
# L64381 4-Port Ethernet Controller Device Technical Manual

This document is preliminary. As such, it contains data derived from functional simulations and performance estimates. LSI Logic has not verified either the functional descriptions, or the electrical and mechanical specifications using production parts.

Document DB14-000002-01, Second Edition (August 1996)
This document describes revision B of LSI Logic Corporation's L64381 4-Port Ethernet Controller (Quad CASCADE®) Device and will remain the official reference source for all revisions/releases of this product until rescinded by an update.

**To receive product literature, call us at 1.800.574.4286 (U.S. and Canada); +32.11.300.531 (Europe); 408.433.7700 (outside U.S., Canada, and Europe) and ask for Department JDS; or visit us at http://www.lsilogic.com.**

# Preface

This book is the primary reference and technical manual for the L64381 4-Port Ethernet Controller (Quad CASCADE) Device. It contains a complete functional description for the L64381 and includes complete physical and electrical specifications for the L64381.

## Audience

This document assumes that you have some familiarity with networking, microprocessors, and related support devices. The people who benefit from this book are:

- Engineers and managers who are evaluating the L64381 for possible use in a networking system
- Engineers who are designing the L64381 into a system

## Organization

This document has the following chapters and appendixes:

- Chapter 1, **Introduction**, describes the general characteristics and capabilities of the L64381 product.
- Chapter 2, **Registers**, defines the on-chip registers.
- Chapter 3, **Signal Descriptions**, describes the input, output, and bidirectional signals of the L64381 chip.
- Chapter 4, **L64381 Operation**, provides a detailed description of the functional blocks and interfaces within the L64381 chip.
- Chapter 5, **Statistics Counters**, lists the purpose and contents of the 32-bit Statistics Counters.
- Chapter 6, **Instruction Set**, defines the L64381 instruction set and provides each instruction's valid modes.
- Chapter 7, **Specifications**, provides the AC and DC specifications and packaging information for the L64381 chip.

- Appendix A, **Customer Feedback**, includes a form that you may use to fax us your comments about this document.

**Related Publications**

*Compacted and Scalable Dedicated Ethernet (CASCADE) Core Technical Manual,* Order No. C14015

**Conventions Used in This Manual**

The first time a word or phrase is defined in this manual, it is *italicized.*

The following signal naming conventions are used throughout this manual:

- A level-significant signal that is true or valid when the signal is LOW always has an overbar ( $\overline{\quad}$ ) over its name.
- An edge-significant signal that initiates actions on a HIGH-to-LOW transition always has an overbar ( $\overline{\quad}$ ) over its name.

The word *assert* means to drive a signal true or active. The word *deassert* means to drive a signal false or inactive.

Hexadecimal numbers are indicated by the prefix “0x” before the number—for example, 0x32CF. Binary numbers are indicated by a subscripted “2” following the number—for example, $0011.0010.1100.1111_2$.

A *word* is 32-bits long. A *halfword* is 16-bits long. A *byte* is 8-bits long.

# Contents

---

**Tables**

# Chapter 1
# Introduction

This document describes the L64381 (Quad CASCADE) 4-Port Ethernet Controller device, a highly integrated CMOS solution for Local Area Network (LAN) applications. The L64381 is derived from the 10-Mbps Ethernet Controller core. This multiport solution allows system designers to build cost-effective switched Ethernet hub or router products.

The L64381 consists of four ports, each with separate 128-byte Transmit and Receive FIFOs. The device architecture includes a high-performance 1-Gbps bus interface designed for fast packet transfers, and a processor interface for control and status checking purposes. The L64381 can be programmed for full-duplex or half-duplex operation, has individual, and promiscuous addressing modes, and supports both twisted-pair and AUI (Attachment Unit Interface) interfaces.

The L64381 has met all IEEE 802.3 signal conformance and interoperability tests.

Chapter 1 provides an overview of the L64381. It contains two sections:

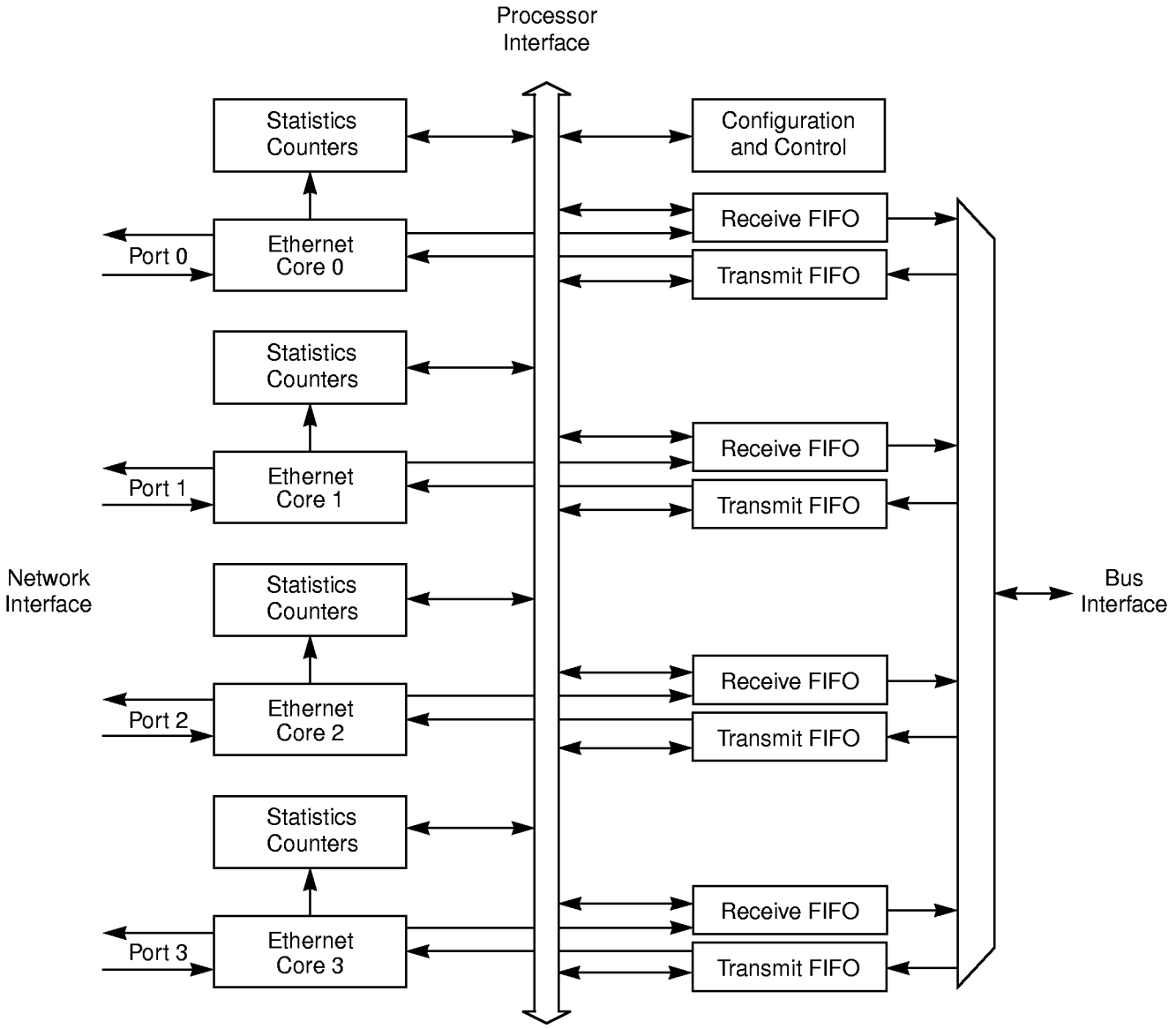
- Section 1.1, “Overview”
- Section 1.2, “Key Features”

## 1.1 Overview

Figure 1.1 shows a block diagram of the L64381 device. The four Ethernet controllers are fully independent. Each L64381 port has its own MAC (Medium Access Control), Manchester Encoder/Decoder, and twisted-pair/AUI transceivers. The device is fully synchronous. It operates on a 16-MHz to 33-MHz host clock and a 20-MHz network clock. The 128-byte FIFOs are configurable for optimizing latency and throughput.

Each port has 23 Statistical Counters, which provide on-chip maintenance. These counters meet the statistics requirements of the Simple Network Management Protocol (SNMP).

Figure 1.1
L64381 Block Diagram




The chip provides several programmable configurations to allow customer product differentiation, such as source address insertion, FCS insertion, immediate retransmission on collision, bus transfer size, and transmit buffer threshold size.

The L64381 can be configured using the 6-bit Address bus and the 16-bit PData bus. All the internal registers can be read, and most also can be written. Special test modes are provided that make functional and in system testing simpler.

The Bus Interface is designed to allow efficient transfers of packet data to and from the L64381. Four bytes of data can be transferred per cycle on the bus with a burst size of 4, 8, or 16 words. One dead cycle is required between each burst cycle.

The scan interface allows for board-level testing using the JTAG (IEEE 1149.1) standard.

LSI Logic fabricates the L64381 in the CMOS cell-based process. The L64381 is available in a 208-pin Plastic Quad Flat Pack (PQFP) package.

---

## 1.2 Key Features

This section lists the general features of the L64381.

### 1.2.1 Ethernet Core

The Ethernet core features are categorized into four subsections: MAC, Encoder/Decoder, twisted-pair transceivers, and AUI transceivers.

#### 1.2.1.1 MAC Features

- IEEE 802.3 Ethernet Standards support
- Complete CSMA/CD Medium Access Control functions
- Twisted-pair, full-duplex mode for 20 Mbit/s throughput
- Individual address filter (individual and promiscuous addressing modes)
- Internal FCS generator and checker

#### 1.2.1.2 Encoder/Decoder

- 10 Mbit/s Manchester encoding/decoding
- Digital PLL with synchronous data recovery
- Manchester data decoded with 18-ns jitter
- PLL with fast-lock time
- Elasticity:
  - Maximum time: 600 ns
  - Maximum clock error @ 1518 byte frame: 0.05%
  - Maximum clock error @ 4500 byte frame: 0.016%

#### 1.2.1.3 Twisted-Pair (10BASE-T) Transceivers

- Integrated transceiver interface
- Collision detect
- Link integrity test
- Loopback test capabilities
- Autopolarity detection and correction

#### 1.2.1.4 Attachment Unit Interface (AUI) Transceivers

- Signal Quality Error test (Heartbeat)
- Smart receive squelch for receive data
- Smart SQE squelch for collision detection
- 10BASE-2, 10BASE-5, or 10BASE-F interface

### 1.2.2 Internal Architecture

This subsection highlights the features of the internal architecture:

- Independent 128-Byte Transmit and Receive FIFOs per Ethernet port
- Autotransmit on collision feature. First 64 transmit bytes buffered on chip to allow retransmission on collision without requiring bus transactions for first 64 bytes of packet
- Ability to generate or suppress FCS
- Next packet to transmit is queued while previous packet is still being transmitted, thus providing multiple back-to-back packet transmissions without wasted bandwidth
- Statistics per Ethernet core maintained on chip
- All registers are fully observable, most are controllable

### 1.2.3 Bus Interface

This subsection summarizes the key features of the L64381 bus interface:

- 32-bit data bus width
- Up to 48 ports (12 devices) at 20 Mbit/s can be connected based on approximately 1-Gbit bandwidth

- $\overline{\text{PKT\_AVAIL}}$ signal informs system when the Receive FIFO has more data than the programmable bus transfer size, or that an End-of-Packet ($\overline{\text{EOP}}$) signal is present in the Receive FIFO, so that the FIFO data can be read by the system
- $\overline{\text{BUF\_AVAIL}}$ signal informs system when the Transmit FIFO has more space than the programmable bus transfer size, so that the system can write transmit data to the FIFO
- Four $\overline{\text{BYTE\_VALID}}$ signals specify which bytes are valid in the current data word (significant only for the first and last transfer in a packet when either the start-of-packet signal $\overline{\text{SOP}}$ or the end-of-packet signal $\overline{\text{EOP}}$ is asserted)
- $\overline{\text{SOP}}$ and $\overline{\text{EOP}}$ signals delimit start of packet and end of packet, respectively
- $\overline{\text{PORT\_BUSY}}$ signal per Ethernet core when operating in full-duplex or half-duplex mode

### 1.2.4 Processor Interface

This subsection highlights the features of the L64381 processor interface:

- 6-bit Address and 16-bit PData buses
- Slave mode device
- Chip Control and Status information can be read from Control and Status Registers
- Burst mode reads of Ethernet Statistics
- Interrupt signal to inform the processor if an error condition occurs

### 1.2.5 Key Programmable Features

The L64381 has several key programmable features:

- Individual twisted-pair or AUI assignment per port
- Automatic source address insertion
- Automatic FCS insertion
- Bus transfer size is configurable to 4, 8, or 16 words
- Addressing modes (individual, promiscuous, or multicast)
- Auto padding of packets less than 64 bytes in length
- Programmable little-endian and big-endian addressing modes

# Chapter 2
# Registers

The L64381 has a number of on-chip registers that configure and control it. In addition, all the FIFO pointers, FIFO Address and Data registers, and Statistics Counters are decoded as register addresses, so that the Processor Interface can read and write their values. This chapter describes the function of the registers with their associated bit fields.

This chapter has the following sections:

## 2.1 Register Addresses

Table 2.1 summarizes the addresses of the L64381 registers. The ADRS[5:0] inputs determine which register is being accessed.

Table 2.1
L64381 Register Summary

| ADRS[5:0] | Accessibility | Register |
|---|---|---|
| 000010 | R/W[1] | Configuration Register 0 |
| 000011 | R/W | Configuration Register 1 |
| 000100 | R/W | Ethernet Address Register 0 |
| 000101 | R/W | Ethernet Address Register 1 |
| 000110 | R/W | Ethernet Address Register 2 |
| 000111 | R/W | Packet Configuration Register |
| 001000 | R/W | Error Register 0 |
| 001010 | R/W | Error Register 1 |
| 001001 | R/W | Error Mask Register 0 |
| 001011 | R/W | Error Mask Register 1 |
| 010000 | R/W | Statistics Counters Data In Register 0 |
| 010001 | R/W | Statistics Counters Data In Register 1 |
| 010010 | R/O[2] | Statistics Counters Data Out Register 0 |
| 010011 | R/O | Statistics Counters Data Out Register 1 |
| 010110 | R/W | Statistics Counters Address Register |
| 011000 | R/W | Data FIFO Data In Register 0 |
| 011001 | R/W | Data FIFO Data In Register 1 |
| 011010 | R/O | Data FIFO Data Out Register 0 |
| 011011 | R/O | Data FIFO Data Out Register 1 |
| 011110 | R/W | Data FIFO Address Register |
| 100000 | R/W | Receive FIFO Head Pointer Register 0 |
| 100001 | R/W | Receive FIFO Head Pointer Register 1 |
| 100010 | R/W | Receive FIFO Tail Pointer Register 0 |
| 100011 | R/W | Receive FIFO Tail Pointer Register 1 |
| 100100 | R/W | Transmit FIFO Head Pointer Register 0 |
| 100101 | R/W | Transmit FIFO Head Pointer Register 1 |
| 100110 | R/W | Transmit FIFO Tail Pointer Register 0 |
| 100111 | R/W | Transmit FIFO Tail Pointer Register 1 |
| 101000 | R/W | Receive EOP Counter 0 |
| 101001 | R/W | Receive EOP Counter 1 |
| 101010 | R/W | Transmit EOP Counter 0 |
| 101011 | R/W | Transmit EOP Counter 1 |

1. Read/Write.
2. Read Only.

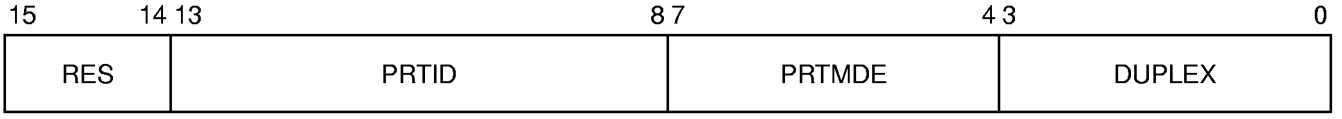
## 2.2 Configuration Registers

There are three configuration registers: Configuration Register 0, Configuration Register 1, and the Packet Configuration Register. These registers initialize the overall L64381. In the following subsections, the letter 'x' is used to imply any one of the four ports. Bit assignments are made in ascending order—Port 0 is assigned the lowest bit, and Port 3 is assigned the highest bit.

### 2.2.1 Configuration Register 0

Figure 2.1 shows the bit assignments for Configuration Register 0. This 16-bit register is read/write.

Figure 2.1
Configuration Register 0




**RES** **Reserved** **[15:14]**
These bits are reserved and read as zeroes.

**PRTID** **PORT ID** **[13:8]**
These six bits assign unique, sequential ID numbers to the ports. This field is useful for applications with multiple L64381 devices. PORT 0 is assigned an eight-bit number, which is the PORT ID field appended with $00_2$, PORT 1 is assigned the value {PORT ID || $01_2$}, PORT 2 is assigned the number {PORT ID || $10_2$}, and PORT 3 is assigned the number {PORT ID || $11_2$}.

The SOPCFG bit must be set to one for this field to be meaningful. Refer to the description of SOPCFG in Section 2.2.2, "Configuration Register 1."

**PRTMDE** **PORT x Mode** **[7:4]**
These four bits configure the corresponding port to receive all packets that are sent on the Ethernet. (Bit 4 corresponds to Port 0, Bit 5 corresponds to Port 1, etc.) If one of these bits is reset to zero (Port Mode is OFF), then the corresponding port receives only the packets that exactly match the Ethernet address specified in the

Ethernet Address Registers. No broadcast or multicast packets are accepted in the OFF mode.

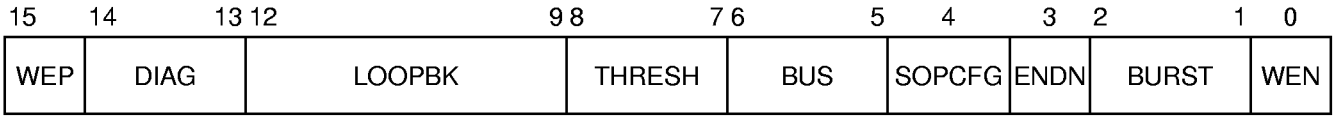
If one of these bits is set to one (Port Mode is ON), then the corresponding port receives all packets that are sent on the Ethernet, regardless of the address specified in the Ethernet Address Registers. The ports also receive all broadcast and multicast packets in the ON mode.

**DUPLEX** **PORT x Duplex Mode** **[3:0]**
These four bits configure the corresponding Ethernet ports to half-duplex or full-duplex mode (0 = Half Duplex; 1 = Full Duplex). Bit 0 configures Port 0, Bit 1 configures Port 1, etc.

### 2.2.2 Configuration Register 1

Figure 2.2 shows the bit assignments for Configuration Register 1. This 16-bit register is read/write. To modify any of these bits, the processor should perform a read-modify-write operation in order to retain the values of the non-affected bits in this register.

Figure 2.2
Configuration Register 1




**WEP** **Write Enable Port** **15**
Setting this bit to one enables writing to the four bits in the ENPORTx field in the Packet Configuration Register. The ENPORTx field enables/disables the four ports.

**DIAG** **Diagnostic Modes** **[14:13]**
These bits determine the type of mode in which the L64381 operates.

| DIAG | Mode | Description |
|---|---|---|
| 0, 2 | Normal | The L64381 is in normal operating mode (active) as described in this document. |
| 1 | R/W Register | In this mode, the L64381 is halted. This mode is used for reading and writing values into the FIFO Pointer Registers, the Data FIFO, and the Statistics Counters. This mode is provided for testing the Transmit and Receive FIFO Pointer registers and the Data FIFO and Counters. In normal mode, it is not possible for the processor to directly write data into the FIFO Pointer registers or the Data FIFO. |
| 3 | FIFO and Counter Test | This mode puts the FIFO address register and the Statistics Counter address counter into an increment mode. An initial value is loaded into the address register, and then the address increments itself on every read and write operation to the FIFO or counter. This mode is useful for testing the Data FIFO and Statistics Counters, as well as initializing them. |

**LOOPBK** **Per Port Loopback Mode** **[12:9]**

When one of these bits is set to one, its corresponding port operates in local loopback mode. In this mode, the transmitter output is fed back into the receiver input, thus allowing local testing without requiring an external loopback connector. The following table shows the relationship between the LOOPBK register bits and their corresponding ports. This mode applies to full-duplex configuration only.

| LOOPBK Bit | Port |
|---|---|
| 12 | 3 |
| 11 | 2 |
| 10 | 1 |
| 9 | 0 |

**THRESH** **Transmit Threshold** **[8:7]**

These two bits determine the number of words that need to be stored in the Transmit FIFO before it initiates a packet transmit on the Ethernet.

| TT | # of Stored Words |
|---|---|
| 00 | 4 words |
| 01 | 8 words |
| 10 | 16 words |
| 11 | 16 words |

**BUS** **Bus Transfer Size** **[6:5]**

These bits specify the maximum number of words that the L64381 transfers across the Bus Interface whenever the L64381 asserts either $\overline{\text{PKT\_AVAIL}}$ or $\overline{\text{BUF\_AVAIL}}$ in response to the assertion of $\overline{\text{STROBE}}$.

| BUS | Transfer Size |
|---|---|
| 00 | 4 words |
| 01 | 8 words |
| 10 | 16 words |
| 11 | 16 words |

Note: that to externally control the length of the transfer, the system could deassert either $\overline{\text{READ\_OUT\_PKT}}$ or $\overline{\text{WRITE\_IN\_PKT}}$ before the BUS amount of words is transferred across the Bus Interface. However, $\overline{\text{READ\_OUT\_PKT}}$ or $\overline{\text{WRITE\_IN\_PKT}}$ should not be kept asserted after the BUS amount of words has been transferred, because the L64381 does not guarantee the response. If an $\overline{\text{EOP}}$ signal is asserted in the middle of a transfer, the L64381 drives idle data (zeros) until it detects the deassertion of $\overline{\text{READ\_OUT\_PKT}}$. If the L64381 detects an EOP while $\overline{\text{WRITE\_IN\_PKT}}$ is asserted, then it stops writing data into the Transmit FIFO after that cycle.

**SOPCFG** **Start of Packet Configuration** **4**

When this bit is set to one, the L64381 assumes that the first word in any packet that the host writes into its Transmit FIFO contains Packet Configuration Information. In this case, the least-significant four bits in the Packet Configuration Register are ignored and are replaced with the four least-significant bits from the first word in the packet. Therefore, each port is configured on a per-packet basis. In addition, when the L64381 receives packets from the Ethernet, it appends one extra word containing the PORT ID number in front of the packet so that the host knows which port the packet is coming from, and can insert the Packet Configuration Information into this space later. (The PORT ID number is placed at the least-significant byte of the first word.)

If SOPCFG is not set, then the Packet Configuration Information bits of the Packet Configuration Register are used as the Packet Configuration Information for all packets that are transmitted by the L64381. In this case, the PORT ID number is not used. Refer to Figure 4.4 for $\overline{SOP}$ timing.

**ENDN** **Endian Addressing** **3**

When this bit is set to zero, the L64381 addressing mode on the Processor Interface is big endian (see Figure 2.3). When this bit is set to one, the addressing mode is little endian (see Figure 2.4). *Because the L64381 is not byte addressable, this bit only affects the Burst Read instruction.*

Figure 2.3
Big-Endian
Addressing

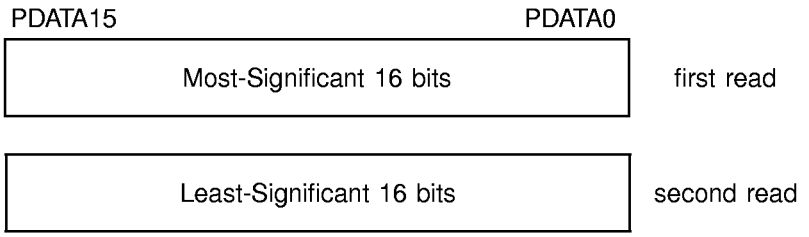
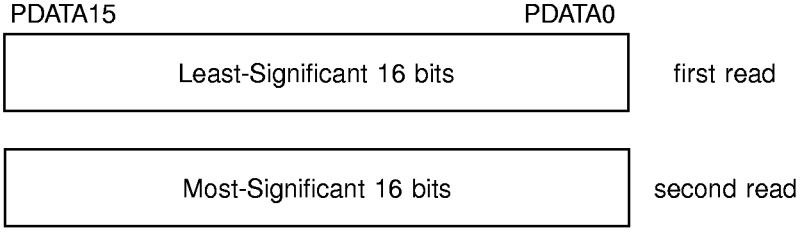



Figure 2.4
Little-Endian
Addressing



When accessing statistics from the Statistics Counters in big-endian and burst modes, the two most-significant bytes of the counter are placed on the PDATA bus before the two least-significant bytes. In little-endian and burst modes, the two least-significant bytes are placed on the PDATA bus before the two most-significant bytes.

