRN Pulse Width | | 150 | ns |
| 5.2.12.15 | MRDN Invalid to MWRN Valid | 165 | | ns |
| 5.2.12.16 | MCSN Valid to address Valid | | 0 | ns |

Note 5: T15 is guaranteed by design.

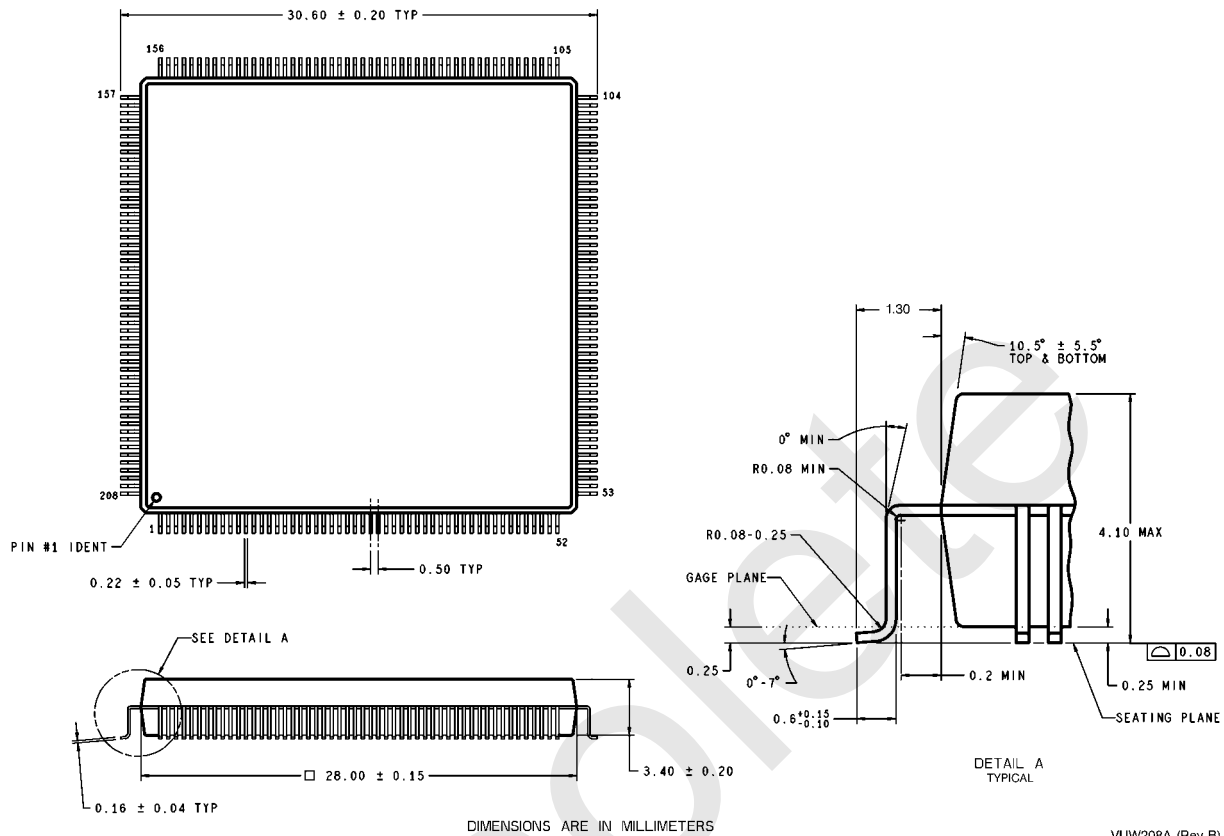
5.0 DC and AC Specifications (Continued)

5.2.13 JTAG Timing



| Number | Parameter | Min | Typ | Units |
|----------|-------------------------------------|-----|-----|-------|
| 5.2.13.1 | TCK Period | 100 | | ns |
| 5.2.13.2 | TCK low/high time | 40 | | ns |
| 5.2.13.3 | TCK to TDO (Output) Delay Time | 0 | 15 | ns |
| 5.2.13.4 | TDI, TMS (Input) to TCK Setup Time | 10 | | ns |
| 5.2.13.5 | TDI, TMS (Input) from TCK Hold Time | 10 | | ns |
| 5.2.13.6 | non-test input Setup time to TCK | 10 | | ns |
| 5.2.13.7 | non-test input Hold time from TCK | 10 | | ns |
| 5.2.13.8 | TCK to non-test outputs Delay Time | | 15 | ns |