The ENDN bit setting affects only the Processor Interface and has no effect on the little-endian Bus Interface. During a Bus Interface write transaction, the L64381 looks for the first valid byte from either DATA[7:0], DATA[15:8], DATA[23:16] or DATA[31:24]. This choice depends on the value of $\overline{\text{BYTE\_VALID}}$[3:0]. The L64381 then sends the valid byte to the network. In the first example below, DATA[7:0] goes to the network first, then DATA[15:8], and finally DATA[23:16]. The second example has only two valid bytes, so DATA[23:16] is passed to the network first, and DATA[31:24] follows.

| $\overline{\text{BYTE\_VALID}}$[3:0] | $1000_2$ | $0011_2$ |
|---|---|---|
| BIts[31:24] | Invalid Byte | Second Byte |
| Bits [23:16] | Third Byte | First Byte |
| Bits [15:8] | Second Byte | Invalid Byte |
| Bits [7:0] | First Byte | Invalid Byte |

**BURST** **Burst Mode Size** **[2:1]**

These bits define the number of burst transfers that the L64381 can perform on the Processor Interface in response to a burst mode instruction. The Burst Mode allows the processor to quickly read all the statistics collected by the L64381 per Ethernet port without having to execute individual read instructions per statistic.

| BURST[2:1] | Transfer Size |
|---|---|
| 00 | 2 16-bit transfers |
| 01 | 4 16-bit transfers |
| 10 | 8 16-bit transfers |
| 11 | 16 16-bit transfers |

Note : These bits refer only to the Processor Interface; the Bus Transfer Size (BUS) bits define the transfer size on the Bus Interface.

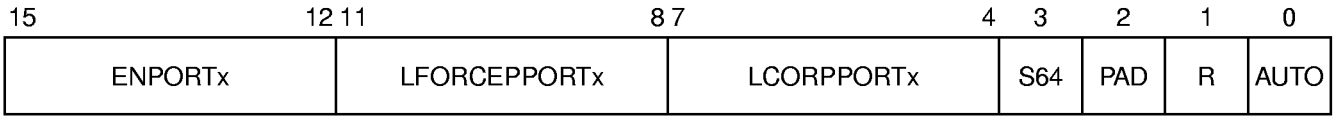
**WEN** **Write Enable Ethernet Address** **0**

Setting this bit to one enables writing to the six-byte Ethernet address of the ports. The user should reset this bit as soon as the address has been assigned in order to reduce the chances of an accidental overwrite of the address.

### 2.2.3 Packet Configuration Register

Figure 2.5 shows the Packet Configuration Register. This 16-bit read/write register contains programming information for packets and control bits that enable and disable the ports. The information stored in the least-significant four bits of this register are applied to all four ports on the L64381. To override the information in these four bits with packet specific information, set the SOPCFG bit in Configuration Register 1 to one. The most-significant four bits of this register enable the individual ports (1 = Enabled). In the disabled state, the ports will not respond to any internal or external stimuli.

Figure 2.5
Packet Configuration Register




**ENPORTx** **Enable Port x** **[15:12]**

When one of these bits is set, the associated port is enabled and it functions as specified in Configuration Registers 0 and 1. When one of these bits is reset, the associated port is reset (disabled), the associated port's outputs are negated, and the Head and Tail Pointers and EOP counters are reset to zeros. This field can only be written if the WEP bit in Configuration Register 1 is set to one.

**LFORCEPPORTx**

**Link Force Port x** **[11:8]**

When one of these bits is set, the associated port allows a transmit even if the link integrity fails. Refer to Chapter 2, "Signal Descriptions," in the *Compacted and Scalable Dedicated Ethernet (CASCADE) Core Technical Manual* for a description of the LFORCEP signal.

**LCORPPORTx** **Link Correct Polarity Port x** **[7:4]**
When one of these bits is set, the associated port corrects for inverted polarity on the twisted-pair interface, if an inverted polarity condition exists on that port. Refer to Chapter 2, "Signal Descriptions," in the *Compacted and Scalable Dedicated Ethernet (CASCADE) Core Technical Manual* for a description of the LCORPP signal.

**S64** **Store First 64 Bytes** **3**
When this bit is set, each port's Transmit FIFO stores the first 64 bytes of a packet until all bytes have been transmitted without collisions on the Network Interface. Once the first 64 bytes are transmitted, the FIFO releases the locations that were used to store these bytes, thereby allowing new data to be fetched into the FIFO. This feature allows the port to retransmit the packet without requiring extra transfers on the Bus Interface. This bit is valid only when the SOPCFG bit is zero.

**PAD** **Automatic Packet Padding** **2**
When this bit is set, the ports automatically pad a runt packet to a length of 64 bytes. This feature allows better utilization of the Transmit FIFO space, because all runt packets do not need to be pre-padded to the minimum Ethernet packet size. This bit has effect only when $\overline{\text{AI\_FCS\_IN}}$ is asserted and the SOPCFG bit is zero, or else the L64381 ignores this bit.

**R** **Reserved** **1**
This bit is reserved and reads as zero.

**AUTO** **Auto Insert Source Address** **0**
When this bit is set, the ports automatically insert the address stored in the Ethernet Address Registers into the source address field of the Ethernet packet. This bit is valid only when the SOPCFG bit is zero.

## 2.3 Ethernet Address Registers

The three Ethernet Address Registers (0 through 2) contain the 48-bit Ethernet address assigned to the L64381. Ethernet Address Register 0 contains the least-significant bytes of the Ethernet port address, and Ethernet Register 2 contains the most-significant bytes of the Ethernet address. These registers can be written only when the WEN bit in Configuration Register 1 is set to one.

Note: The user should immediately reset the write enable bits after the address has been written. This action reduces the likelihood of an inadvertent change of the address due to a software bug. When the Port Mode is ON, the ports accept all packets received on the Ethernet regardless of the value in these registers. When the Port Mode is OFF, only those packets that exactly match the address in these registers are accepted.

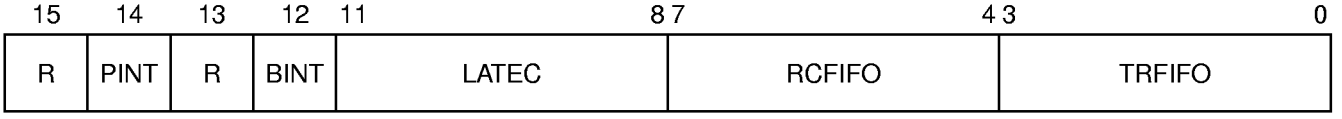
## 2.4 Error Registers

The L64381 has two Error Registers, Error Register 0 and Error Register 1. The fields within these registers indicate the status of all errors the L64381 can detect. Upon detection of an error, the L64381 asserts the $\overline{\text{INTERRUPT}}$ output. To clear the interrupt, take the one's complement of the Error Register that caused the interrupt and write the value back into that register.

### 2.4.1 Error Register 0

Figure 2.6 shows Error Register 0. Error Register 0 is a 16-bit read-write register. Upon power-up or when $\overline{\text{RESET}}$ is asserted, all bits in this register are cleared to zeros (all interrupts are disabled).

Figure 2.6
Error Register 0




**R** — **Reserved** — **15, 13**
These bits are reserved and read as zeros.

**PINT** — **Processor Interface Error** — **14**
A one on this bit indicates that a Processor Interface error occurred (for example, a write to an unwritable location).

**BINT** — **Bus Interface Error** — **12**
A one on this bit indicates that a Bus Interface error occurred. This error occurs when $\overline{\text{WRITE\_IN\_PKT}}$ and $\overline{\text{READ\_OUT\_PKT}}$ are both asserted at the same time.

**LATEC** **PORT x Late Collision Error** **[11:8]**
A one on any of the bits in this field indicates that a Late Collision error occurred for the specific port. Each bit in this field corresponds to one of the four ports. The least-significant bit corresponds to Port 0; the most-significant bit corresponds to Port 3.

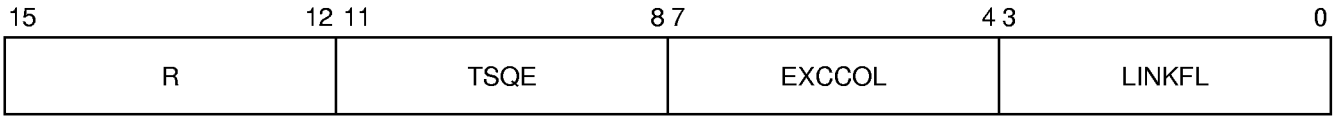
**RCFIFO** **PORT x Receive FIFO Overrun Error** **[7:4]**
A one on any of the bits in this field indicates that a Receive FIFO Overrun error occurred for the specific port. Each bit in this field corresponds to one of the four ports. The least-significant bit corresponds to Port 0; the most-significant bit corresponds to Port 3.

**TRFIFO** **PORT x Transmit FIFO Underrun Error** **[3:0]**
A one on any of the bits in this field indicates that a Transmit FIFO Underrun error occurred for the specific port. Each bit in this field corresponds to one of the four ports. The least-significant bit corresponds to Port 0; the most-significant bit corresponds to Port 3.

### 2.4.2 Error Register 1

Figure 2.7 shows Error Register 1. Error Register 1 is a 16-bit read/write register. Upon power-up or when $\overline{\text{RESET}}$ is asserted, all bits in this register will be zeros (all interrupts are disabled). Each active field in Error Register 1 contains four bits. Each bit corresponds to one of the four ports. The least-significant bit corresponds to Port 0; the most-significant bit corresponds to Port 3.

Figure 2.7
Error Register 1




**R** **Reserved** **[15:12]**
These bits are reserved and read as zeros.

**TSQE** **Transmit SQE Error Port x** **[11:8]**
A one on any of these bits indicates that an SQE error occurred for the specific port.

**EXCCOL** **Excessive Collision Port x Error** **[7:4]**
A one on any of the bits in this field indicates that the frame could not be transmitted after 16 attempts for the specific port. The L64381 aborts the transmission with no further retries.

**LINKFL** **Link Failure Port x Error** **[3:0]**
A one on any of the bits in this field indicates that a Link Failure error (either no data or link integrity pulses were received for 100 ms) occurred for the specific port.

---

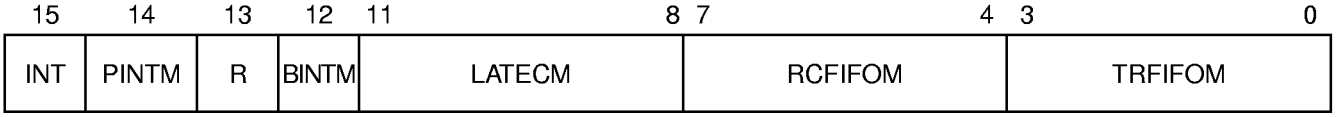
## 2.5 Error Mask Registers

The two Error Mask Registers contain the mask bits for the error fields in the corresponding Error Registers. When a bit is set to one in the Mask Register, the corresponding error is masked and the L64381 does not assert the $\overline{\text{INTERRUPT}}$ signal for that error condition.

### 2.5.1 Error Mask Register 0

Figure 2.8 shows Error Mask Register 0. This register contains the mask bits for the error fields in Error Register 0. Upon power-up or when $\overline{\text{RESET}}$ is asserted, all bits in this register are zeros.

Figure 2.8
Error Mask Register 0




**INT** **Enable Interrupts** **15**
When this bit is a one, the L64381 generates an interrupt whenever a bit in the Error Register is written to a one, provided that the bit's corresponding mask bit is not set in the Error Mask Register.

**PINTM** **Processor Interface Error Mask** **14**
A one on this bit masks the Processor Interface error.

**R** **Reserved** **13**
This bit is reserved and reads as zero.

**BINTM** **Bus Interface Error Mask** **12**
A one on this bit masks the Bus Interface error.

**LATECM** **PORT x Late Collision Error Mask** **[11:8]**
A one on any of the bits in this field masks the corresponding port's late collision error. Each bit in this field corresponds to one of the four ports. The least-significant bit corresponds to Port 0; the most-significant bit corresponds to Port 3.

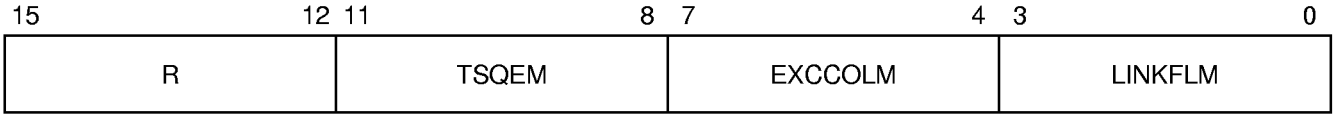
**RCFIFOM** **PORT x Receive FIFO Overrun Error Mask** **[7:4]**
A one on any of the bits in this field masks the corresponding port's Receive FIFO overrun error. Each bit in this field corresponds to one of the four ports. The least-significant bit corresponds to Port 0; the most-significant bit corresponds to Port 3.

**TRFIFOM** **PORT x Transmit FIFO Underrun Error Mask** **[3:0]**
A one on any of the bits in this field masks the corresponding port's Transmit FIFO underrun error. Each bit in this field corresponds to one of the four ports. The least-significant bit corresponds to Port 0; the most-significant bit corresponds to Port 3.

### 2.5.2 Error Mask Register 1

Figure 2.9 shows Error Mask Register 1. This register contains the mask bits for the error fields in Error Register 1. Upon power-up or when $\overline{\text{RESET}}$ is asserted, all bits in this register are cleared to zeros.

Figure 2.9
Error Mask Register 1




**R** **Reserved** **[15:12]**
These bits are reserved and read as zeros.

**TSQEM** **Transmit SQE Error Port x Mask** **[11:8]**
A one on any of the bits in this field masks the corresponding port's transmit SQE error. Each bit in this field corresponds to one of the four ports. The least-significant bit corresponds to Port 0; the most-significant bit corresponds to Port 3.

**EXCCOLM** **Excessive Collision Port x Error Mask** **[7:4]**
A one on any of the bits in this field masks the corresponding port's excessive collision error. Each bit in this field corresponds to one of the four ports. The least-significant bit corresponds to Port 0; the most-significant bit corresponds to Port 3.

**LINKFLM** **Link Failure Port x Error Mask** **[3:0]**
A one on any of the bits in this field masks the corresponding port's link failure error. Each bit in this field corresponds to one of the four ports. The least-significant bit corresponds to Port 0; the most-significant bit corresponds to Port 3.

---

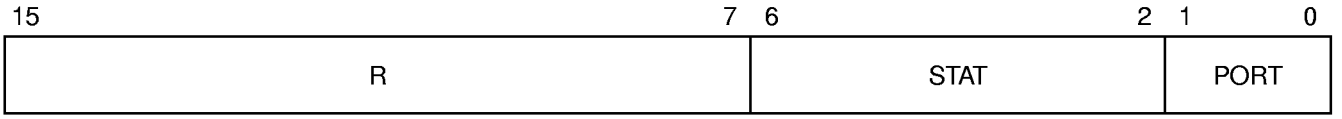
## 2.6 Statistics Counters Registers

This section describes the five registers that are used to access the Statistics Counters: the Statistics Counters Address Register, the two Statistics Counters Data-In Registers, and the two Statistics Counters Data-Out Registers.

### 2.6.1 Statistics Counters Address Register

Figure 2.10 shows the Statistics Counters Address Register. This read/write register stores a pointer to the Statistics Counters. This pointer is used to either read a counter value or write a value to the specified counter. When a burst read instruction is performed, the L64381 increments the counter value automatically, but the port number remains unaffected. Similarly in the autoincrement diagnostics mode, the L64381 does not increment the port number.

Figure 2.10
Statistics Counters
Address Register




**R** **Reserved** **[15:7]**
These bits are reserved and read as zeros.

**STAT** **Statistics Counter** **[6:2]**
These bits determine to which counter the read/write access pertains.

**PORT** **Port Number** **[1:0]**
These bits select the port number of the selected counter in the STAT field.

### 2.6.2 Statistics Counters Data-In Registers

Figures 2.11 and 2.12 show the Statistics Counters Data-In Registers. The values in these two 16-bit registers are written into the Statistics Counters.

Note: The write statistical counter operation may occur in any mode. Because the L64381 internally uses a big-endian addressing scheme, the lower 16 bits of the 32-bit data are written to Statistics Counters Data-In Register 0, and the upper 16 bits are written to Statistics Counters Data-In Register 1.

Figure 2.11
Statistics Counters
Data In-Register 0

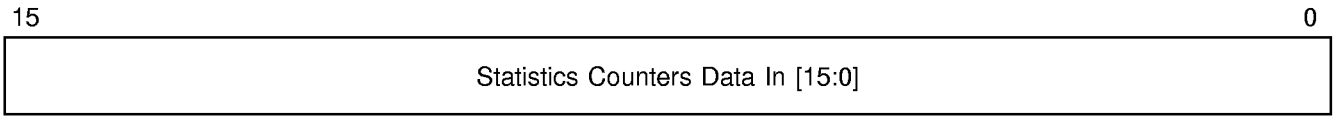
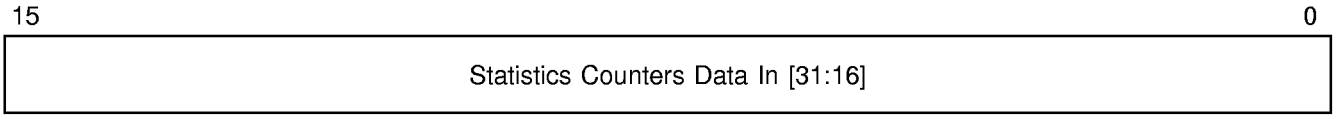



Figure 2.12
Statistics Counters
Data In-Register 1




### 2.6.3 Statistics Counters Data-Out Registers

Figures 2.13 and 2.14 show the Statistics Counters Data-Out Registers. Values read from the Statistics Counters are placed in these two 16-bit, read-only registers. Because the L64381 internally uses a big-endian addressing scheme, the lower 16 bits of the 32-bit data are read out of Statistics Counters Data-Out Register 0, and the upper 16 bits are read out of Statistics Counters Data-Out Register 1.

Figure 2.13
Statistics Counters
Data Out Register 0

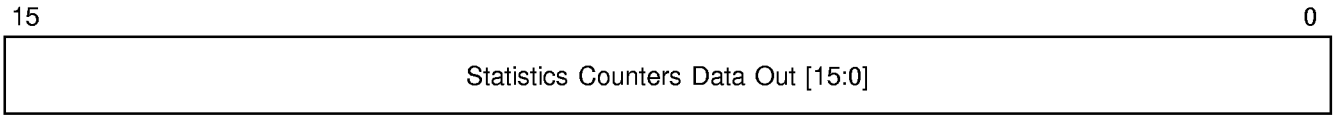
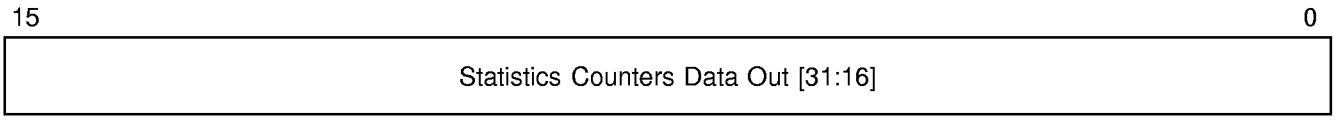



Figure 2.14
Statistics Counters
Data Out Register 1




## 2.7 Data FIFO Registers

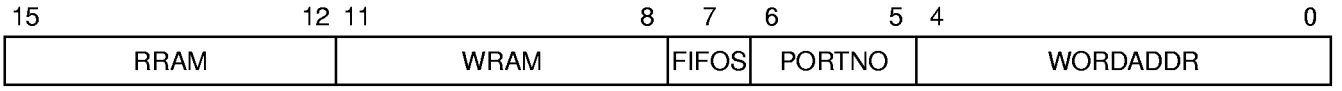
This section describes the five registers that are used to access the Data FIFO: the Data FIFO Address Register, the two Data FIFO Data In Registers, and the two Data FIFO Data Out Registers. The Transmit and Receive FIFOs contain 128 bytes in a 32 x 36 bit array. Refer to Section 4.2, “On-chip FIFOs,” for more information on the configuration of the FIFOs.

### 2.7.1 Data FIFO Address Register

Figure 2.15 shows the Data FIFO Address Register. This 16-bit register contains the address for either a Data FIFO read or a Data FIFO write. Note that transfers between the Data FIFO Address Register and the Data FIFO occur only in Diagnostic Modes 1 and 3.

In Diagnostic Modes 1 and 3, the entire 36-bit wide FIFO can be tested. To write to a 36-bit location in the Data FIFO, the lower 32 bits are placed in the Data FIFO Data In Registers. The upper four bits are written to the Data FIFO from the WRAM field of the Data FIFO Address Register. To read back a Data FIFO location, the lower 32 bits are read into the Data FIFO Data Out Registers. The upper four bits are placed in the RRAM field of the Data FIFO Address Register.

Figure 2.15
Data FIFO Address Register




**RRAM** **Read Data FIFO** **[15:12]**
During a Data FIFO read, the L64381 writes the value of the four most-significant bits in the selected Data FIFO location into these four bits. This value is the user-encoded value of $\overline{\text{BYTE\_VALID}}$[3:0], $\overline{\text{SOP}}$, $\overline{\text{EOP}}$, and $\overline{\text{AI\_FCS\_IN}}$. The addition of this field allows reading of all 36 bits of the selected Data FIFO without requiring an additional register decode.

**WRAM** **Write Data FIFO** **[11:8]**
These bits contain the value to be written into the upper four bits of one of the 32 words in the selected Data FIFO. When a bus interface transaction occurs, this value is a user-encoded value for $\overline{\text{BYTE\_VALID}}$[3:0], $\overline{\text{SOP}}$, $\overline{\text{EOP}}$, and $\overline{\text{AI\_FCS\_IN}}$. The addition of this field allows writing of all 36 bits of the Data FIFO without requiring an additional register decode.

**FIFOS** **FIFO Select** **7**
This bit selects either the Transmit FIFO (FIFOS = 0) or the Receive FIFO (FIFOS = 1) for the read or write operation.

**PORTNO** **Port Number** **[6:5]**
This field selects the port number (3:0) for the read or the write operation.

**WORDADDR** **Word Address** **[4:0]**
This field selects one of the words within the 128-byte FIFO.

When the processor writes to the Data FIFO Address Register, all 16 bits of the register are written, allowing for testing of the register. If the processor executes a Read Data FIFO instruction, the highest four bits are overwritten with the encoded value field of the Data FIFO. Therefore, when testing the Data FIFO Address Register, the user should execute a Read Data FIFO Address Register instruction before executing a Read Data FIFO instruction.

### 2.7.2 Data FIFO Data-In Registers

Figures 2.16 and 2.17 show the Data FIFO Data-In Registers. Values stored in these registers can be written into the Data FIFO in Diagnostic Modes 1 and 3 (refer to the DIAG field in Configuration Register 1). Because the L64381 uses a big-endian addressing scheme, it places the lower 16 bits of the 32-bit data in Data FIFO Data-In Register 0, and places the upper 16 bits in Data FIFO Data-In Register 1.

Figure 2.16
Data FIFO Data In
Register 0

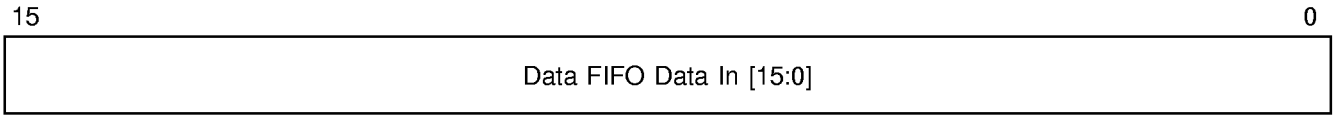
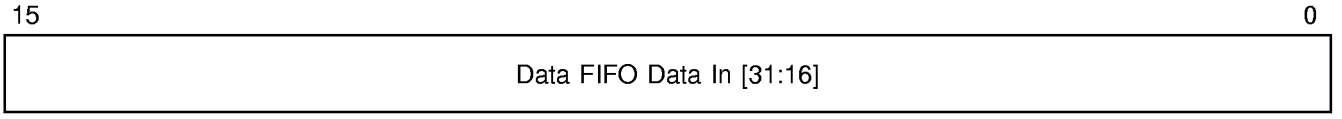



Figure 2.17
Data FIFO Data In
Register 1




### 2.7.3 Data FIFO Data-Out Registers

Figures 2.18 and 2.19 show the Data FIFO Data-Out Registers. Values read from the Data FIFO are written into these read-only registers. Because the L64381 uses a big-endian addressing scheme, the lower 16 bits of the 32-bit data are read out of Data FIFO Data-Out Register 0, and the upper 16 bits are read out of Data FIFO Data-Out Register 1.

Figure 2.18
Data FIFO Data Out
Register 0

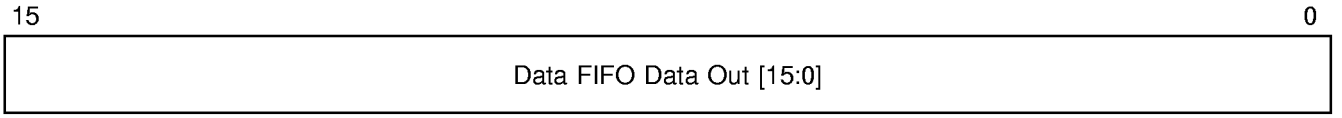
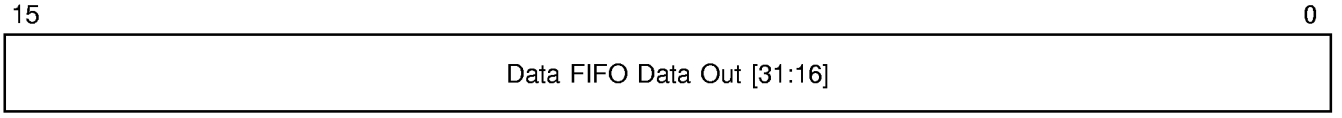



Figure 2.19
Data FIFO Data Out
Register 1



## 2.8 Receive FIFO Head Pointer Registers

Figures 2.20 and 2.21 show the Receive FIFO Head Pointer Registers. These registers are used to read and write the Receive FIFO head pointers. The head pointers point to the first location in the Receive FIFO with valid data to read. Register 0 contains the head pointers for Ports 0 and 1, and register 1 contains the head pointers for Ports 2 and 3. The FIFO pointers are each five bits in length. The most-significant bits of both bytes (bits [15:13] and [7:5]) are ignored on a write instruction, and a read of these registers returns a zero in bits [15:13] and [7:5].

Figure 2.20
Receive FIFO Head
Pointer Register 0

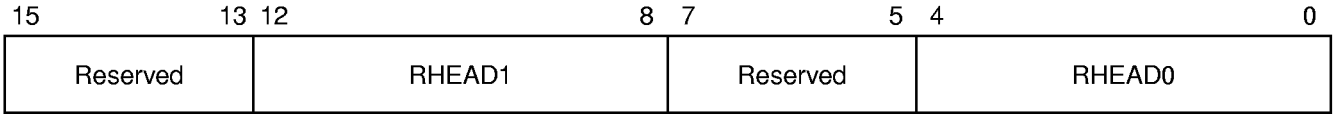
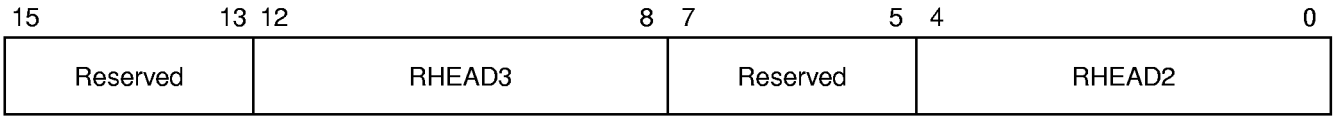



Figure 2.21
Receive FIFO Head
Pointer Register 1




## 2.9 Receive FIFO Tail Pointer Registers

Figures 2.22 and 2.23 show the Receive FIFO Tail Pointer Registers. These registers are used to read and write the Receive FIFO tail pointers. The tail pointers point to the first empty location in the Receive FIFO for writing data. Register 0 contains the tail pointers for Ports 0 and 1, and register 1 contains the tail pointers for Ports 2 and 3. The FIFO pointers are each five bits in length. The most-significant bits of both bytes (bits [15:13] and [7:5]) are ignored on a write instruction, and a read of these registers returns a zero in bits [15:13] and [7:5].

Figure 2.22
Receive FIFO Tail
Pointer Register 0

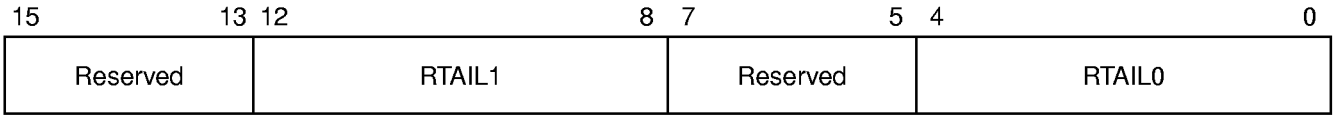
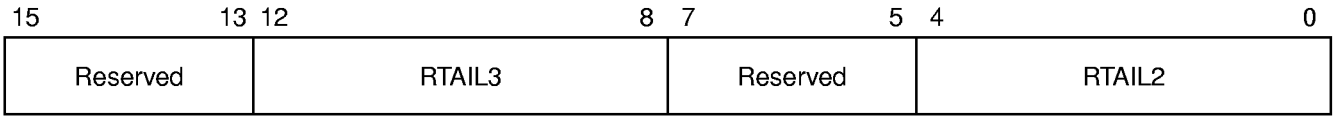



Figure 2.23
Receive FIFO Tail
Pointer Register 1




## 2.10 Transmit FIFO Head Pointer Registers

Figures 2.24 and 2.25 show the Transmit FIFO Head Pointer Registers. These registers are used to read and write the Transmit FIFO head pointers. The head pointers point to the first location in the Transmit FIFO with valid data to read. Register 0 contains the head pointers for Ports 0 and 1, and Register 1 contains the head pointers for Ports 2 and 3. The FIFO pointers are each five bits in length. The most-significant bits of both bytes (bits [15:13] and [7:5]) are ignored on a write instruction, and a read of these registers returns a zero in bits [15:13] and [7:5].

Figure 2.24
Transmit FIFO Head
Pointer Register 0

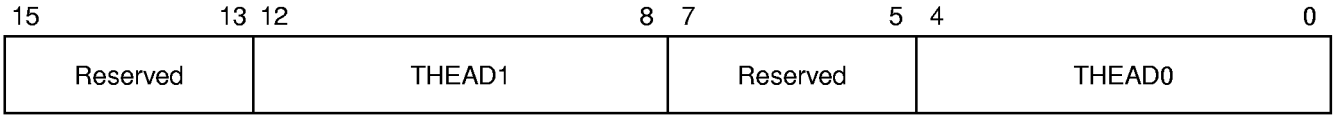
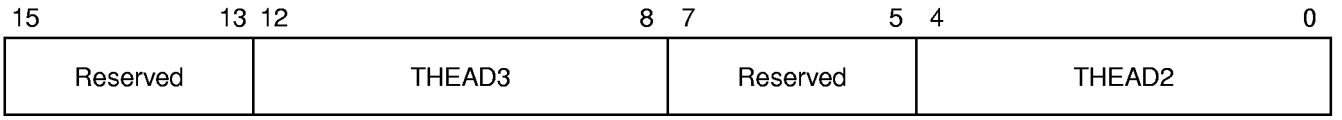



Figure 2.25
Transmit FIFO Head
Pointer Register 1



## 2.11 Transmit FIFO Tail Pointer Registers

Figures 2.26 and 2.27 show the Transmit FIFO Tail Pointer Registers. These registers are used to read and write the Transmit FIFO tail pointers. The tail pointers point to the first empty location in the Transmit FIFO for writing data. Register 0 contains the tail pointers for Ports 0 and 1, and register 1 contains the tail pointers for Ports 2 and 3. The FIFO pointers are each five bits in length. The most-significant bits of both bytes (bits [15:13] and [7:5]) are ignored on a write instruction, and a read of these registers returns a zero in bits [15:13] and [7:5].

Figure 2.26
Transmit FIFO Tail
Pointer Register 0

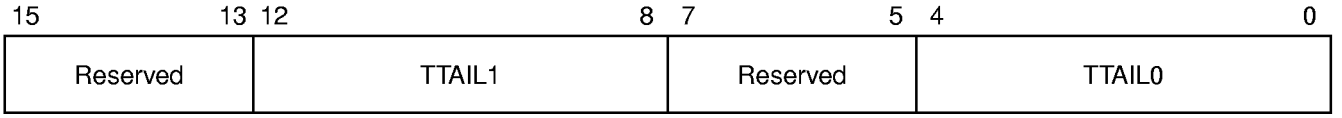
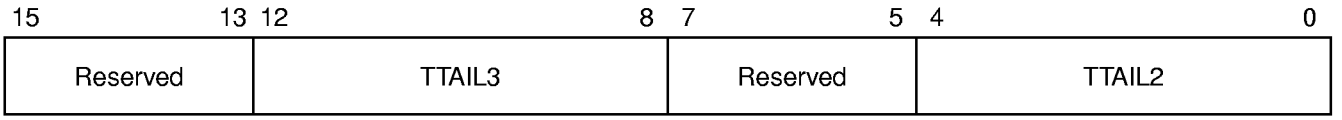



Figure 2.27
Transmit FIFO Tail
Pointer Register 1




## 2.12 Receive EOP Counters

Figures 2.28 and 2.29 show the Receive EOP Counters. These counters count the number of end of packets (EOPs) that are in the Receive FIFO. The port increments the counter when it receives the end of a packet, and decrements the counter when the Bus Interface asserts $\overline{\text{EOP}}$ on the bus. If the counter value is greater than or equal to one, there is at least one EOP in the Receive FIFO. Even if the number of bytes left in the FIFO is less than the programmed bus transfer size (BUS field), the L64381 still asserts $\overline{\text{PKT\_AVAIL}}$ to allow the host to drain the Receive FIFO. This scheme permits better use of the Receive FIFO, because no padding of data is needed to align it with the Bus Transfer Size, hence FIFO space is not wasted with pad data. Counter 0 contains the counter values for Ports 0 and 1, and Counter 1 contains the counter values for Ports 2 and 3.

Figure 2.28
Receive EOP Counter 0

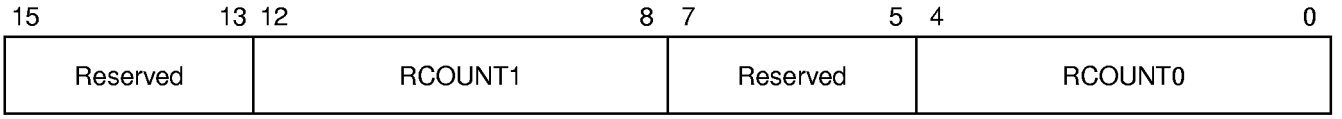
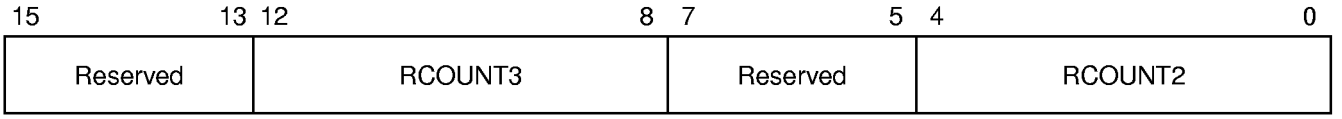



Figure 2.29
Receive EOP Counter 1




## 2.13 Transmit EOP Counters

Figures 2.30 and 2.31 show the Transmit EOP Counters. These counters count the number of EOPs that are present in the Transmit FIFO. The port increments the counter when the Bus Interface receives an end of a packet from the bus ($\overline{EOP}$ is asserted), and decrements the counter when the end of packet is sent to the port for transmission. Similar to the Receive EOP Counter, this scheme guarantees that a packet is transmitted when the packet size is smaller than the Transmit Threshold field. Note that this transmit can occur because the L64381 allows automatic padding of packets. This scheme permits better use of the Transmit FIFO, since no padding of data is needed to align it with the Transmit Threshold, hence FIFO space is not wasted with pad data. Register 0 contains the counter values for Ports 0 and 1, and Register 1 contains the counter values for Ports 2 and 3.

Figure 2.30
Transmit EOP
Counter 0

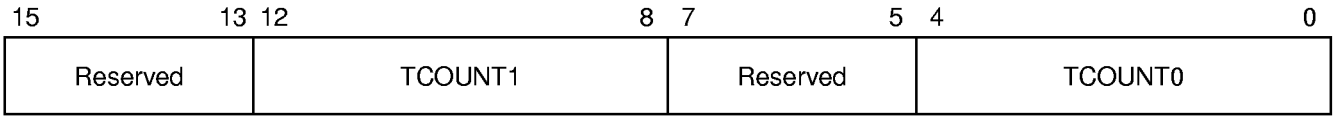
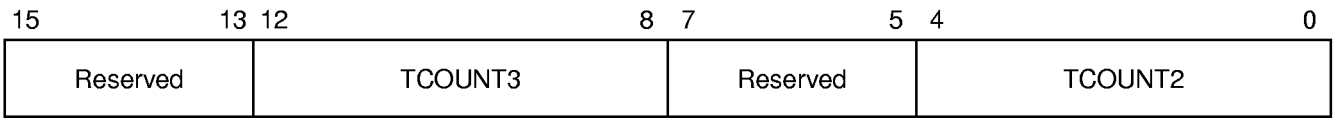



Figure 2.31
Transmit EOP
Counter 1



# Chapter 3
# Signal Descriptions

This chapter describes the signals that comprise the L64381's bit-level interface to other devices. This description is intended for hardware designers who are connecting the product to other components.

This chapter is divided into seven sections:

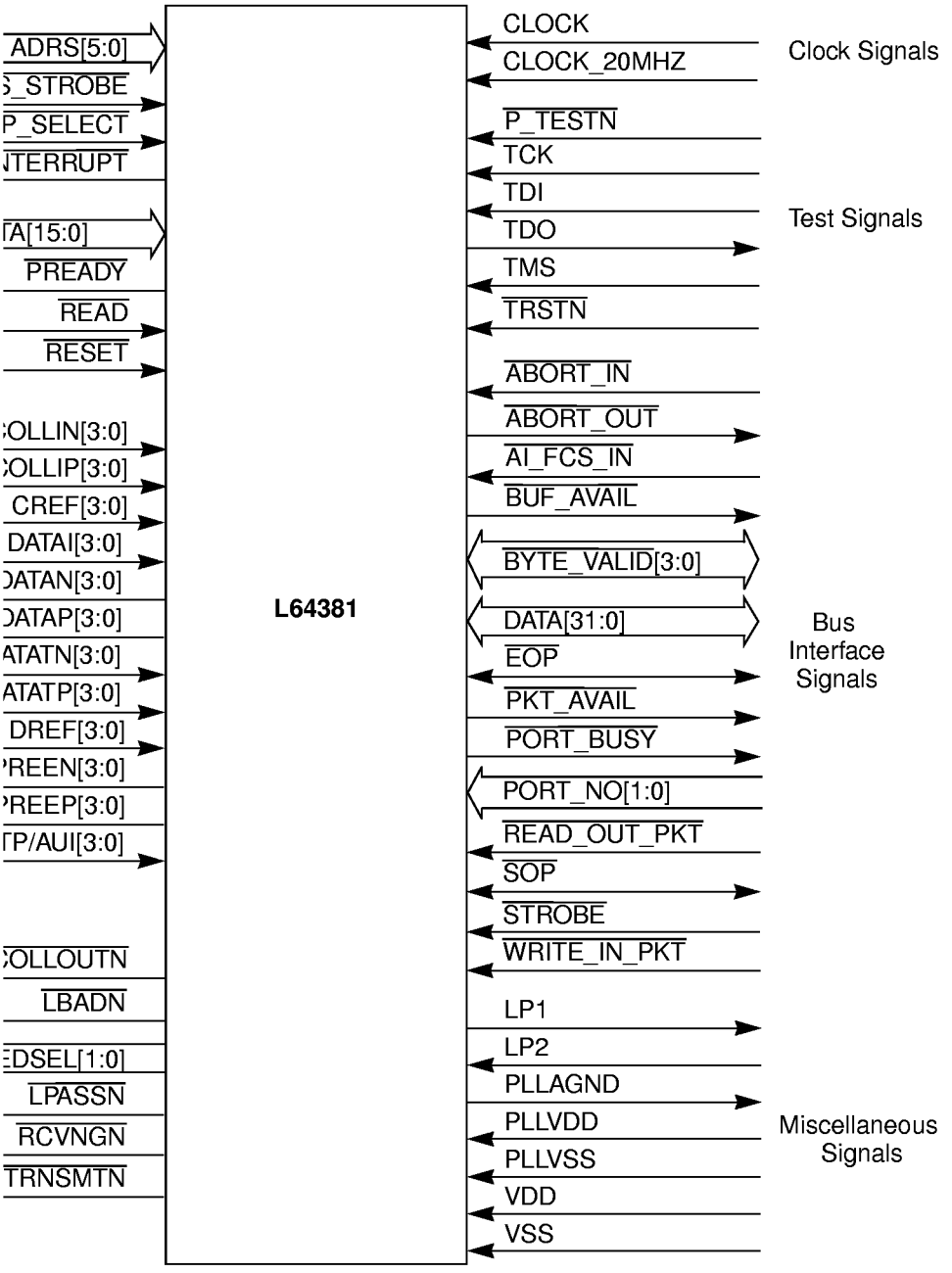
- Section 3.1, "Network Interface Signals"
- Section 3.2, "Bus Interface Signals"
- Section 3.3, "Processor Interface Signals"
- Section 3.4, "Status Signals"
- Section 3.5, "Clock Signals"
- Section 3.6, "Test Signals"
- Section 3.7, "Miscellaneous Signals"

The product's interface signals are described in alphabetical order by mnemonic. Each signal definition contains the mnemonic and the full signal name. The mnemonics for signals that are active LOW are marked with an overbar; all others are active HIGH. For example, $\overline{WE}$ is active LOW, and RD is active HIGH.

In the descriptions that follow, the verb *assert* means to drive TRUE or active. The verb *deassert* means to drive FALSE or inactive.

Figure 3.1 shows the logic diagram for the L64381.

Figure 3.1
L64381 Logic Diagram



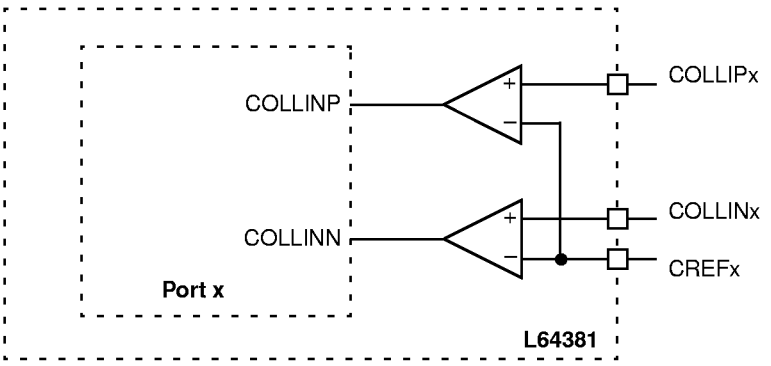
## 3.1 Network Interface Signals

This section describes the signals that comprise the L64381 Network Interface.

**COLLIN[3:0]** **Negative Collision Threshold** **Input**
COLLIN[3:0] and COLLIP[3:0] are used to detect collisions. COLLIN[3:0] are valid only in AUI mode (TP/AUI signal is LOW), and should be connected to the outputs of the negative collision threshold differential receivers. COLLIN[3:0] correspond to Ports [3:0], respectively.

**COLLIP[3:0]** **Positive Collision Threshold** **Input**
COLLIN[3:0] and COLLIP[3:0] are used to detect collisions. COLLIP[3:0] are valid only in AUI mode (TP/AUI signal is LOW), and should be connected to the outputs of the positive collision threshold differential receivers. COLLIP[3:0] correspond to Ports [3:0], respectively.

**CREF[3:0]** **Collision Reference** **Input**
These inputs are the reference voltages for the COLLIN[3:0] and COLLIP[3:0] inputs. The internal circuitry follows:




**DATAI[3:0]** **Serial Data In** **Input**
Each port receives serial data from the network on these pins.

**DATAN[3:0]** **Negative Manchester-Encoded Data** **Output**
Each port transmits negative Manchester-encoded data on these pins.

**DATAP[3:0]** **Positive Manchester-Encoded Data** **Output**
Each port transmits positive Manchester-encoded data on these pins.

**DATATN[3:0]** **Data Threshold Negative** **Input**

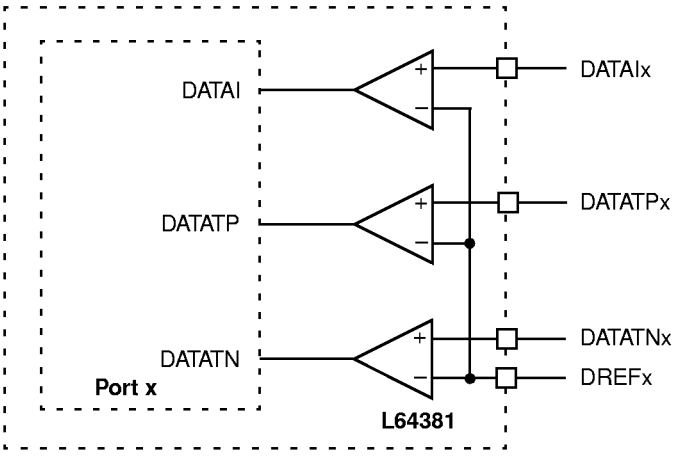
These pins, along with the DATATP pins, control the CASCADE squelch function and link integrity pulse detection for their respective port. The DATATN inputs use a differential comparator that compares the serial data received from the network with a negative voltage reference.

**DATATP[3:0]** **Data Threshold Positive** **Input**

These pins, along with the DATATN pins, control the CASCADE squelch function and link integrity pulse detection for their respective port. The DATATP inputs compares the serial data received with a positive voltage reference.

**DREF[3:0]** **Data Threshold Reference** **Input**

These inputs are the reference voltages for the DATATN, DATATP, and DATAI inputs. The internal circuitry is shown in the following figure:




**PREEN[3:0]** **Negative Preemphasis** **Output**

The ports transmit negative preemphasis on these outputs. These outputs are used in twisted-pair mode only and control the negative preemphasis output. The purpose of PREEN is to increase the amplitude of the higher frequencies in the output signal. Data preemphasis guarantees that twisted-pair mode can drive signals properly at 10 Mbit/s on cable up to 100 meters long. These signals are resistively combined: DATAP with PREEP and DATAN with PREEN. The technique of resistively combining the signals is known as digital preemphasis.

**PREEP[3:0]** **Positive Preemphasis** **Output**

The ports transmit positive preemphasis on these outputs. These outputs are used in twisted-pair mode only and control the positive preemphasis output. The purpose of PREEP is to increase the amplitude of the higher frequencies in the output signal. Data preemphasis guarantees that twisted-pair mode can drive signals properly at 10 Mbit/s on cable up to 100 meters long. These signals are resistively combined: DATAP with PREEP and DATAN with PREEN. The technique of resistively combining the signals is known as digital preemphasis.

**TP/AUI[3:0]** **Twisted-Pair/AUI Select** **Input**

These signals determine whether the specific port [3:0] is in AUI mode or in twisted-pair mode. A LOW on one of these signals indicates that the corresponding port is in AUI mode; a HIGH indicates that the corresponding port is in twisted-pair mode. The following table lists the principal differences between these two modes.

| Feature | Twisted Pair | AUI |
|---|---|---|
| SQE Test (heartbeat) | No | Yes |
| Link Pulse Detection | Yes | No |
| Link Pulse Polarity Detection and Correction | Yes | No |
| Link Integrity Pulses Generated | Yes | No |
| Collision Pair Pins (COLLIP, COLLIN) are Used | No | Yes |
| Cause of Transmit Carrier Loopback Error | Link Fail | No Carrier Loopback |

## 3.2 Bus Interface Signals

This section describes the signals that comprise the L64381's Bus Interface.

**$\overline{\text{ABORT\_IN}}$** **Abort In** **Input**

The system uses the $\overline{\text{ABORT\_IN}}$ signal to indicate that the L64381 should ignore the current bus transfer and restore the internal FIFO pointers to their value before the current bus transfer began. In effect, it is as if the current bus transfer did not happen. Assertion of $\overline{\text{ABORT\_IN}}$ does not cause any action on the network side.