Package Dimensions inches (millimeters) unless otherwise noted



VUW208A (Rev B)

Order Number DP83820VUW
See NS Package Number NVUW208A

LIFE SUPPORT POLICY

NATIONAL'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF THE PRESIDENT AND GENERAL COUNSEL OF NATIONAL SEMICONDUCTOR CORPORATION. As used herein:

1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into the body, or (b) support or sustain life, and whose failure to perform, when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury to the user.
2. A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.



National Semiconductor Corporation
 Tel: 1-800-272-9959
 Fax: 1-800-737-7018
 Email: support@nsc.com

National Semiconductor Europe
 Deutsch Tel: +49 (0) 69 9508 6208
 Fax: +49 (0) 180 530 85 86
 English Tel: +44 (0) 87 0240 2171
 Francais Tel: +33 (0) 1 41 91 8790
 Email: europe.support@nsc.com

National Semiconductor Asia Pacific Customer Response Group
 Tel: 65-254-4466
 Fax: 65-250-4466
 Email: ap.support@nsc.com

National Semiconductor Japan Ltd.
 Tel: 81-3-5639-7560
 Fax: 81-3-5639-7507

www.national.com

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third-party products or services does not constitute a license from TI to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of TI information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

Resale of TI products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

TI products are not authorized for use in safety-critical applications (such as life support) where a failure of the TI product would reasonably be expected to cause severe personal injury or death, unless officers of the parties have executed an agreement specifically governing such use. Buyers represent that they have all necessary expertise in the safety and regulatory ramifications of their applications, and acknowledge and agree that they are solely responsible for all legal, regulatory and safety-related requirements concerning their products and any use of TI products in such safety-critical applications, notwithstanding any applications-related information or support that may be provided by TI. Further, Buyers must fully indemnify TI and its representatives against any damages arising out of the use of TI products in such safety-critical applications.

TI products are neither designed nor intended for use in military/aerospace applications or environments unless the TI products are specifically designated by TI as military-grade or "enhanced plastic." Only products designated by TI as military-grade meet military specifications. Buyers acknowledge and agree that any such use of TI products which TI has not designated as military-grade is solely at the Buyer's risk, and that they are solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI products are neither designed nor intended for use in automotive applications or environments unless the specific TI products are designated by TI as compliant with ISO/TS 16949 requirements. Buyers acknowledge and agree that, if they use any non-designated products in automotive applications, TI will not be responsible for any failure to meet such requirements.

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

Products

| | |
|------------------------|--|
| Audio | www.ti.com/audio |
| Amplifiers | amplifier.ti.com |
| Data Converters | dataconverter.ti.com |
| DLP® Products | www.dlp.com |
| DSP | dsp.ti.com |
| Clocks and Timers | www.ti.com/clocks |
| Interface | interface.ti.com |
| Logic | logic.ti.com |
| Power Mgmt | power.ti.com |
| Microcontrollers | microcontroller.ti.com |
| RFID | www.ti-rfid.com |
| OMAP Mobile Processors | www.ti.com/omap |
| Wireless Connectivity | www.ti.com/wirelessconnectivity |

Applications

| | |
|-------------------------------|--|
| Communications and Telecom | www.ti.com/communications |
| Computers and Peripherals | www.ti.com/computers |
| Consumer Electronics | www.ti.com/consumer-apps |
| Energy and Lighting | www.ti.com/energy |
| Industrial | www.ti.com/industrial |
| Medical | www.ti.com/medical |
| Security | www.ti.com/security |
| Space, Avionics and Defense | www.ti.com/space-avionics-defense |
| Transportation and Automotive | www.ti.com/automotive |
| Video and Imaging | www.ti.com/video |

TI E2E Community Home Page

e2e.ti.com

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265
Copyright © 2011, Texas Instruments Incorporated