Two cases need to be considered when $\overline{\text{ABORT\_IN}}$ is asserted while either writing into the Transmit FIFO or reading from the Receive FIFO:

- While writing into the Transmit FIFO, if the L64381 detects the assertion of $\overline{\text{ABORT\_IN}}$, then the L64381 ignores that bus transfer and restores the internal Transmit FIFO tail pointer to the value it contained before the current bus transfer began.
- While reading from the Receive FIFO, if the L64381 detects the assertion of $\overline{\text{ABORT\_IN}}$, then it stops the current transfer and restores the Receive FIFO head pointer to the value it contained just before the transfer was started. On the bus side, the L64381 drives all zeros on the data bus and negates the $\overline{\text{BYTE\_VALID}}${3:0], $\overline{\text{SOP}}$, and $\overline{\text{EOP}}$ signals for the rest of the bus transfer.

If $\overline{\text{ABORT\_IN}}$ is asserted after the assertion of $\overline{\text{EOP}}$ on the current bus transfer, the L64381 ignores it. The PORT_NO inputs determine which port is selected.

**$\overline{\text{ABORT\_OUT}}$ Abort Out** **Output**

The L64381 asserts $\overline{\text{ABORT\_OUT}}$ upon detection of a failing condition for either a Transmit or a Receive operation. For a Receive operation, the L64381 asserts $\overline{\text{ABORT\_OUT}}$ only if $\overline{\text{EOP}}$ is asserted during a packet read operation. A packet read operation can be a Runt/Jabber packet, a packet with bad FCS, an internal PLL error, or a receive FIFO overrun error. When a failing condition occurs on a receiving port, the L64381 reads the last received byte (or bytes) from that port through the Bus Interface, and asserts $\overline{\text{ABORT\_OUT}}$ and $\overline{\text{EOP}}$ to signal the corrupted packet. If necessary, the L64381 also updates all associated Statistics Counters.

A receive FIFO overrun error may destroy a portion of the packet and can occur for one of two reasons:

- The first error occurs when a long packet is overrun in the middle of the packet (only some data is missing). The L64381 asserts $\overline{\text{ABORT\_OUT}}$ when the last byte (or bytes) of the current packet is read.
- The second error occurs when two packets arrive from the network: packet A arrives first, and packet B follows. Because of a receive FIFO overrun condition,

the L64381 drops the last several bytes of packet A (no $\overline{\text{EOP}}$ signal for packet A) before it receives packet B. Even if the L64381 successfully receives all of packet B without an error, it asserts $\overline{\text{ABORT\_OUT}}$ with $\overline{\text{EOP}}$ for packet B.

For a Transmit operation, the L64381 may assert $\overline{\text{ABORT\_OUT}}$ anytime during a packet write operation for one of many reasons:

- L64381 detects a late collision
- It detects excessive collisions
- It detects a collision and S64 is set to 0 in the Packet Configuration Register
- It detects a transmit FIFO underrun error
- It detects an invalid $\overline{\text{BYTE\_VALID[3:0]}}$ combination

In any case, the L64381 asserts $\overline{\text{ABORT\_OUT}}$ only during a write attempt for the same packet to a port that has just encountered an error. $\overline{\text{ABORT\_OUT}}$ is asserted when the first write operation after the failure occurs. The L64381 ignores subsequent write bus transfers for the same packet until it detects the start of a new packet. If a nontransmit FIFO underrun error occurs during the last write transfer of the packet, the L64381 does not assert $\overline{\text{ABORT\_OUT}}$ for the next new packet. The L64381 does not assert $\overline{\text{ABORT\_OUT}}$ for the current packet if that packet ends within the FIFO. The L64381 simply flushes the packet from the FIFO.

The L64381 requires three cycles after a write transfer to send the transmit FIFO pointer values to the network. The L64381 may detect a transmit FIFO underrun error while a write transfer for the last portion of a packet is occurring (with $\overline{\text{EOP}}$ asserted). The L64381 detects this error within three cycles after it asserts $\overline{\text{EOP}}$. If a new write transfer starts on the same port within five cycles of the previous write transfer, the L64381 asserts $\overline{\text{ABORT\_OUT}}$.

The failing condition is one of the following:

- L64381 detected a Receive FIFO overrun
- Packet being received was a runt packet
- Packet being received was an oversized packet

- Packet being received had a bad FCS
- L64381 detected a PLL error.

If the L64381 does not read the failing packet out on the Bus Interface and the packet resides in the Receive FIFO, the L64381 does not flush the Receive FIFO. The L64381 updates the appropriate Statistics Counters and waits for the external device to remove the data. When the packet drains completely, the L64381 asserts $\overline{\text{ABORT\_OUT}}$ with $\overline{\text{EOP}}$ to the external device for errored packet signaling.

When data is being written to the Transmit FIFO on the Bus Interface, if the current packet being written is also being transmitted to the Network Interface, and the L64381 detects a failing condition on the Network Interface, the L64381 asserts the $\overline{\text{ABORT\_OUT}}$ signal, updates the appropriate Statistics Counters based on the failing condition, and flushes the current packet from the Transmit FIFO. The failing condition is one of the following:

- L64381 detected a late collision
- L64381 detected excessive collisions
- L64381 detected a collision, and it is not in autostore 64-byte mode
- L64381 detected a Transmit FIFO underrun error
- L64381 detected an invalid $\overline{\text{BYTE\_VALID}}$[3:0] combination

**$\overline{\text{AI\_FCS\_IN}}$** **Autoinsert FCS** **Input**

The L64381 latches in $\overline{\text{AI\_FCS\_IN}}$ when $\overline{\text{SOP}}$ and $\overline{\text{WRITE\_IN\_PKT}}$ are both asserted. Assertion of this input while $\overline{\text{SOP}}$ is asserted causes the L64381 to automatically calculate and insert the CRC value from the packet data into the FCS field of the Ethernet packet. If this input is deasserted when $\overline{\text{SOP}}$ and $\overline{\text{WRITE\_IN\_PKT}}$ are both asserted, then the data provided for transmission must contain the FCS field.

**$\overline{\text{BUF\_AVAIL}}$** **Buffer Available** **Output**

The L64381 asserts this signal in response to the assertion of $\overline{\text{STROBE}}$ when the Transmit FIFO has enough empty space available to store data from a bus transfer. The FIFO is deemed to have enough space if it

has space equal to or more than the number of words specified in the Bus Transfer Size field (BUS) in the L64381 Configuration Register 1. This signal is valid on the four cycles following the assertion of $\overline{\text{STROBE}}$. The state of this signal on the four cycles following the assertion of $\overline{\text{STROBE}}$ indicates the Transmit FIFO space availability for Ports [3:0], respectively. If at the time of assertion of $\overline{\text{STROBE}}$, a bus transfer is occurring, then $\overline{\text{BUF\_AVAIL}}$ is not asserted for that particular port but it does indicate correct status for the other ports.

**$\overline{\text{BYTE\_VALID}}$[3:0]**

**Valid Bytes** **Bidirectional**

These pins indicate which bytes are valid in the current word. Byte 0 refers to DATA[7:0], Byte 1 refers to DATA[15:8], Byte 2 refers to DATA[23:16], and Byte 3 refers to DATA[31:24]. The following table shows the valid combinations of $\overline{\text{BYTE\_VALID}}$[3:0].

| $\overline{\text{BYTE\_VALID}}$[3:0] | DATA[31:0] |
|---|---|
| 0000 | All Bytes Valid |
| 1000 | Bytes 2, 1, and 0 Valid |
| 1100 | Bytes 1 and 0 Valid |
| 1110 | Byte 0 Valid |
| 0001 | Bytes 3, 2, and 1 Valid |
| 0011 | Bytes 3, 2 Valid |
| 0111 | Byte 3 Valid |

Note: The bytes transferred with $\overline{\text{SOP}}$ or $\overline{\text{EOP}}$ can have any of the above combinations except for $\overline{\text{BYTE\_VALID}}$[3:0] = $1111_2$. Bytes transferred between $\overline{\text{SOP}}$ and $\overline{\text{EOP}}$ can have only $\overline{\text{BYTE\_VALID}}$[3:0] = $0000_2$. All other combinations not listed above are invalid.

While writing a packet from the bus, the L64381 accepts any of the seven valid $\overline{\text{BYTE\_VALID}}$[3:0] combinations: $0000_2$, $1000_2$, $1100_2$, $1110_2$, $0001_2$, $0011_2$, or $0111_2$. However, for receiving PKT with $\overline{\text{SOP}}$ asserted, the L64381 always asserts $\overline{\text{BYTE\_VALID}}$[3:0] as $0000_2$. For PKT receiving with $\overline{\text{EOP}}$ asserted, the L64381 asserts the $\overline{\text{BYTE\_VALID}}$[3:0] as one of four combinations:

$1110_2$, $1100_2$, $1000_2$, or $0000_2$. For receiving PKT with $\overline{\text{EOP}}$ asserted and dribbling bits, the L64381 asserts both $\overline{\text{BYTE\_VALID}}$[3:0] as $1111_2$ and $\overline{\text{ABORT\_OUT}}$ to signal an error condition.

If an invalid combination is encountered and the packet has already started transmission on the network, then the L64381 aborts the packet on the network (after transmitting 32 alternating ones and zeros). If the entire packet is in the FIFO then it is flushed from the FIFO without being transmitted to the network. If the entire packet is not in the FIFO, then the L64381 asserts $\overline{\text{ABORT\_OUT}}$, and the system is expected to drop the packet externally and continue with a new packet. If the system continues to write the current packet, then the L64381 ignores it until the next $\overline{\text{SOP}}$.

**DATA[31:0]** **L64381 Data** **Bidirectional**

This bus is the data interface to and from the L64381.

**$\overline{\text{EOP}}$** **End of Packet** **Bidirectional**

Assertion of $\overline{\text{EOP}}$ indicates the End Of Packet when it occurs in the current word that is being placed on the Bus Interface.

**$\overline{\text{PKT\_AVAIL}}$** **Packet Available** **Output**

The L64381 asserts $\overline{\text{PKT\_AVAIL}}$ in response to the assertion of $\overline{\text{STROBE}}$ when the Receiver FIFO has enough data to transfer on the Bus Interface. The FIFO is deemed to have enough data if it either:

- Contains data that is equal to or more than the number of words specified in the Bus Transfer Size field (BUS) of Configuration Register 1
- Contains an End of Packet

This signal is valid on the four cycles following the assertion of $\overline{\text{STROBE}}$. The state of this signal on the four cycles following the assertion of $\overline{\text{STROBE}}$ indicates the Receiver FIFO space availability for Ports [3:0], respectively. If at the time of assertion of the $\overline{\text{STROBE}}$, a bus transfer is occurring, then $\overline{\text{PKT\_AVAIL}}$ is not asserted for that particular port, but it continues to indicate the correct status of the other ports.

**PORT_BUSY** **Port Status** **Output**

In half-duplex mode (the DUPLEX field in Configuration Register 0 is reset to zero), the L64381 asserts PORT_BUSY to indicate the selected port is either transmitting or receiving. In full-duplex mode, the L64381 asserts PORT_BUSY to indicate the port is transmitting.

**PORT_NO[1:0]** **Port Number** **Input**

The state of these inputs determines to which port number [3:0] the READ_OUT_PKT and WRITE_IN_PKT inputs and the ABORT_OUT output refer. The following table shows the encoding of these inputs.

| PORT_NO[1:0] | Port Number |
|---|---|
| 00 | 0 |
| 01 | 1 |
| 10 | 2 |
| 11 | 3 |

**READ_OUT_PKT**

**Read Data from Receive FIFO** **Input**

This input informs the L64381 to place the data in its Receive FIFO on the Bus Interface. The PORT_NO bits indicate which port's FIFO is to be read.

**SOP** **Start of Packet** **Bidirectional**

Assertion of SOP indicates the Start Of Packet when it occurs in the current data transfer.

**STROBE** **Strobe** **Input**

On the four cycles following the assertion of STROBE, the L64381 places the status of Ports [3:0] respectively on the PKT_AVAIL, BUF_AVAIL, and PORT_BUSY signals. If STROBE is held asserted, the status of each of the four ports is placed on PKT_AVAIL, BUF_AVAIL, and PORT_BUSY every four cycles.

**WRITE_IN_PKT** **Write Data to Transmit FIFO** **Input**

This input informs the L64381 to write the data that is on the Bus Interface to the Transmit FIFO. The PORT_NO bits indicate which port's Transmit FIFO is to be written.

## 3.3 Processor Interface Signals

This section describes the signals that comprise the L64381's Processor Interface.

**ADRS[5:0]** **Register Select Address** **Input**

These pins determine which L64381 register to access. ADRS[5:0] are latched into the L64381 on the rising system clock edge (CLOCK) immediately following the assertion of $\overline{\text{ADRS\_STROBE}}$ and $\overline{\text{CHIP\_SELECT}}$.

**$\overline{\text{ADRS\_STROBE}}$** **Address Strobe** **Input**

Assertion of $\overline{\text{ADRS\_STROBE}}$ along with the assertion of $\overline{\text{CHIP\_SELECT}}$ causes the L64381 to latch the ADRS inputs.

**$\overline{\text{CHIP\_SELECT}}$** **Chip Select** **Input**

Assertion of this input enables access to and from the L64381 from the Processor Interface. For an absolute worst-case condition, all the statistics counters need to be updated when the host tries to access one of them. The access time from $\overline{\text{ADRS\_STROBE}}$ to $\overline{\text{PREADY}}$ may be 100+ cycles. For applications with multiple L64381 devices sharing the same processor interface, when such a worst-case condition occurs, you can deassert a particular L64381 chip select, which terminates the current transaction and allows other devices to use the bus, if desired.

**$\overline{\text{INTERRUPT}}$** **Interrupt** **Output**

The L64381 asserts $\overline{\text{INTERRUPT}}$ whenever an error bit in the Error Register is set and that bit has not been masked. The interrupt generally indicates any condition which the processor needs to resolve.

**PDATA[15:0]** **Processor Data Bus** **Bidirectional**

This 16-bit bus contains data that is transferred between the L64381 and the processor.

**$\overline{\text{PREADY}}$** **Ready** **Output**

During a write to the L64381's registers, the L64381 asserts $\overline{\text{PREADY}}$ LOW for one cycle after the rising edge on which the PDATA data is latched. For a read instruction, the L64381 drives $\overline{\text{PREADY}}$ LOW for one cycle after it places data on the PDATA bus. Data is to be read on

the rising edge of CLOCK during which $\overline{\text{PREADY}}$ is driven LOW.

**$\overline{\text{READ}}$** **Read** **Input**

This input determines whether the operation is going to be a write to the L64381 ($\overline{\text{READ}}$ = HIGH) or a read from the L64381 ($\overline{\text{READ}}$ = LOW). As with ADRS[5:0], the L64381 latches this signal when $\overline{\text{ADRS\_STROBE}}$ and $\overline{\text{CHIP\_SELECT}}$ are asserted.

**$\overline{\text{RESET}}$** **Reset** **Input**

Assertion of $\overline{\text{RESET}}$ resets the L64381. The RESET signal should be asserted for at least 1 μs. At the end of the reset, the L64381 resets all register contents to zeros, 3-states all bidirectional signals, and places all outputs in their nonasserted states.

Note: Asserting $\overline{\text{RESET}}$ sets the Enable Port x bits to all ones to disable all the ports.

---

## 3.4 Status Signals

This section describes the signals whose status is intended to be viewed using LEDs.

**$\overline{\text{COLLOUTN}}$** **Collision Out** **Output**

This pin multiplexes the Collision Status from the four ports. It is asserted when any port detects a collision on the Network Interface. The LEDSEL[1:0] pins and this pin can be used along with external logic to drive LEDs.

**$\overline{\text{LBADN}}$** **Link Inverted** **Output**

This pin is the Link Inverted Signal from one of the four ports (a multiplexer selects which port's $\overline{\text{LBADN}}$ signal is output). Each port asserts $\overline{\text{LBADN}}$ when an inverted connection is detected on the Network Interface. $\overline{\text{LBADN}}$ combined with external logic can drive an LED, allowing the status of $\overline{\text{LBADN}}$ to be viewed. LEDSEL[1:0] indicate which port is selected.

**LEDSEL[1:0]** **LED Select** **Output**

These pins indicate which port's status is being output on $\overline{\text{COLLOUTN}}$, $\overline{\text{LBADN}}$, $\overline{\text{LPASSN}}$, $\overline{\text{RCVNGN}}$, and $\overline{\text{TRNSMTN}}$ in the current cycle.

| LEDSEL[1:0] | Port Selected |
|---|---|
| 00 | 0 |
| 01 | 1 |
| 10 | 2 |
| 11 | 3 |

**$\overline{\text{LPASSN}}$** **Link Test Status** **Output**

This pin is the Link Test Status of one of the four ports (a multiplexer selects which port's $\overline{\text{LPASSN}}$ signal is output). The L64381 asserts $\overline{\text{LPASSN}}$ when the Link test passes. $\overline{\text{LPASSN}}$ combined with external logic can drive an LED, allowing the status of $\overline{\text{LPASSN}}$ to be viewed. LEDSEL[1:0] indicate which port is selected.

**$\overline{\text{RCVNGN}}$** **Receiver Status** **Output**

This pin is the Receiver Status from one of the four ports (a multiplexer selects which port's $\overline{\text{RCVNGN}}$ signal is output). The L64381 asserts $\overline{\text{RCVNGN}}$ when the port is receiving data on the Network Interface. $\overline{\text{RCVNGN}}$ combined with external logic can drive an LED, allowing the status of $\overline{\text{RCVNGN}}$ to be viewed. LEDSEL[1:0] indicate which port is selected.

**$\overline{\text{TRNSMTN}}$** **Transmitter Status** **Output**

This pin is the Transmitter Status from one of the four ports (a multiplexer selects which port's $\overline{\text{TRNSMTN}}$ signal is output). It is asserted when the port is transmitting data on the Network Interface. $\overline{\text{TRNSMTN}}$ combined with external logic can drive an LED, allowing the status of $\overline{\text{TRNSMTN}}$ to be viewed. LEDSEL[1:0] indicate which port is selected.

## 3.5 Clock Signals

This section describes the input clocks required for the L64381's operation.

**CLOCK** — **System Clock** — **Input**
This 16-MHz to 33-MHz clock is used for the internal operation of the L64381, the Bus Interface, and the Processor Interface.

**CLOCK_20MHZ** — **Internal Clock** — **Input**
The L64381 uses this 20-MHz clock to generate the 80-, 20-, and 10-MHz clocks used by the ports.

## 3.6 Test Signals

This section describes the signals that comprise the L64381's test circuitry. The JTAG signals operate as described in the *IEEE Standard Test Access Port and Boundary-Scan Architecture* (IEEE1149) document.

**P_TESTN** — **Global 3-State** — **Input**
Assertion of this input LOW 3-states all of the output and bidirectional pins of the L64381. This input is held HIGH during normal operation.

**TCK** — **JTAG Test Clock** — **Input**
TCK is the JTAG Test Clock. It typically is a 50% duty cycle clock running at 10 MHz minimum.

**TDI** — **JTAG Test Data In** — **Input**
Serial test instructions and data are input through the TDI pin.

**TDO** — **JTAG Test Data Out** — **Output**
The result of the serial test instructions and data that were input through TDI is output on TDO.

**TMS** — **JTAG Test Mode Select** — **Input**
The TAP controller decodes this input to determine test operations.

**$\overline{\text{TRSTN}}$** — **JTAG Test Reset** — **Input**
Assertion of this active LOW signal provides asynchronous initialization of the TAP controller.

## 3.7 Miscellaneous Signals

This section describes the remaining signals in the L64381.

**LP1** — **Phase Detector Output** — **Output**
LP1 is the output of the phase detector, which is embedded within the on-chip PLL.

**LP2** — **VCO Input** — **Input**
LP2 is the input to the voltage-controlled oscillator (VCO).

**PLLAGND** — **Phased-Locked Loop Analog Ground** — **Output**
This output is the analog ground for the phased locked loop.

**PLLVDD** — **Phased-Locked Loop Power** — **Input**
This input provides the power to the phased locked loop.

**PLLVSS** — **Phased-Locked Loop Digital Ground** — **Input**
This input is the digital ground for the phased locked loop.

**VDD** — **Power** — **Input**
The VDD inputs are the +5 V power pins. All VDD inputs can be connected to a single supply. However, to minimize EMI and digital noise coupling, the L64381 provides individual VDD inputs for Ports [3:0]. Refer to Section 7.3, “Special VDD and VSS Connections,” for the pin assignments of these power pins.

**VSS** — **Ground** — **Input**
The VSS inputs are the ground pins. All VSS inputs can be tied together. However, to minimize EMI and digital noise coupling, the L64381 provides individual VSS inputs for Ports [3:0]. Refer to Section 7.3, “Special VDD and VSS Connections,” for the pin assignments of these ground pins.

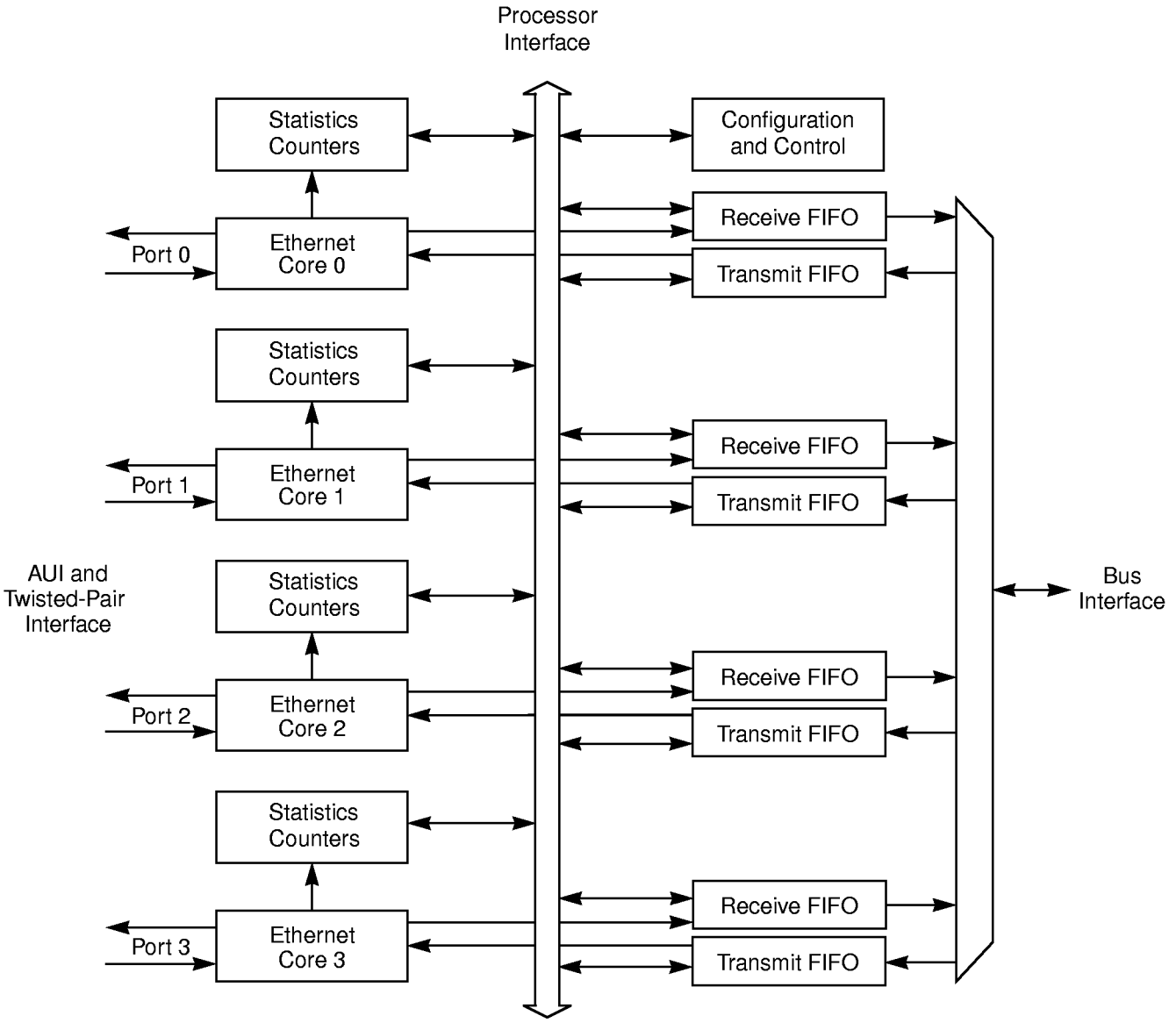
# Chapter 4
# L64381 Operation

This chapter provides a detailed description of the functional blocks and interfaces within the L64381. This chapter contains eight sections:

- Section 4.1, “Ethernet Cores”
- Section 4.2, “On-chip FIFOs”
- Section 4.3, “Bus Interface”
- Section 4.4, “Processor Interface”
- Section 4.5, “PLL Connections”
- Section 4.6, “Twisted-Pair Interface”
- Section 4.7, “AUI Interface”
- Section 4.8, “JTAG Initialization”

Figure 4.1 shows the L64381 block diagram.

Figure 4.1
L64381 Block Diagram




## 4.1 Ethernet Cores

The Ethernet cores that comprise the four ports are fully synchronous designs. A synchronous design has the benefit of being predictable and reliable because logic output signals change at known times with respect to the clock.

The core processes data at 10 Mbit/s at half-duplex mode and 20 Mbit/s in full-duplex mode and provides a comprehensive solution for 10BASE-T based systems. The core engine gives the designer a very effective

systems integration building block for developing next-generation networking products for switched hub or router applications. The core also enables the highest form of system integration for workstation or high-end PCs by converting all the I/O and system logic into a single-chip solution.

The functional blocks within the core are:

- Manchester Encoder/Decoder. The core contains an integrated Encoder/Decoder function that performs Manchester encoding and decoding, and utilizes a digital phase-locked loop for data recovery at 10 Mbit/s.
- Media Access Control (MAC). The Media Access Control function provides simple and efficient packet transmission and reception control by means of parallel eight-bit data interfaces.
- Transceiver Interface. The core provides inputs and outputs that are easily connected to on-chip integrated 10BASE-T (twisted-pair) transceivers compatible with IEEE 802.3 networks. The core can also provide AUI signals for 10BASE-2, 10BASE-5, or 10BASE-F media. The core may be connected directly to the MAC interface.
- The transceiver interface logic incorporates the transmitter, receiver, collision, link integrity, and loopback functions as defined in the standard.

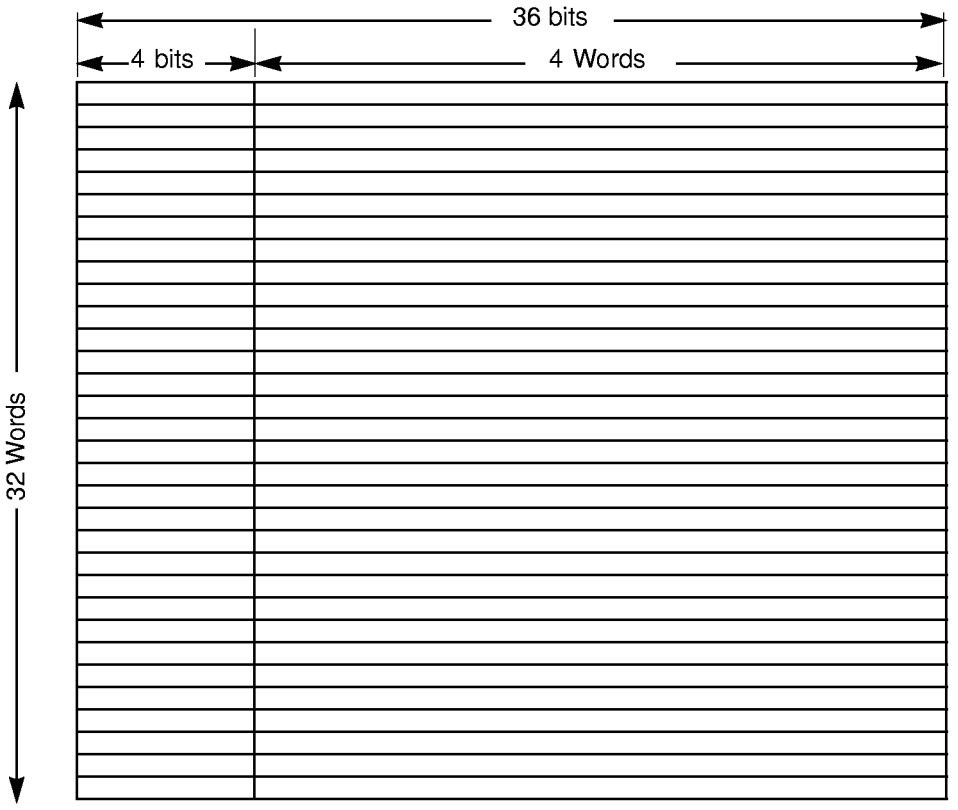
## 4.2 On-chip FIFOs

Each Ethernet port in the L64381 contains a 128-byte Transmit FIFO and a 128-byte Receive FIFO. Each location addresses a word (four bytes) of data, along with the four encoded value bits. The FIFO also maintains the Start-of-Packet and End-of-Packet information, which makes the actual FIFO structure 32 x 36 bits. FIFO pointers are maintained in registers on the L64381. The FIFO access circuit is designed in a way that allows FIFO reads and writes from the Ethernet ports and from the Bus Interface to happen without conflicts or delays.

Figure 4.2 shows the 128-byte FIFO configuration. The lower 32 bits (four words) are data; the most-significant four bits are user-encoded values for $\overline{\text{BYTE\_VALID}}$[3:0], $\overline{\text{SOP}}$, $\overline{\text{EOP}}$, and $\overline{\text{AI\_FCS\_IN}}$ when a bus interface transaction occurs. The RRAM and WRAM fields in the Data FIFO Address Register correspond to the upper four bits in the FIFO for reads and writes, respectively. Refer to Section 2.7, “Data FIFO Regis-

ters," for more information on accessing the Data FIFOs using the Data FIFO registers.

Figure 4.2
Data FIFO Block Diagram




### 4.2.1 Transmit FIFO

The Transmit FIFO can be configured to store the first 64 bytes of a packet until the first 64 bytes have been successfully transmitted. This allows the chip to retransmit the packet without requiring a bus access when a collision occurs in the first 51.2 µs. Once the first 64 bytes have been successfully transmitted, the FIFO releases the first 64 bytes, thus allowing utilization of the whole FIFO for storage.

The Transmit FIFO also allows storage of succeeding packets to be transmitted while the previous packet is still being transmitted. This feature allows back-to-back transmissions of packets without any wasted bandwidth on the Ethernet, and also allows better usage of the FIFO.

If a late collision occurs in the transmission of a packet, then the chip generates an interrupt and updates the appropriate Statistics Counters. The Processor can then read the error register, and decide on the appropriate action. If another packet is available for transmission in the

Transmit FIFO, the L64381 starts transmitting it after waiting the appropriate back-off time.

The Transmit FIFO asserts the $\overline{\text{BUF\_AVAIL}}$ signal when it has at least as many entries empty as specified in the Bus Transfer Size field (BUS). This assertion signals the system that the L64381 can accommodate the Bus Transfer Size amount of data into its Transmit FIFO. The system can, at this point, write packet data into the FIFO. The Transmit FIFO initiates a transmission on the Ethernet when it detects that the FIFO has at least as many words as specified by the Transmit Threshold field, or the Transmit FIFO detects an $\overline{\text{EOP}}$ stored in the FIFO. This situation can arise when the Auto Pad feature is enabled for the port. The system must guarantee that it can place into the FIFO data to be transmitted faster than the time taken to transmit the data already in the FIFO.

The Transmit Threshold field provides a means of delaying the start of transmission for slower systems. The Transmit FIFO sets the Transmit FIFO Underrun Error bit in the Error Register if the FIFO detects an underrun condition. The Transmit FIFO can never overrun, because the Bus-In State Machine (described in Section 4.3.2, "Bus-In State Machine") does not request a write into the Transmit FIFO until it has enough room for the data.

### 4.2.2 Receive FIFO

The Receive FIFO can be configured to initiate a bus transfer when the Bus Transfer Size threshold has been met, or if the Receive FIFO detects an $\overline{\text{EOP}}$ stored in the FIFO. This situation can arise when the packet size is not a multiple of the Bus Transfer Size, or the packet is a runt packet. The Receive FIFO initiates this transfer by setting the $\overline{\text{PKT\_AVAIL}}$ signal. This threshold can be programmed to be 4, 8, or 16 words. A threshold setting of four words allows transmission of the packet with minimal latency, while a setting of 16 words allows greater bus bandwidth (since fewer idle cycles need to be inserted between transfers). The Receive FIFO sets the Receive FIFO Overrun Error bit in the Error register if the FIFO detects an overrun condition. The Receive FIFO can never underrun, since the Bus-Out State Machine (4.3.1, "Bus-Out State Machine,") does not request a read from the Receive FIFO until it has enough data in the FIFO.

## 4.3 Bus Interface

The L64381 Bus Interface is comprised of two state machines: Bus Out and Bus In. This section describes both state machines in detail.

### 4.3.1 Bus-Out State Machine

The Bus Interface has four Bus-Out State Machines, one for each Ethernet port. Each Bus-Out State Machine is responsible for transferring data received on the port to other devices via the Bus Interface. When a word of data has been received from the Ethernet, the L64381 places it into the Receive FIFO. When the FIFO contains as many words as specified in the Bus Transfer Size field of Configuration Register 1 (or if the FIFO contains an EOP), the Bus-Out State Machine asserts the appropriate $\overline{\text{PKT\_AVAIL\_PORTx}}$ signal, where x equals the port number (0 to 3).

The $\overline{\text{PKT\_AVAIL\_PORTx}}$ signals from the four on-chip Ethernet ports are multiplexed on the $\overline{\text{PKT\_AVAIL}}$ signal on the device pins. The multiplexing is done with respect to the $\overline{\text{STROBE}}$ input. When the L64381 receives the assertion of the $\overline{\text{STROBE}}$ input, it places the states of the $\overline{\text{PKT\_AVAIL\_PORTx}}$ signals onto the device pin in the following four cycles, with PORT 0 placed first, and PORT 3 placed last. Figure 4.3 shows the timing relationship between these signals.

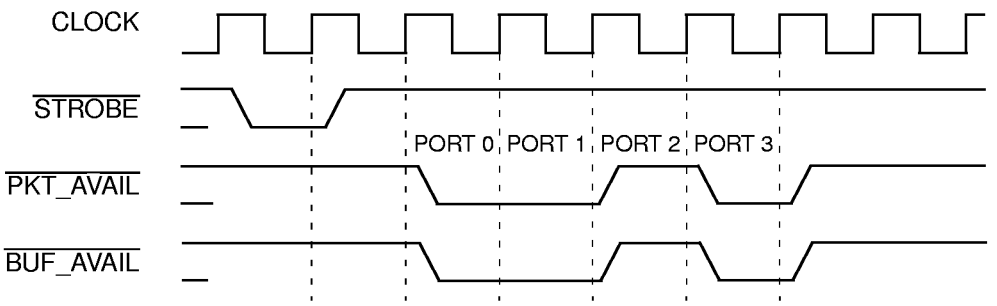
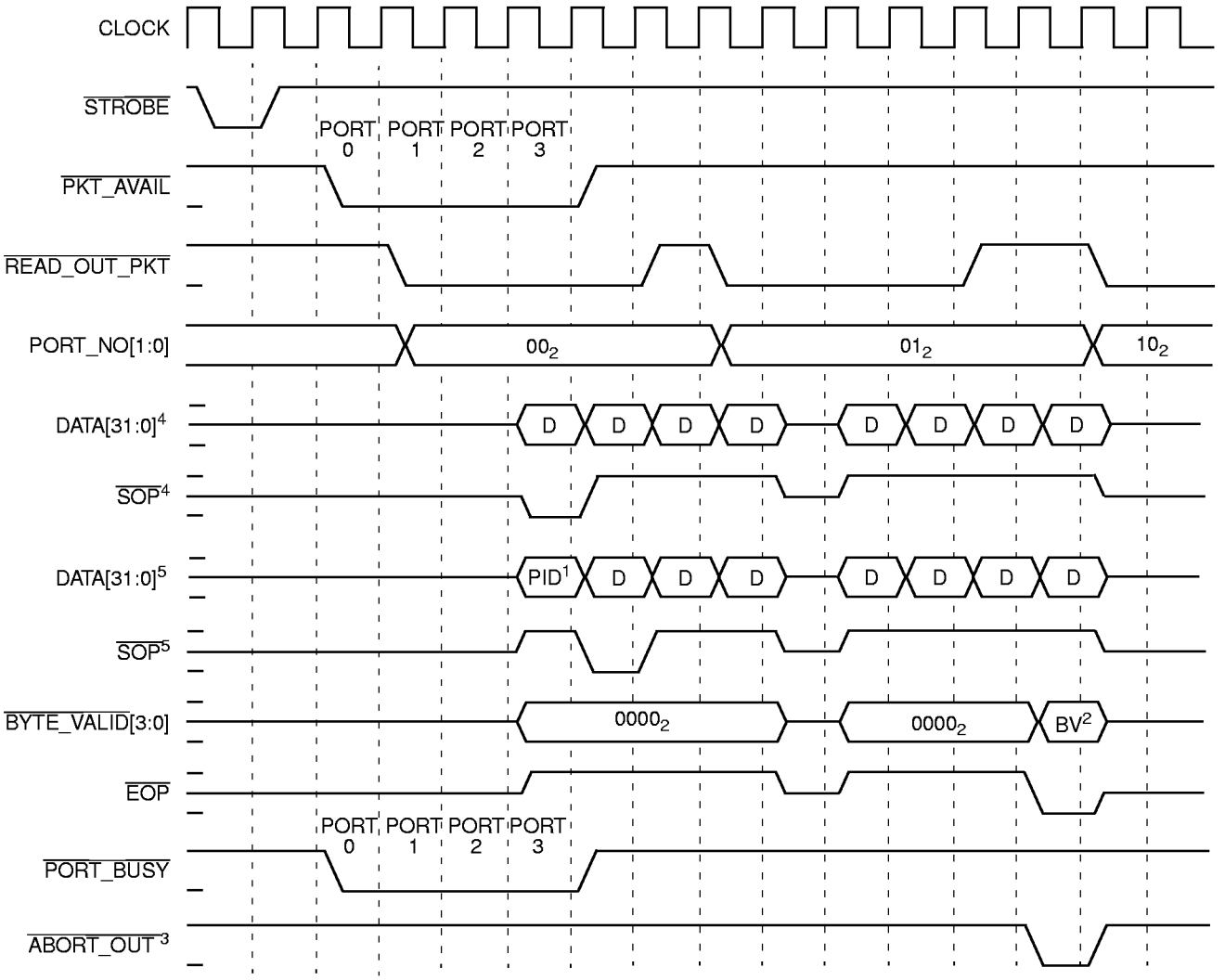



Figure 4.3 $\overline{\text{STROBE}}$ to $\overline{\text{PKT\_AVAIL}}$ Timing Relationship

When the Bus-Out State Machine receives assertion of the $\overline{\text{READ\_OUT\_PKT}}$ signal with the corresponding port number on the PORT_NO[1:0] inputs, it starts sending data on to the DATA[31:0], $\overline{\text{BYTE\_VALID[3:0]}}$, $\overline{\text{SOP}}$, and $\overline{\text{EOP}}$ outputs in the following cycle. The L64381 drives these output signals in the cycle following detection of the $\overline{\text{READ\_OUT\_PKT}}$ signal assertion. The external device will be unable to latch the data on the third cycle. Refer to the functional waveforms for the cycle count between signal transitions when such conflict occurs.

The state machine fetches data from the head of the Receive FIFO, and transfers the number of words specified in the Bus Transfer Size field. Figure 4.4 shows the details of the handshaking during a bus read.

Figure 4.4
Reading Packets from
the L64381



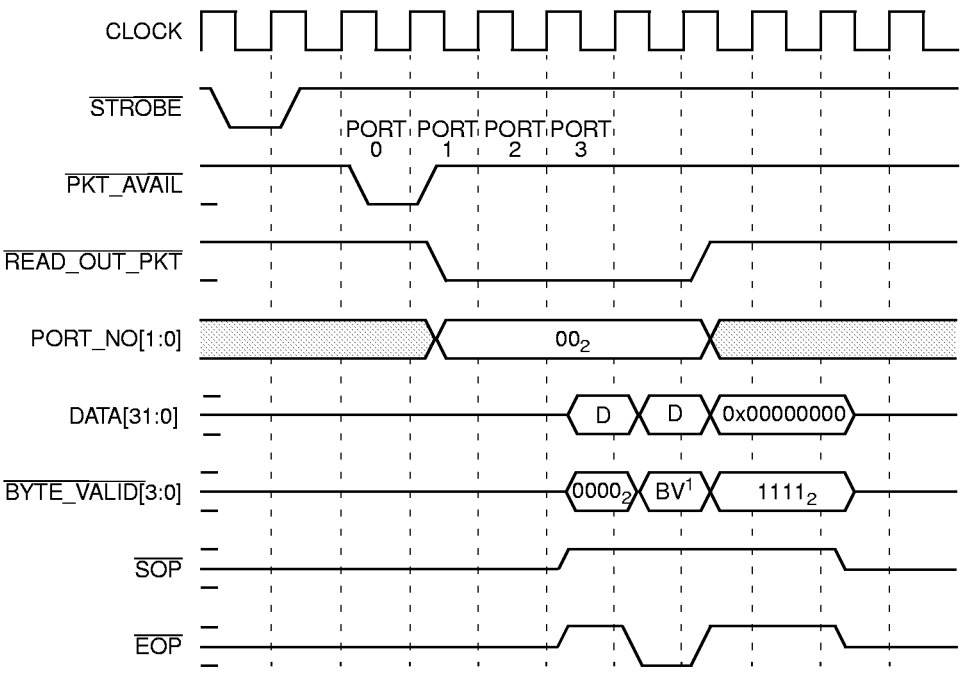

1. The first word contains the port ID# when SOPCFG bit 4 of Configuration Register 1 is set to 1.
2. Valid values for BYTE_VALID[3:0] are $0000_2$, $1000_2$, $1100_2$ and $1110_2$.
3. When the L64381 detects dribble bits, it asserts $\overline{\text{ABORT\_OUT}}$ and $\overline{\text{EOP}}$ with BYTE_VALID[3:0] = $1111_2$
4. SOPCFG, Bit 4 of Configuration Register 1 = 0
5. SOPCFG, Bit 4 of Configuration Register 1 = 1

If the data being transferred on the bus is the start of a new packet, then the state machine also asserts the $\overline{\text{SOP}}$ output (Port 0 in Figure 4.4). Similarly, if the data being transmitted is the end of a packet, then the state machine asserts the $\overline{\text{EOP}}$ signal (Port 1 in Figure 4.4). Note that

the $\overline{SOP}$ always occurs in the first or second word of a transfer, but the $\overline{EOP}$ can occur anywhere within the transfer. The L64381 may assert $\overline{ABORT\_OUT}$ due to a failing condition only with the $\overline{EOP}$ during a bus read. $\overline{ABORT\_OUT}$ is asserted only for one cycle.

There should be at least one wait cycle between two consecutive bus reads. Figure 4.4 shows two wait states asserted between reads from Port 1 and Port 2. When the $\overline{EOP}$ signal occurs in the middle of a transfer, the state machine drives idle data (zeros) on the DATA[31:0] outputs and ones on the $\overline{BYTE\_VALID[3:0]}$, $\overline{SOP}$, and $\overline{EOP}$ outputs in the following cycles until it detects a deassertion of the $\overline{READ\_OUT\_PKT}$ signal, at which time it 3-states DATA[31:0], $\overline{BYTE\_VALID[3:0]}$, $\overline{SOP}$, and $\overline{EOP}$. Figure 4.5 shows a bus read from Port 0 with $\overline{EOP}$ assertion in the middle of the transfer.

Figure 4.5
Read Packet with $\overline{EOP}$ in the Middle of a Transfer




1. Valid values for BYTE_VALID[3:0] are $0000_2$, $0001_2$, $0011_2$ and $0111_2$

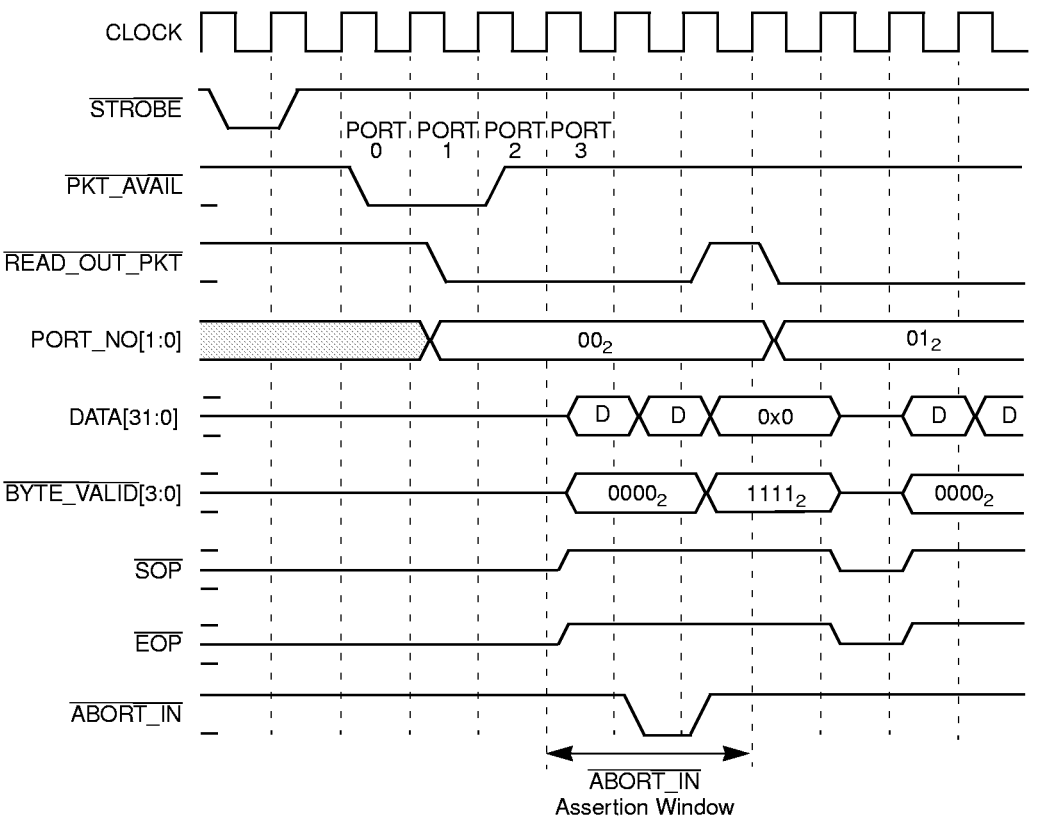
If the current data transfer on the bus does not contain an end of packet, then the state machine transfers the number of words specified by the Bus Transfer Size, and then 3-states the DATA[31:0], $\overline{BYTE\_VALID[3:0]}$, $\overline{SOP}$, and $\overline{EOP}$ signals after the $\overline{READ\_OUT\_PKT}$ signal is deasserted. This operation allows the external arbitration logic to grant the bus to another device without having to wait for the Bus Transfer Size amount of cycles. If the state machine detects that the $\overline{READ\_OUT\_PKT}$ signal

is deasserted before the Bus Transfer Size amount of data has been transferred (and this is not an end of packet), then the state machine assumes that the external arbitration wanted to preempt this transfer, and 3-states the DATA[31:0], $\overline{\text{BYTE\_VALID[3:0]}}$, $\overline{\text{SOP}}$, and $\overline{\text{EOP}}$ signals.

When the next transfer from this FIFO is to be initiated, the state machine starts fetching data from the FIFO location following the one that was last transferred in the previous access. If the L64381 detects the assertion of the $\overline{\text{ABORT\_IN}}$ signal while a bus transfer is in progress, then it stops the current transfer and resets the Receive FIFO head pointer to the value it contained just before the transfer was started. After the asserted $\overline{\text{ABORT\_IN}}$ signal is latched by the L64381, if the $\overline{\text{READ\_OUT\_PKT}}$ is still asserted, the L64381 drives idle data (zeros) on the DATA[31:0] outputs and ones on the $\overline{\text{BYTE\_VALID[3:0]}}$, $\overline{\text{SOP}}$, and $\overline{\text{EOP}}$ outputs throughout the current cycle. To abort a read transfer, $\overline{\text{ABORT\_IN}}$ should be asserted so that it is latched at one of the clock edges in which the L64381 puts data on the data bus. $\overline{\text{ABORT\_IN}}$ needs to be asserted for one cycle only, and it is ignored if asserted after $\overline{\text{EOP}}$.

Figure 4.6 shows a read transfer that is aborted by asserting $\overline{\text{ABORT\_IN}}$. It also shows the window in which $\overline{\text{ABORT\_IN}}$ should be asserted.

Figure 4.6
Read Data with ABORT_IN Assertion



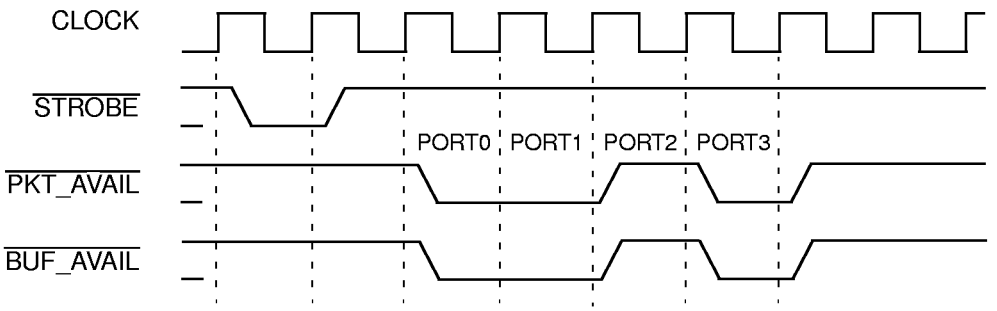

The arbitration chip receives the assertion of the PKT_AVAIL signal and grants the request by asserting the READ_OUT_PKT and PORT_NO[1:0] signals. In response to these inputs, the Bus-Out State Machine drives the DATA[31:0], BYTE_VALID[3:0], SOP, and EOP signals. After the state machine writes the Bus Transfer Size number of words, it 3-states the DATA[31:0], BYTE_VALID[3:0], SOP, and EOP signals.

Because the arbitration chip knows that the L64381 only drives the bus for Bus Transfer Size cycles, it can pipeline the READ_OUT_PKT and PORT_NO[1:0] signals to different devices to ensure minimum idle cycles.

### 4.3.2 Bus-In State Machine

There are four Bus-In State Machines on the chip, one per Ethernet port. The Bus-In State Machine is responsible for receiving data that is to be transmitted on the Ethernet. When the state machine detects that the Transmit FIFO has at least as many empty locations as specified by the Bus Transfer Size, it drives the BUF_AVAIL signal active for the corresponding port. This signal is multiplexed onto the BUF_AVAIL device output in the same manner as the PKT_AVAIL signal. Figure 4.7 shows the timing relationship between the STROBE and BUF_AVAIL signals.

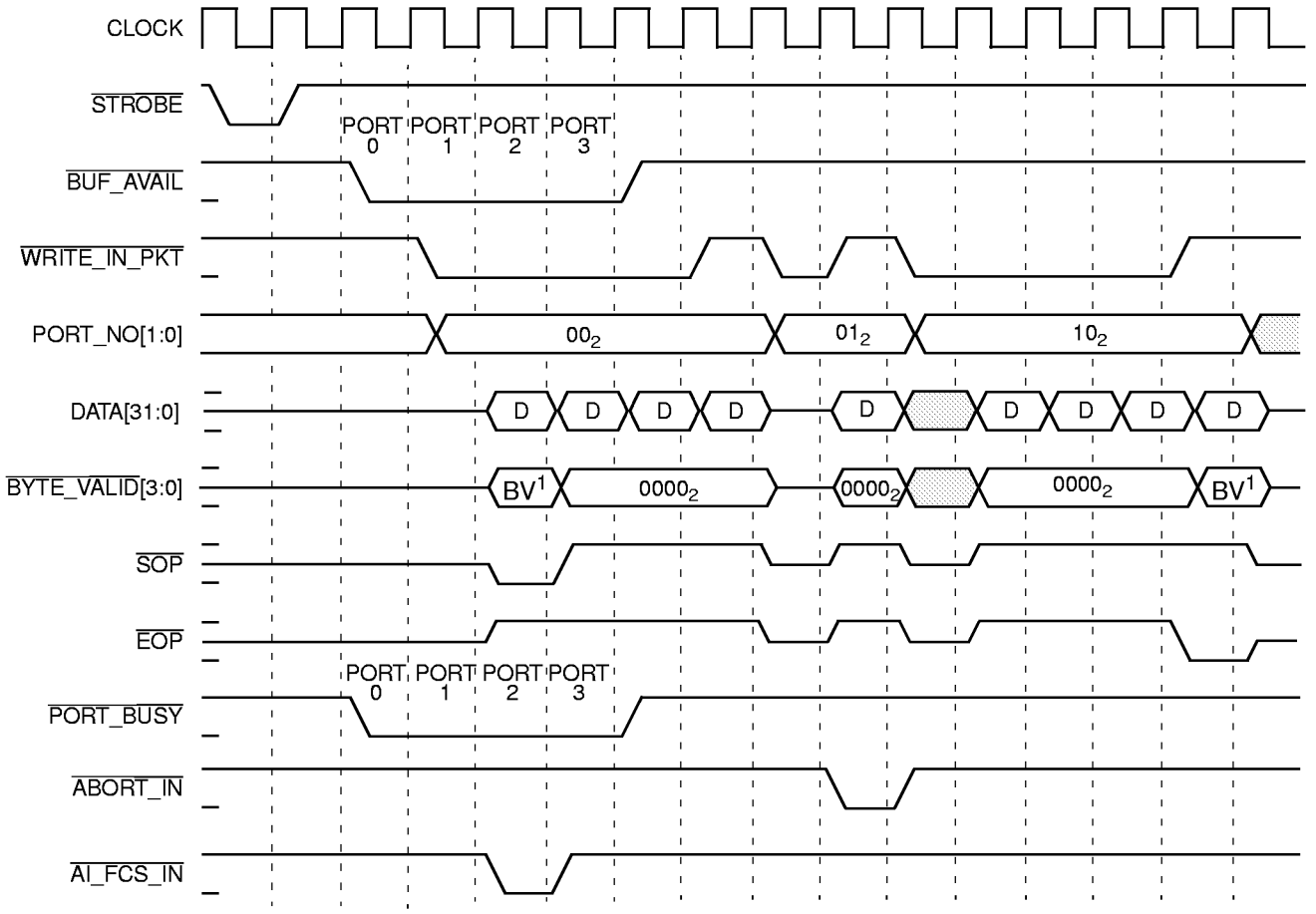
Figure 4.7
$\overline{\text{STROBE}}$ to
$\overline{\text{BUF\_AVAIL}}$ Timing




When the Bus-In State Machine receives a $\overline{\text{WRITE\_IN\_PKT}}$ signal and its corresponding port number is on the PORT_NO[1:0] inputs, it writes the DATA[31:0] along with the encoded $\overline{\text{BYTE\_VALID[3:0]}}$, $\overline{\text{SOP}}$, and $\overline{\text{EOP}}$ signals into the Transmit FIFO using the tail pointer (refer to Chapter 2, “Registers,” for more information on the tail pointer). If the state machine detects either an asserted $\overline{\text{EOP}}$ or a deasserted $\overline{\text{WRITE\_IN\_PKT}}$ signal on its input, it stops writing any more data into the Transmit FIFO after that cycle.

Figure 4.8 shows the details of the handshaking during a bus write. There should be at least one wait cycle between two consecutive bus writes.

Figure 4.8
Writing Packets to the
L64381 with Port 1
Operation Aborted

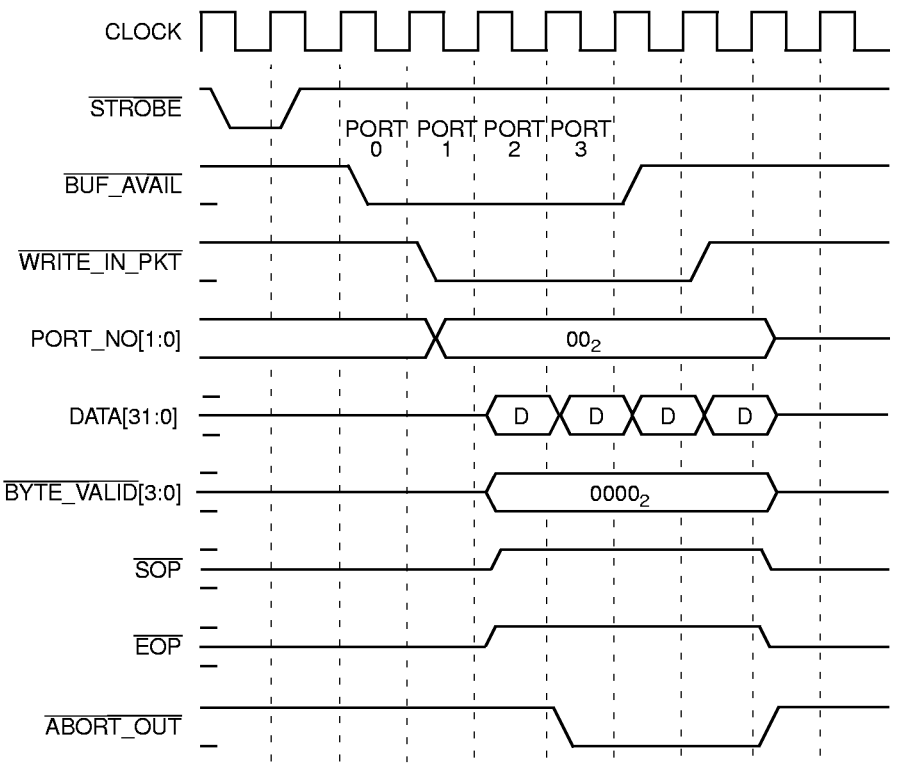



1. Valid values for BYTE_VALID[3:0] are $0000_2$, $0001_2$, $0011_2$, $0111_2$, $1000_2$, $1100_2$ and $1110_2$.

If the L64381 detects the assertion of the $\overline{\text{ABORT\_IN}}$ signal while a bus write is in progress, then it restores the Transmit FIFO Tail Pointer to the value it contained just before the transfer was started. $\overline{\text{ABORT\_IN}}$ should be asserted so that it is latched at one of the clock edges in which the L64381 latches data from the data bus. It is ignored if asserted after the assertion of $\overline{\text{EOP}}$. $\overline{\text{ABORT\_IN}}$ is asserted for one cycle only. When $\overline{\text{ABORT\_IN}}$ is asserted, the $\overline{\text{WRITE\_IN\_PKT}}$ can be terminated at the same time. Figure 4.8 shows the $\overline{\text{ABORT\_IN}}$ assertion for Port 1 write operation, which has no effect on the following Port 2 operation.

The L64381 may assert $\overline{\text{ABORT\_OUT}}$, due to a failing condition, during a bus write. $\overline{\text{ABORT\_OUT}}$ is deasserted only after $\overline{\text{WRITE\_IN\_PKT}}$ is deasserted. Figure 4.9 shows the assertion of $\overline{\text{ABORT\_OUT}}$ during a write transfer in Port 0.

Figure 4.9
Writing Packets to the L64381 with $\overline{\text{ABORT\_OUT}}$ Asserted




The arbitration chip receives assertion of the $\overline{\text{BUF\_AVAIL}}$, and grants the request by asserting the $\overline{\text{WRITE\_IN\_PKT}}$ and PORT_NO[1:0] signals when it has data to write to the L64381. The source of the transfer then drives the DATA[31:0], $\overline{\text{BYTE\_VALID[3:0]}}$, $\overline{\text{SOP}}$, and $\overline{\text{EOP}}$ signals, and the Bus-In State Machine writes the data on the bus into the Transmit FIFO. After the state machine writes the Bus Transfer Size number of words, it stops writing to the FIFO.

Because the arbitration chip knows that the L64381 only drives the bus for Bus Transfer Size cycles, it can pipeline the $\overline{\text{WRITE\_IN\_PKT}}$ and PORT_NO[1:0] signals to different devices to ensure minimum latency.

Figure 4.10 shows a read transfer followed by a write, and Figure 4.11 illustrates a write transfer followed by a read. Figure 4.10 shows that at least two cycles are needed between the deassertion of $\overline{\text{READ\_OUT\_PKT}}$ and the assertion of $\overline{\text{WRITE\_IN\_PKT}}$ when a read is followed by a write. This is to prevent contention for the data bus. Figure 4.11 shows that there should be at least one cycle between the

deassertion of $\overline{\text{WRITE\_IN\_PKT}}$ and the assertion of $\overline{\text{READ\_OUT\_PKT}}$, so that PORT_NO[1:0] is properly asserted throughout the entire write cycle.

Figure 4.10
Read-then-Write
Operations

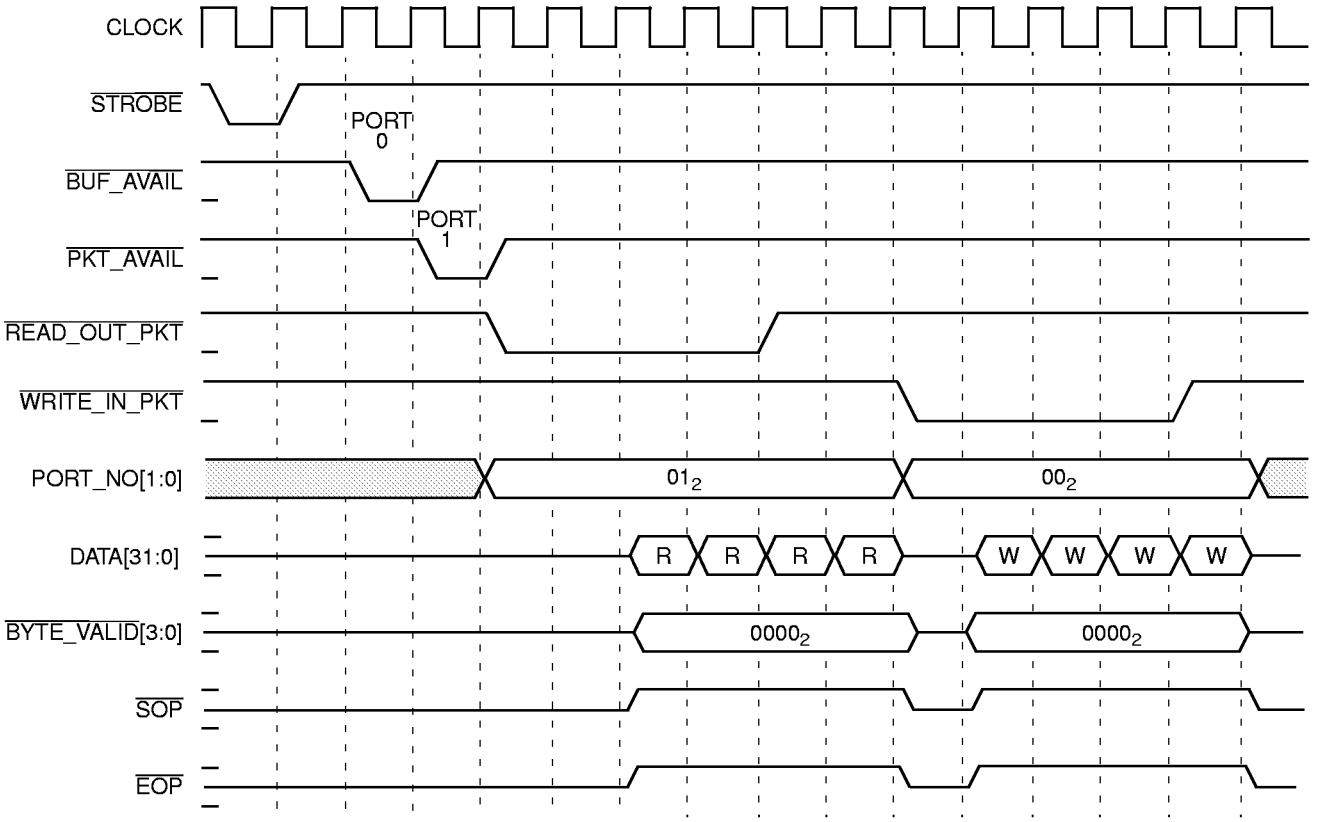
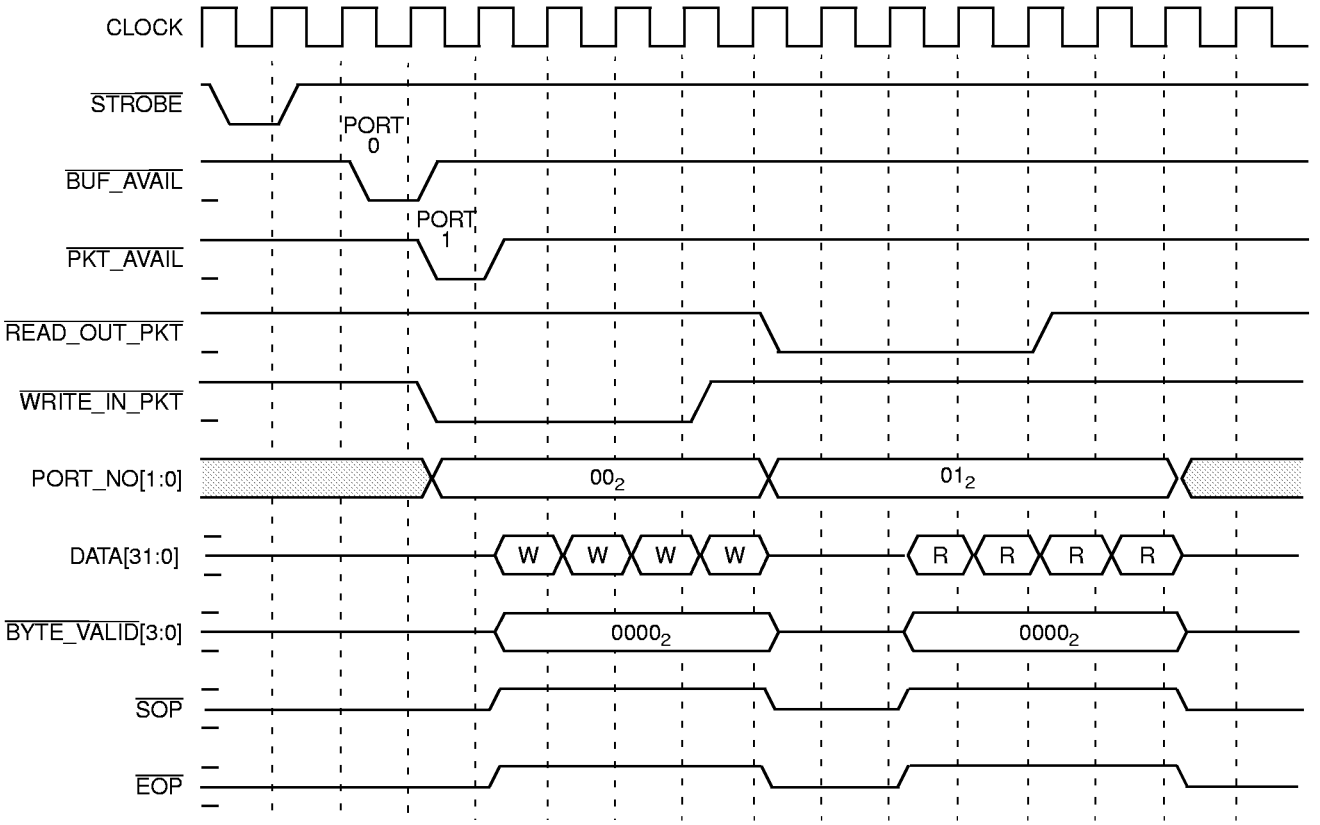


Figure 4.11
Write-then-Read
Operations




## 4.4 Processor Interface

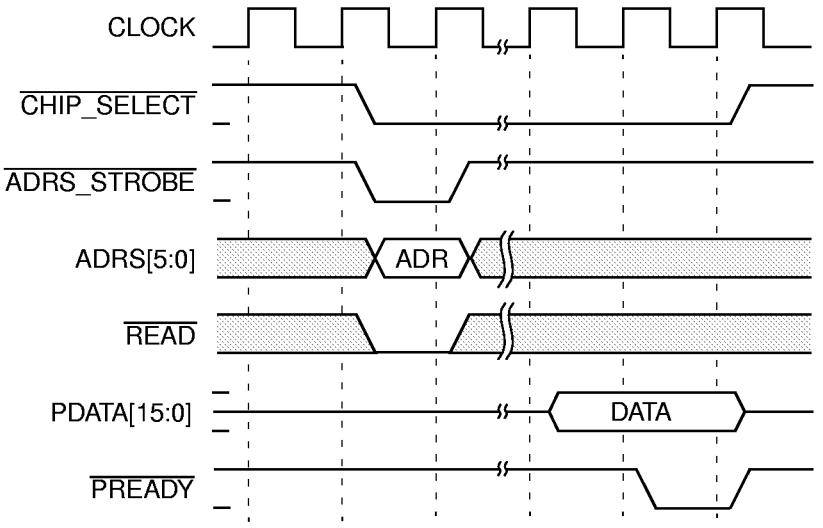
The Processor Interface is used to configure the L64381, to read the port statistics, and to read and write the internal registers of the L64381. Since the statistics counters are a shared resource between the processor interface and the internal network interface, the internal network interface has higher priority over the processor interface. The access time for all processor accesses to the statistics counters may vary. Refer to the description of $\overline{\text{CHIP\_SELECT}}$ on page 3-12 for more information.

### 4.4.1 Reading the L64381 Internal Registers

The processor asserts the $\overline{\text{CHIP\_SELECT}}$, $\overline{\text{READ}}$, and $\overline{\text{ADRS\_STROBE}}$ signals to read the internal registers of the L64381. When the chip detects both $\overline{\text{ADRS\_STROBE}}$ and $\overline{\text{CHIP\_SELECT}}$ asserted, then on the next rising edge of CLOCK, it latches in the address from the ADRS[5:0] inputs, and latches in the instruction type from the $\overline{\text{READ}}$ input. The L64381 asserts the $\overline{\text{PREADY}}$ signal for one cycle after it has placed the read data onto the PDATA[15:0] bus.

Figure 4.12 shows the timing relationship for read accesses.

Figure 4.12
Processor Read
Timing

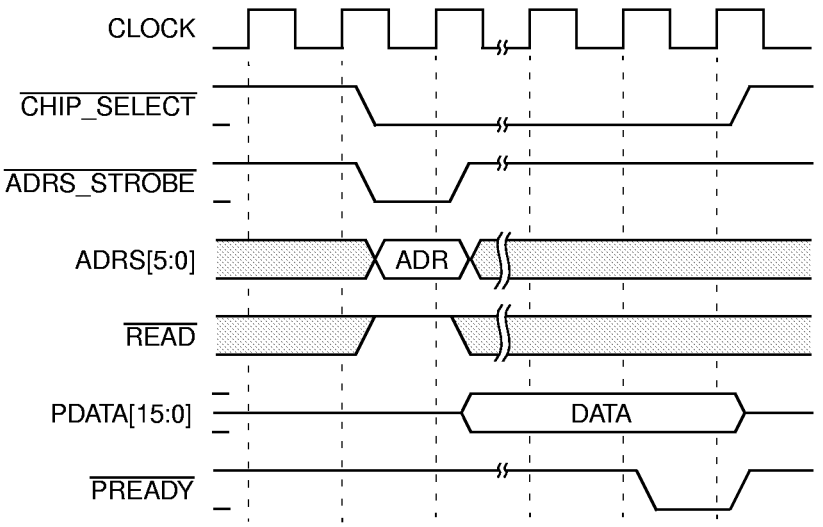



### 4.4.2 Writing the L64381 Internal Registers

To write the internal registers of the L64381, the processor asserts the CHIP_SELECT and ADRS_STROBE signals and deasserts the READ signal. When the L64381 detects both ADRS_STROBE and CHIP_SELECT asserted, then on the next rising edge of CLOCK, it latches in the address from the ADRS[5:0] inputs, and latches in the instruction type from the READ input. The L64381 asserts the PREADY signal for one cycle after it has latched in the data on PDATA[15:0].

Figure 4.13 shows the timing relationship for write accesses.

Figure 4.13
Processor Write
Timing

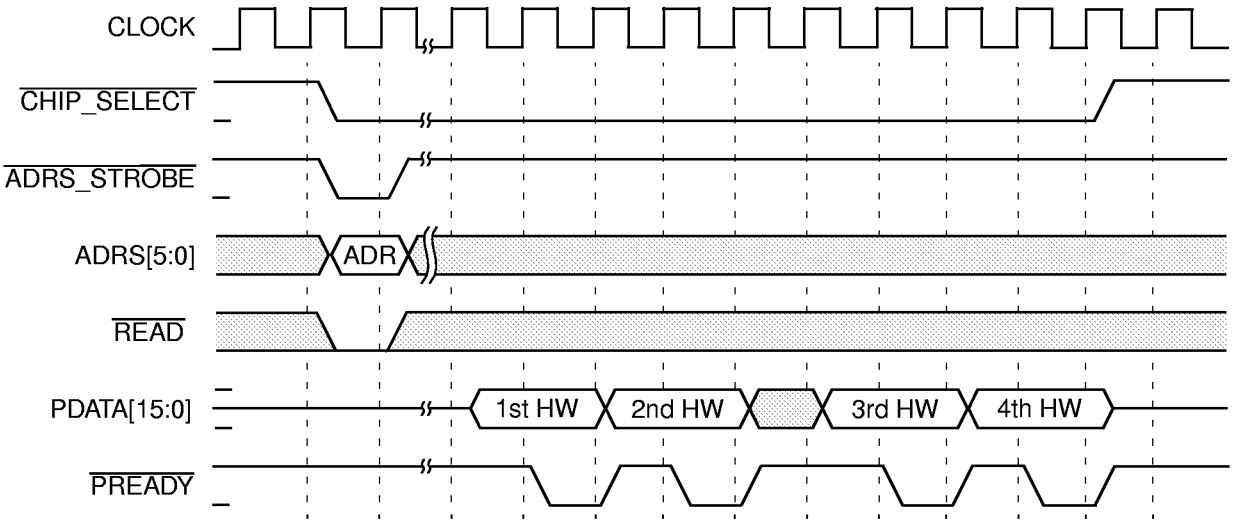


### 4.4.3 Burst Mode Reads of the Statistics Counters

For burst mode read accesses from the Statistics Counters, the L64381 places new data on the PDATA[15:0] bus, and toggles the $\overline{\text{PREADY}}$ signal for one cycle every time it places new data on the PDATA[15:0] bus.

Figure 4.14 shows the timing relationship for burst mode accesses for a burst size of four halfwords. Note that even though the figure shows new data being placed on the PDATA[15:0] bus every other clock cycle, the actual number of cycles that elapse between successive data being placed on the PDATA[15:0] bus can vary. The $\overline{\text{PREADY}}$ signal indicates when data is valid on the PDATA[15:0] bus.

Figure 4.14
Burst Read Access
Timing to the L64381

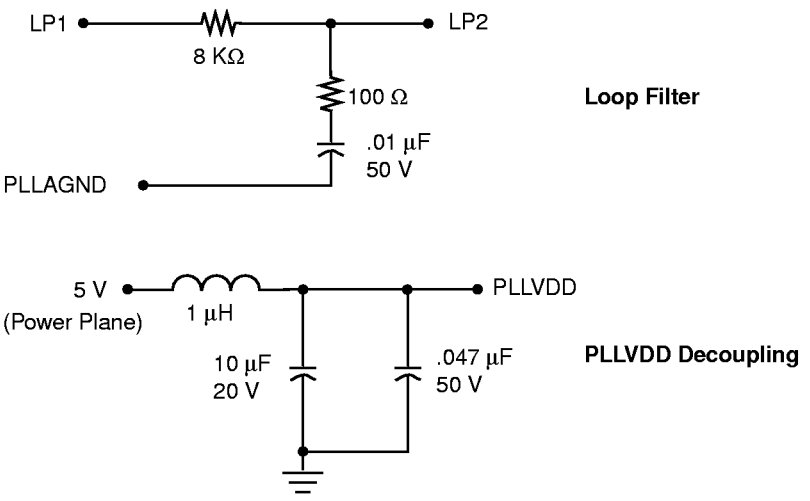



## 4.5 PLL Connections

Each of the four ports within the L64381 contains a digital phase-locked loop (DPLL) for data recovery. The 80-MHz clock that drives the DPLL is synthesized using an analog PLL on the L64381. The analog PLL consists of a phase comparator, a voltage-controlled oscillator, and an external loop filter. The reference clock input is a 20-MHz clock (CLOCK_20MHZ).

Figure 4.15 shows the external loop filter circuit, which connects to the internal analog PLL. It also shows the PLLVDD decoupling circuit.

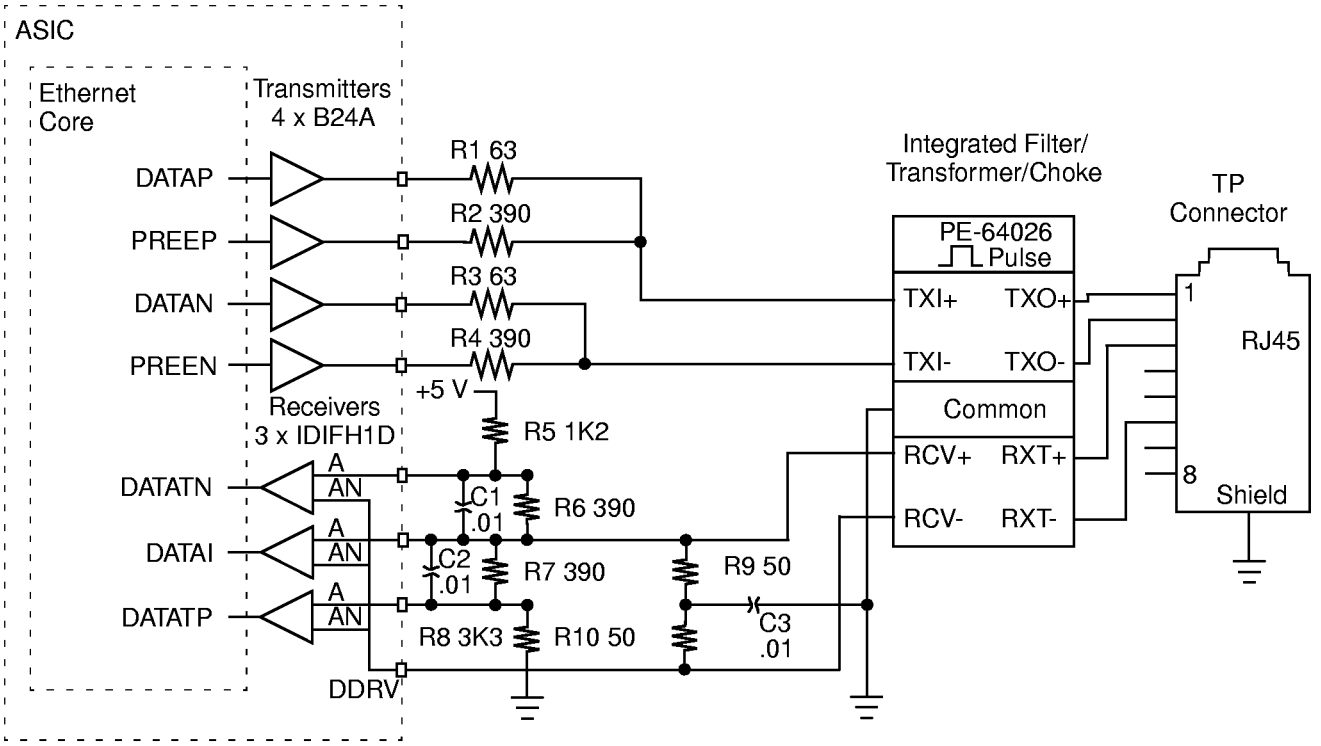
Figure 4.15
Loop Filter and
PLLVDD
Decoupling




## 4.6 Twisted-Pair Interface

Figure 4.16 shows how an L64381 port can be connected to a 10Base-T twisted-pair connector.

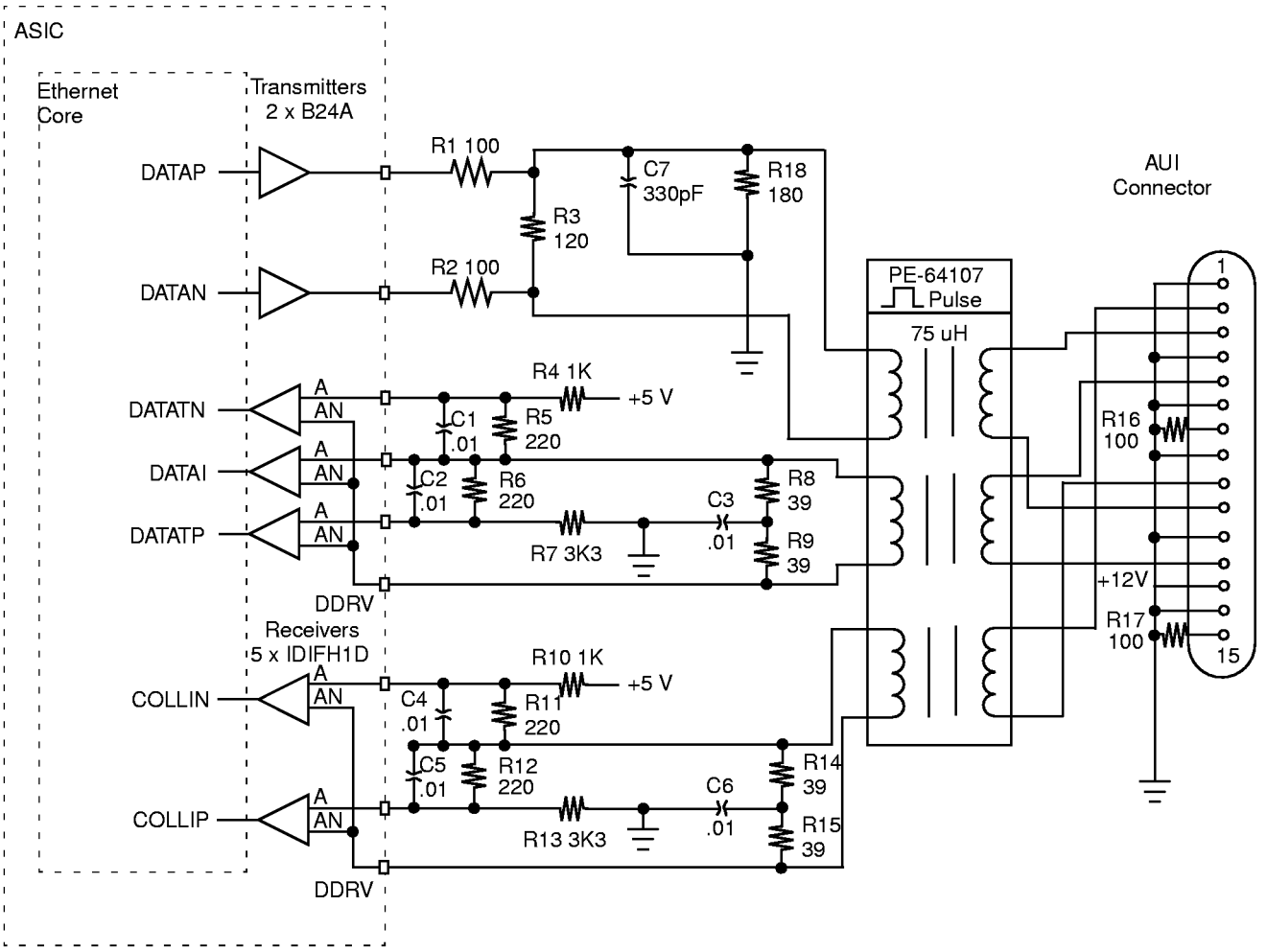
Figure 4.16
Typical L64381 Twisted-
Pair Application



## 4.7 AUI Interface

Figure 4.17 shows how an L64381 port can be connected to an AUI connector.

Figure 4.17
Typical L64381 AUI Application




## 4.8 JTAG Initialization

To ensure proper operation of the L64381, the internal JTAG circuitry must be disabled. The L64381 uses the $\overline{\text{TRSTN}}$ signal for this purpose. Asserting $\overline{\text{TRSTN}}$ LOW asynchronously initializes the TAP controller to the Test-Logic-Reset state. Holding $\overline{\text{TRSTN}}$ LOW keeps the TAP controller in this state and prevents the JTAG circuitry from interfering with the L64381's other on-chip logic. $\overline{\text{TRSTN}}$ may be connected to GND to guarantee a permanent LOW signal. See Section 3.6, "Test Signals", for more information on the $\overline{\text{TRSTN}}$ signal.

# Chapter 5
# Statistics Counters

The L64381 provides the following statistics maintained in 25 32-bit counters. In this section, unless otherwise stated, the size of a packet includes the Source and Destination Fields, the Length or Type field, the Data and the Frame Check Sequence fields, but does not include the Preamble and Start of Frame Delimiter fields. All Statistics are updated after an attempt to transfer or receive the packet is made, regardless of whether the attempt was successful or not. The Statistics Counters must be initialized before they can be used. To initialize the counters, write all zeros to all memory locations in the Statistics Counters. Each counter has a unique address that is stored in the Statistics Counter Address Register. Table 5.1 lists all the Statistics Counters, their contents, and brief descriptions.

Table 5.1
Descriptions of
Statistics Counters

| Counter # | Contents | Description |
|---|---|---|
| 0 | Number of bytes received | This counter counts all bytes within packets received from the receiver. |
| 1 | Number of internal receive errors | This counter counts the number of packets that were lost because of internal errors in the L64381, and that affect this particular Ethernet connection. These errors are FIFO overrun and PLL. |
| 2 | Number of packets received | This counter counts the total number of packets received from the receiver, regardless of whether they were received with or without error. |
| 3 | Number of broadcast packets received | This counter counts the number of packets received without errors, and that also had a broadcast address in the destination field of the packet (all ones). |
| 4 | Number of multicast packets received | This counter counts the number of packets received without errors, and also that had a multicast address in the destination field of the packet. This number does not include broadcast packets. |
| 5 | Number of receive CRC errors | This counter counts the number of packets received between 64 bytes and 1518 bytes in length, but were not an integral number of bytes in length, or had a bad Frame Check Sequence. |
| 6 | Number of runt packets received | This counter counts the number of packets less than 64 bytes long that were an integral number of bytes long. |
| 7 | Number of oversized packets received | This counter counts the number of packets more than 1518 bytes long that were an integral number of bytes long. |
| 8 | Number of total collisions | This counter counts the number of collisions detected when transmitting in the half-duplex mode. |
| 9 | Number of 64 byte packets received | This counter counts all packets received by the L64381 that were 64 bytes in length. |
| 10 | Number of 65 to 127 byte packets received | This counter counts all packets received by the L64381 that were 65 to 127 bytes in length. |
| 11 | Number of 128 to 255 byte packets received | This counter counts all packets received by the L64381 that were 128 to 255 bytes in length. |

(Sheet 1 of 2)

Table 5.1 (Cont.)
Descriptions of
Statistics Counters

| Counter # | Contents | Description |
|---|---|---|
| 12 | Number of 256 to 511 byte packets received | This counter counts all packets received by the L64381 that were 256 to 511 bytes in length. |
| 13 | Number of 512 to 1023 byte packets received | This counter counts all packets received by the L64381 that were 512 to 1023 bytes in length. |
| 14 | Number of 1024 to 1518 byte packets received | This counter counts all packets received by the L64381 that were 1024 to 1518 bytes in length. |
| 15 | Number of internal transmit errors | This counter counts the number of internal errors in the L64381, and affect this particular Ethernet connection. Typical errors would be FIFO underrun errors or invalid data in the $\overline{\text{BYTE\_VALID}}$[3:0], $\overline{\text{SOP}}$, or $\overline{\text{EOP}}$ fields. |
| 16 | Number of late collisions | This counter counts the number of collisions after the first 64 bytes of the packet were successfully transmitted on the Ethernet. |
| 17 | Number of runt packets transmitted | This counter counts the number of packets transmitted from the L64381 that were less than 64 bytes in length. |
| 18 | Number of greater than 1518 byte packets transmitted | This counter counts all packets transmitted from the L64381 that were greater than 1518 bytes in length and were an integral number of bytes long. |
| 19 | Number of packets successfully transmitted | This counter counts all packets successfully transmitted from the L64381. |
| 20 | Number of bytes successfully transmitted | This counter counts the total number of bytes successfully transmitted from the L64381. |
| 21 | Number of multicast packets transmitted | This counter counts the total number of multicast packets transmitted from the L64381. |
| 22 | Number of broadcast packets transmitted | This counter counts the total number of broadcast packets transmitted from the L64381. |
| 23 | Reserved | |
| 24 | Reserved | |

(Sheet 2 of 2)

# Chapter 6
# Instruction Set

Chapter 6 lists the instructions that can be sent to the L64381 on the Processor Interface. Most instructions read and write the L64381 internal registers. The L64381 also provides special instructions that facilitate transfers of statistical information and that control and observe registers and FIFOs. Refer to Chapter 2, "Registers," for more details on the register descriptions.

This chapter contains the following sections:

- Section 6.1, "Instruction Set Summary"
- Section 6.2, "Accessing the Data FIFO"
- Section 6.3, "Accessing the Statistics Counters"

## 6.1 Instruction Set Summary

Table 6.1 lists all of the L64381 instructions. The instructions are ordered according to instruction type: read/write register, read and autoincrement, and burst read. Certain instructions operate only in specific Diagnostic Modes. Table 6.1 also specifies the particular operating mode(s).

Table 6.1
L64381 Instructions

| Opcode | $\overline{\text{READ}}$ | Valid Mode(s) | Operation |
|---|---|---|---|
| **Read/Write Register Instructions** | | | |
| 000000 | 0 | All | Read NOP |
| 000000 | 1 | All | Write NOP |
| 000010 | 0 | All | Read Configuration Register 0 |
| 000010 | 1 | All | Write Configuration Register 0 |
| 000011 | 0 | All | Read Configuration Register 1 |
| 000011 | 1 | All | Write Configuration Register 1 |
| 000100 | 0 | All | Read Ethernet Address Register 0 |

(Sheet 1 of 3)

Table 6.1 (Cont.)
L64381 Instructions

| Opcode | $\overline{\text{READ}}$ | Valid Mode(s) | Operation |
|---|---|---|---|
| **Read/Write Register Instructions (Cont.)** | | | |
| 000100 | 1 | All | Write Ethernet Address Register 0 |
| 000101 | 0 | All | Read Ethernet Address Register 1 |
| 000101 | 1 | All | Write Ethernet Address Register 1 |
| 000110 | 0 | All | Read Ethernet Address Register 2 |
| 000110 | 1 | All | Write Ethernet Address Register 2 |
| 000111 | 0 | All | Read Packet Configuration Register |
| 000111 | 1 | All | Write Packet Configuration Register |
| 001000 | 0 | All | Read Error Register 0 |
| 001000 | 1 | All | Write Error Register 0 |
| 001010 | 0 | All | Read Error Register 1 |
| 001010 | 1 | All | Write Error Register 1 |
| 001001 | 0 | All | Read Error Mask Register 0 |
| 001001 | 1 | All | Write Error Mask Register 0 |
| 001011 | 0 | All | Read Error Mask Register 1 |
| 001011 | 1 | All | Write Error Mask Register 1 |
| 010110 | 0 | All | Read Statistics Counters Address Register |
| 010110 | 1 | All | Write Statistics Counters Address Register |
| 100000 | 0 | All | Read Receive FIFO Head Pointer Register 0 |
| 100000 | 1 | 1, 3[1] | Write Receive FIFO Head Pointer Register 0 |
| 100001 | 0 | All | Read Receive FIFO Head Pointer Register 1 |
| 100001 | 1 | 1, 3[1] | Write Receive FIFO Head Pointer Register 1 |
| 100010 | 0 | All | Read Receive FIFO Tail Pointer Register 0 |
| 100010 | 1 | 1, 3[1] | Write Receive FIFO Tail Pointer Register 0 |
| 100011 | 0 | All | Read Receive FIFO Tail Pointer Register 1 |
| 100011 | 1 | 1, 3[1] | Write Receive FIFO Tail Pointer Register 1 |
| 100100 | 0 | All | Read Transmit FIFO Head Pointer Register 0 |
| 100100 | 1 | 1, 3[1] | Write Transmit FIFO Head Pointer Register 0 |
| 100101 | 0 | All | Read Transmit FIFO Head Pointer Register 1 |
| 100101 | 1 | 1, 3[1] | Write Transmit FIFO Head Pointer Register 1 |
| 100110 | 0 | All | Read Transmit FIFO Tail Pointer Register 0 |
| 100110 | 1 | 1, 3[1] | Write Transmit FIFO Tail Pointer Register 0 |
| 100111 | 0 | All | Read Transmit FIFO Tail Pointer Register 1 |
| 100111 | 1 | 1, 3[1] | Write Transmit FIFO Tail Pointer Register 1 |
| 010000 | 0 | All | Read Statistics Counters Data In Register 0 |

(Sheet 2 of 3)

Table 6.1 (Cont.)
L64381 Instructions

| Opcode | $\overline{\text{READ}}$ | Valid Mode(s) | Operation |
|---|---|---|---|
| **Read/Write Register Instructions (Cont.)** | | | |
| 010000 | 1 | All | Write Statistics Counters Data In Register 0 |
| 010001 | 0 | All | Read Statistics Counters Data In Register 1 |
| 010001 | 1 | All | Write Statistics Counters Data In Register 1 |
| 010010 | 0 | All | Read Statistics Counters Data Out Register 0 |
| 010011 | 0 | All | Read Statistics Counters Data Out Register 1 |
| 011000 | 0 | All | Read Data FIFO Data In Register 0 |
| 011000 | 1 | 1, 3[1] | Write Data FIFO Data In Register 0 |
| 011001 | 0 | All | Read Data FIFO Data In Register 1 |
| 011001 | 1 | 1, 3[1] | Write Data FIFO Data In Register 1 |
| 011010 | 0 | All | Read Data FIFO Data Out Register 0 |
| 011011 | 0 | All | Read Data FIFO Data Out Register 1 |
| 011110 | 0 | All | Read Data FIFO Address Register |
| 011110 | 1 | All | Write Data FIFO Address Register |
| 101000 | 0 | All | Read Receive EOP Counter 0 |
| 101000 | 1 | 1, 3[1] | Write Receive EOP Counter 0 |
| 101001 | 0 | All | Read Receive EOP Counter 1 |
| 101001 | 1 | 1, 3[1] | Write Receive EOP Counter 1 |
| 101010 | 0 | 1, 3[1] | Read Transmit EOP Counter 0 |
| 101010 | 1 | All | Write Transmit EOP Counter 0 |
| 101011 | 0 | All | Read Transmit EOP Counter 1 |
| 101011 | 1 | 1, 3[1] | Write Transmit EOP Counter 1 |
| **Read and Autoincrement Instructions** | | | |
| 110100 | 0 | All | Read Single Statistics Counter |
| 110100 | 1 | All | Write Single Statistics Counter |
| 111000 | 0 | All | Read Data FIFO |
| 111000 | 1 | 1, 3[1] | Write Data FIFO |
| **Burst Read Instructions** | | | |
| 110011 | 0 | All | Burst Read Statistics Counters |

(Sheet 3 of 3)

1. These instructions are provided for testing purposes, and should be used only when the L64381 is in Diagnostics Modes 1 and 3. However, the L64381 does not restrict the use of these instructions in other modes. The result of these instructions is not guaranteed in other modes.

Care must be taken when modifying the configuration of the L64381 while it is actively transmitting or receiving a packet on the Network Interface or on the Bus Interface. A change in configuration can lead to unpredictable or incorrect operation. Thus the configuration modes should only be changed when the L64381 is either in Diagnostics Modes 1 or 3 or is idle in Diagnostic Modes 0 or 2.

---

## 6.2 Accessing the Data FIFO

This section details the instructions that read and write the contents of the Data FIFO. These operations are possible only in Modes 1 and 3.

### 6.2.1 Reading the Data FIFO

The following steps describe how to read a location in the Data FIFO:

1. Load the FIFO address to be read (use the Write Data FIFO Address Register instruction).
2. Execute the Read Data FIFO instruction.
3. Read the least-significant halfword from Data FIFO Data Out Register 0.
4. Read the most-significant halfword from Data FIFO Data Out Register 1.
5. Read bits [15:12] of the Data FIFO Address Register (the RRAM field), which contain bits [35:32] of the Data FIFO.

### 6.2.2 Writing to the Data FIFO

The following steps describe how to write to a location in the Data FIFO:

1. Load the FIFO address to be written (use the Write Data FIFO Address Register instruction). At the same time, load bits [11:8] of the Data FIFO Address Register (the WRAM field) with the desired value in bits [35:32] of the Data FIFO.
2. Load the least-significant halfword into Data FIFO Data In Register 0.
3. Load the most-significant halfword into Data FIFO Data In Register 1.
4. Execute the Write Data FIFO instruction.

## 6.3 Accessing the Statistics Counters

This section details the three types of instructions that access the Statistics Counters: read, burst read, and write.

### 6.3.1 Reading the Statistics Counters

The following steps describe how to read a single Statistics Counter:

1. Load the Statistics Counter number to be read (use the Write Statistics Counters Address Register instruction).
2. Execute the Read Single Statistics Counter instruction. When in autoincrement mode, this instruction causes the Statistics Counters Address Register to autoincrement it's value after the read has been performed.
3. Read the least-significant halfword from Statistics Counters Data-Out Register 0.
4. Read the most-significant halfword from Statistics Counters Data-Out Register 1.

The data read is placed into the PDATA[15:0] bus. Because the Statistics FIFO is 32-bits wide and the PDATA[15:0] bus is 16-bits wide, two Read Single Statistics Counter instructions are executed. These instructions facilitate reads of statistics from the L64381 if the processor is not capable of performing burst mode read operations.

### 6.3.2 Writing the Statistics Counters

The following steps describe how to write a single Statistics Counter:

1. Load the Statistics Counter number to be written (use the Write Statistics Counters Address Register instruction).
2. Load the least-significant halfword into Statistics Counters Data-In Register 0.
3. Load the most-significant halfword into Statistics Counters Data-In Register 1.
4. Execute the Write Single Statistics Counter instruction. This instruction causes the data in the Statistics Counters Data-In Registers to be loaded into the Statistics FIFO, and the address in the Statistics Counters Address Register to be autoincremented. This feature is provided in autoincrement mode to reduce the number of instructions needed to test the Statistics Counters.

### 6.3.3 Burst Read of the Statistics Counters

The L64381 Statistics Counters can be read from the Processor Interface by using Burst Read instructions. The following steps describe how to implement the burst read:

1. Load the Statistics Counter number to be read (use the Write Statistics Counters Address Register instruction).
2. Execute the Burst Read Statistics Counter instruction. Upon detecting this instruction, the L64381 automatically fetches the statistics for a given port, and places the data in the Statistics Counters Data-Out registers. The L64381 then places the data from these registers on the PDATA[15:0] bus. The number of counters read depends on the value programmed in the Burst Size Field of Configuration Register 1.

   The $\overline{\text{PREADY}}$ signal is toggled after each register data is placed on the PDATA[15:0] bus in order to delimit when data is valid on the bus. Using a burst mode fetch of statistics allows the Processor Interface to fetch statistics without having to generate separate read accesses.

# Chapter 7
# Specifications

This chapter provides the electrical and mechanical specifications for the L64381. This chapter is useful for hardware engineers who are interconnecting the L64381 with other devices and for PCB designers who are designing a printed circuit board.

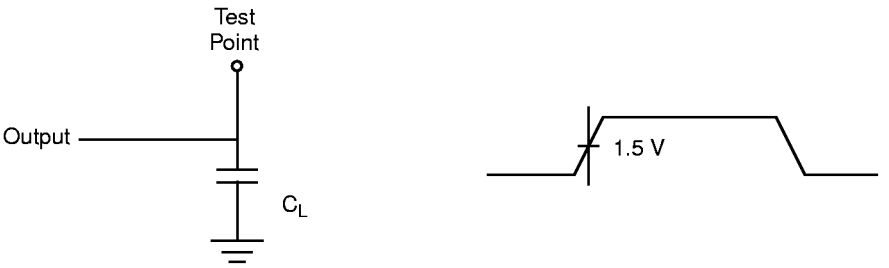
This chapter contains four sections:

- Section 7.1, “AC Timing”
- Section 7.2, “Electrical Requirements”
- Section 7.3, “Special VDD and VSS Connections”
- Section 7.4, “Packaging”

## 7.1 AC Timing

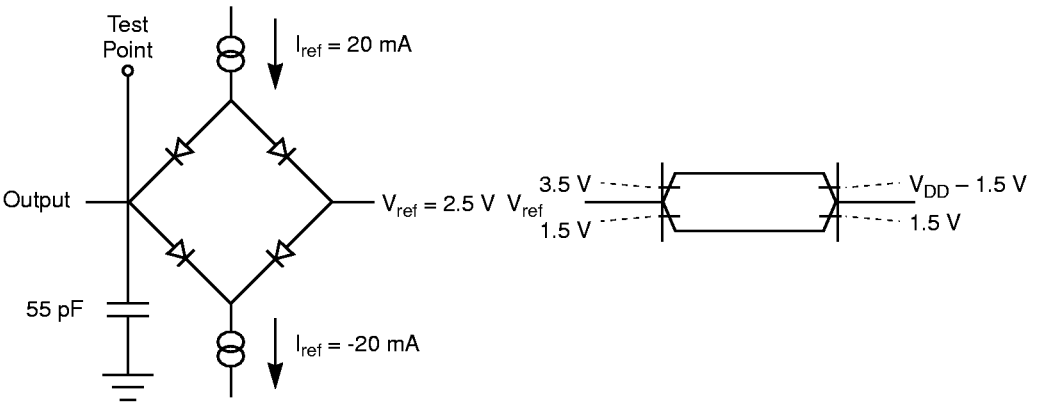
This section describes the AC timing characteristics of the L64381 interface. During AC testing, HIGH inputs are driven at 3.0 V, and LOW inputs are driven at 0 V. For transitions between HIGH, LOW, and invalid states, timing measurements are made at 1.5 V, as shown in Figure 7.1. The test load, $C_L$, for each output signal is given in Table 7.1.

Figure 7.1
AC Test Load and Waveform for Standard Outputs




For three-state outputs, timing measurements are made from the point at which the output turns ON or OFF. An output is ON when its voltage is greater than 3.5 V, or less than 1.5 V. An output is OFF when its voltage is less than $V_{DD}$ – 1.5 V, or greater than 1.5 V, as shown in Figure 7.2.

Figure 7.2
AC Test Load and
Waveform for
3-State Outputs




The timing relationships between signals are depicted in Figures 7.3 through 7.9. The figures depict:

- Figure 7.3, “Bus Interface Read and Write Timing”
- Figure 7.4, “Abort Operation Timing”
- Figure 7.5, “Reset Timing”
- Figure 7.6, “Processor Interface Timing”
- Figure 7.7, “Interrupt Timing”
- Figure 7.8, “Status Timing”
- Figure 7.9, “Clock Timing”

Table 7.1 lists the AC timing for the L64381. The numbers in Figures 7.3 through 7.9 correspond to the timing parameters listed in column 1 of the table. All parameters are valid for the specified ambient temperature range up to 70 °C. These parameters are preliminary and subject to change.

Table 7.1
AC Timing Values

| Parameter | Description | Min | Max | Units | Load (pF) |
|---|---|---|---|---|---|
| 1. | $\overline{\text{STROBE}}$ Setup to CLOCK | 3 | – | ns | – |
| 2. | $\overline{\text{STROBE}}$ Hold from CLOCK | 2.5 | – | ns | – |
| 3. | $\overline{\text{PKT\_AVAIL}}$ Delay from CLOCK | – | 16 | ns | 50 |
| 4. | $\overline{\text{READ\_OUT\_PKT}}$ Setup to CLOCK | 3 | – | ns | – |
| 5. | $\overline{\text{READ\_OUT\_PKT}}$ Hold from CLOCK | 1.5 | – | ns | – |
| 6. | PORT_NO[1:0] Setup to CLOCK | 5 | – | ns | – |
| 7. | PORT_NO[1:0] Hold from CLOCK | 1 | – | ns | – |
| 8. | DATA[31:0] Delay from CLOCK During Read | – | 18 | ns | 50 |
| 9. | DATA[31:0] 3-state from CLOCK | – | 16 | ns | 50 |
| 10. | $\overline{\text{BYTE\_VALID}}$[3:0] Delay from CLOCK During Read | – | 21 | ns | 50 |
| 11. | $\overline{\text{BYTE\_VALID}}$[3:0] 3-state from CLOCK During Read | – | 21 | ns | 50 |
| 12. | $\overline{\text{SOP}}$ Delay from CLOCK During Read | – | 20 | ns | 50 |
| 13. | $\overline{\text{SOP}}$ 3-state from CLOCK During Read | – | 20 | ns | 50 |
| 14. | $\overline{\text{EOP}}$ Delay from CLOCK During Read | – | 21 | ns | 50 |
| 15. | $\overline{\text{EOP}}$ 3-state from CLOCK During Read | – | 21 | ns | 50 |
| 16. | $\overline{\text{BUF\_AVAIL}}$ Delay from CLOCK | – | 16 | ns | 50 |
| 17. | $\overline{\text{WRITE\_IN\_PKT}}$ Setup to CLOCK | 3 | – | ns | – |
| 18. | $\overline{\text{WRITE\_IN\_PKT}}$ Hold from CLOCK | 1.5 | – | ns | – |
| 19. | DATA[31:0] Setup to CLOCK During Write | 5 | – | ns | – |
| 20. | DATA[31:0] Hold from CLOCK During Write | 2.5 | – | ns | – |
| 21. | $\overline{\text{BYTE\_VALID}}$[3:0] Setup to CLOCK During Write | 3 | – | ns | – |
| 22. | $\overline{\text{BYTE\_VALID}}$[3:0] Hold from CLOCK During Write | 1.5 | – | ns | – |
| 23. | $\overline{\text{SOP}}$ Setup to CLOCK During Write | 5 | – | ns | – |
| 24. | $\overline{\text{SOP}}$ Hold from CLOCK During Write | 1 | – | ns | – |
| 25. | $\overline{\text{EOP}}$ Setup to CLOCK During Write | 3 | – | ns | – |
| 26. | $\overline{\text{EOP}}$ Hold from CLOCK During Write | 2 | – | ns | – |
| 27. | $\overline{\text{ABORT\_IN}}$ Setup to CLOCK | 3 | – | ns | – |
| 28. | $\overline{\text{ABORT\_IN}}$ Hold from CLOCK | 2 | – | ns | – |
| 29. | $\overline{\text{PORT\_BUSY}}$ Delay from CLOCK | – | 16 | ns | 50 |

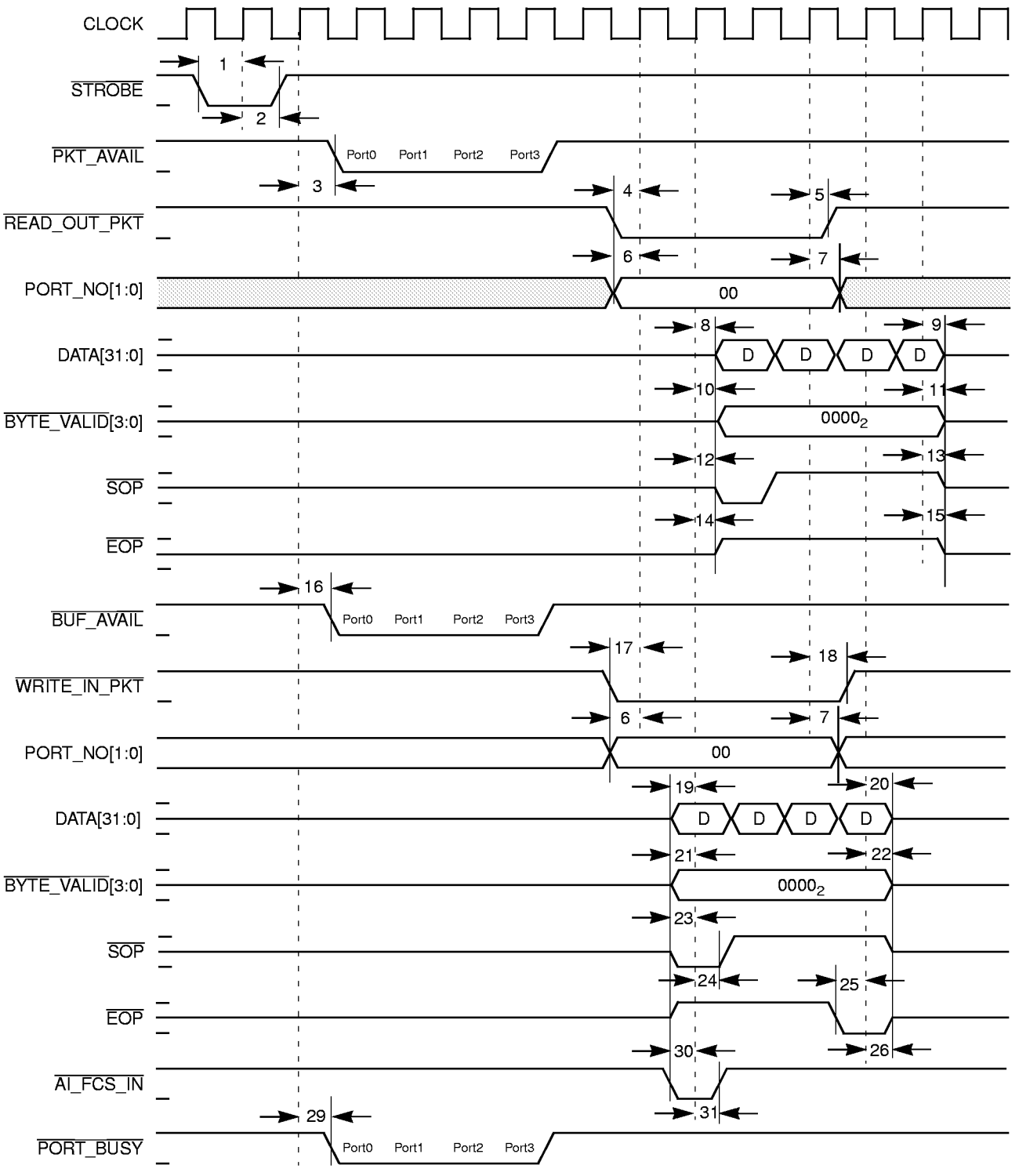
(Sheet 1 of 2)

Table 7.1 (Cont.)
AC Timing Values

| Parameter | Description | Min | Max | Units | Load (pF) |
|---|---|---|---|---|---|
| 30. | $\overline{\text{AI\_FCS\_IN}}$ Setup to CLOCK | 3 | – | ns | – |
| 31. | $\overline{\text{AI\_FCS\_IN}}$ Hold from CLOCK | 2 | – | ns | – |
| 32. | $\overline{\text{ABORT\_OUT}}$ Delay from CLOCK | – | 18 | ns | 50 |
| 33. | $\overline{\text{RESET}}$ Pulse Width | 1 | – | µs | – |
| 34. | $\overline{\text{ADRS\_STROBE}}$ Setup to CLOCK | 3 | – | ns | – |
| 35. | $\overline{\text{ADRS\_STROBE}}$ Hold from CLOCK | 2 | – | ns | – |
| 36. | $\overline{\text{CHIP\_SELECT}}$ Setup to CLOCK | 8 | – | ns | – |
| 37. | $\overline{\text{CHIP\_SELECT}}$ Hold to CLOCK | 2.5 | – | ns | – |
| 38. | $\overline{\text{READ}}$ Setup to CLOCK | 2 | – | ns | – |
| 39. | $\overline{\text{READ}}$ Hold from CLOCK | 1 | – | ns | – |
| 40. | ADRS[5:0] Setup to CLOCK | 2 | – | ns | – |
| 41. | ADRS[5:0] Hold to CLOCK | 3 | – | ns | – |
| 42. | PDATA[15:0] Setup to CLOCK During Write | 4 | – | ns | – |
| 43. | PDATA[15:0] Hold from CLOCK During Write | 1.5 | – | ns | – |
| 44. | $\overline{\text{PREADY}}$ Delay from CLOCK | – | 17 | ns | 50 |
| 45. | PDATA[15:0] Delay from CLOCK During Read | – | 17 | ns | 50 |
| 46. | PDATA[15:0] 3-state from CLOCK During Read if $\overline{\text{PREADY}}$ Asserted or $\overline{\text{CHIP\_SELECT}}$ Sampled Deasserted | – | 17 | ns | 50 |
| 47. | $\overline{\text{INTERRUPT}}$ Delay from CLOCK | – | 16 | ns | 50 |
| 48. | LEDSEL[1:0] Delay from CLOCK | – | 17 | ns | 50 |
| 49. | $\overline{\text{RCVNGN}}$ Delay from CLOCK | – | 17 | ns | 50 |
| 50. | $\overline{\text{TRNSMTN}}$ Delay from CLOCK | – | 17 | ns | 50 |
| 51. | $\overline{\text{COLLOUTN}}$ Delay from CLOCK | – | 17 | ns | 50 |
| 52. | $\overline{\text{LBADN}}$ Delay from CLOCK | – | 16 | ns | 50 |
| 53. | $\overline{\text{LPASSN}}$ Delay from CLOCK | – | 23 | ns | 50 |
| 54. | CLOCK Period | 30 | 62.5 | ns | – |
| 55. | CLOCK High/Low | 5 | – | ns | – |
| 56. | CLOCK_20MHZ Period | 50 | – | ns | – |
| 57. | CLOCK_20MHZ High/Low | 10 | – | ns | – |

(Sheet 2 of 2)

Figure 7.3
Bus Interface Read and
Write Timing




MD95.134

Figure 7.4
Abort Operation Timing

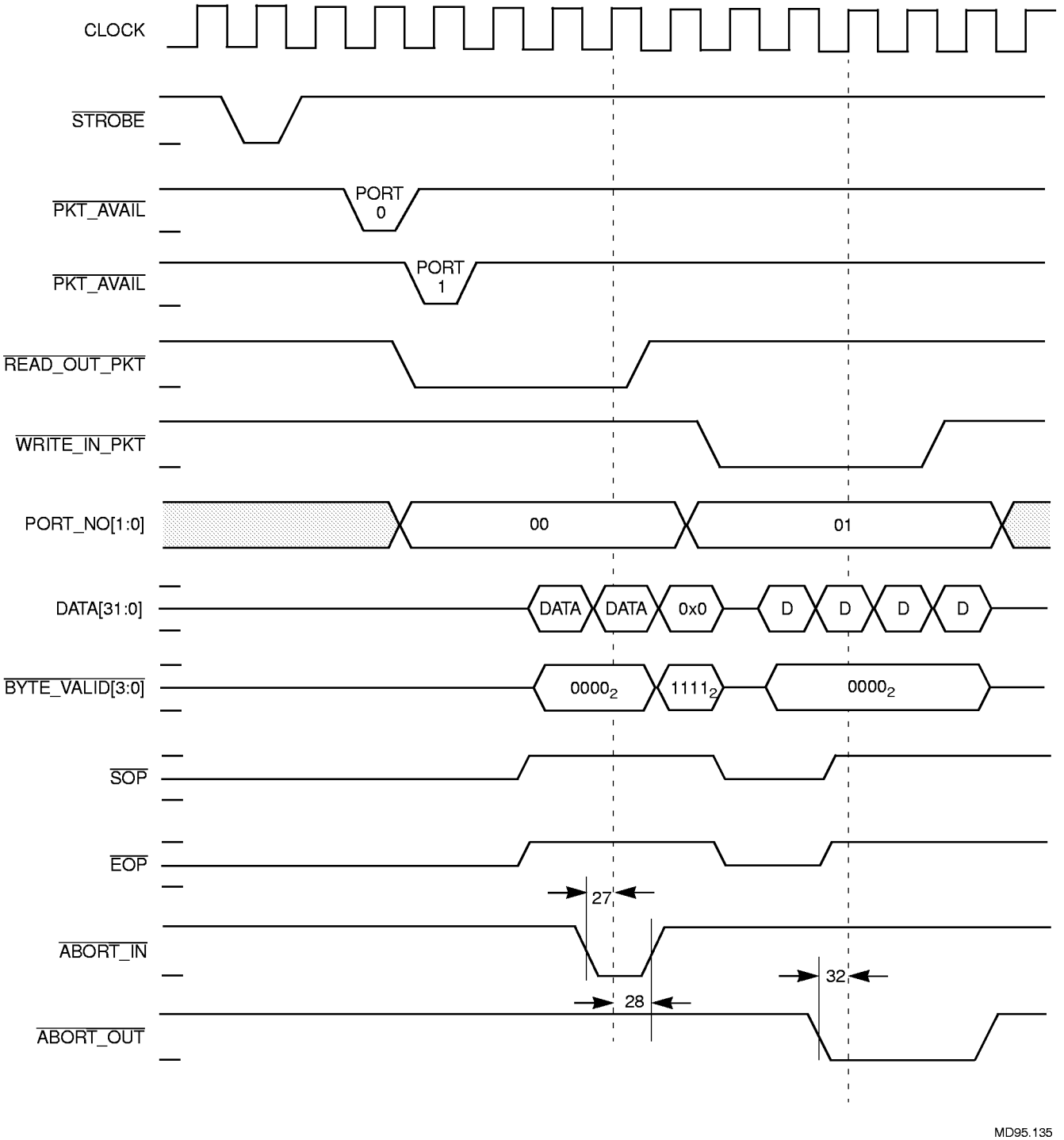


Figure 7.5
Reset Timing

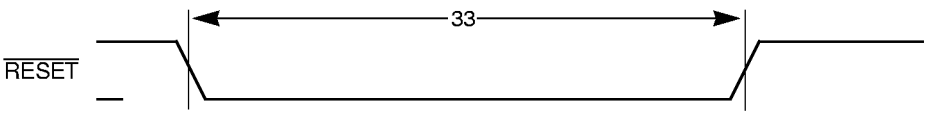



Figure 7.6
Processor Interface
Timing

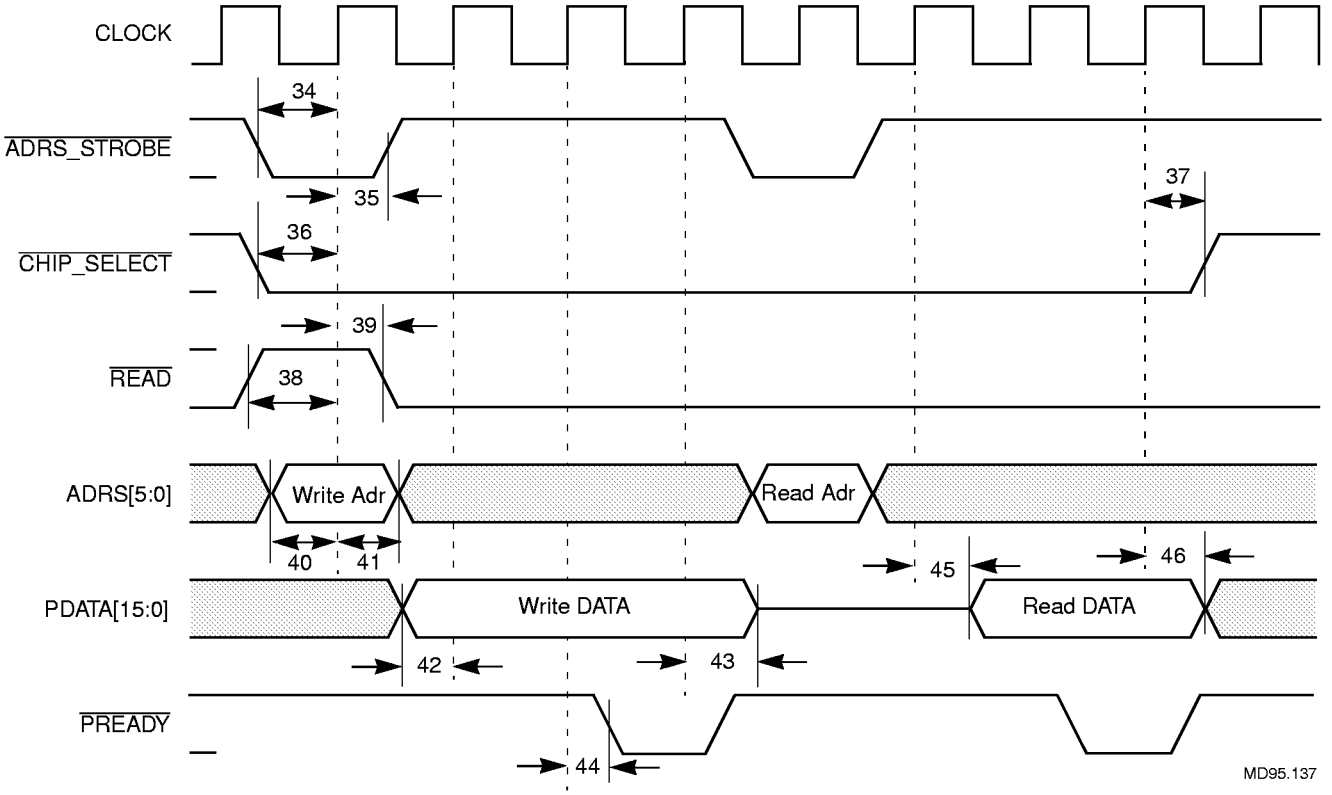



Figure 7.7
Interrupt Timing

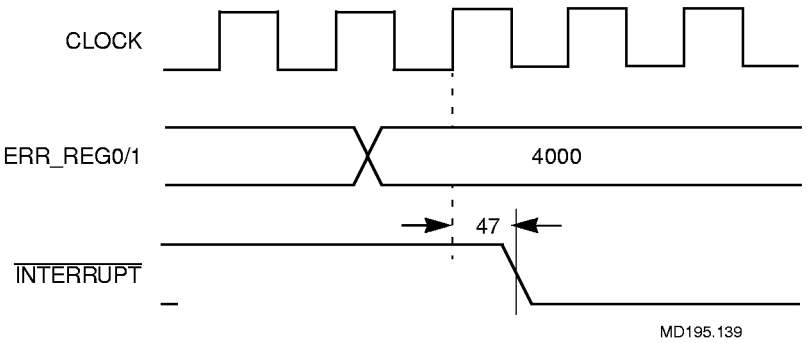


Figure 7.8
Status Timing

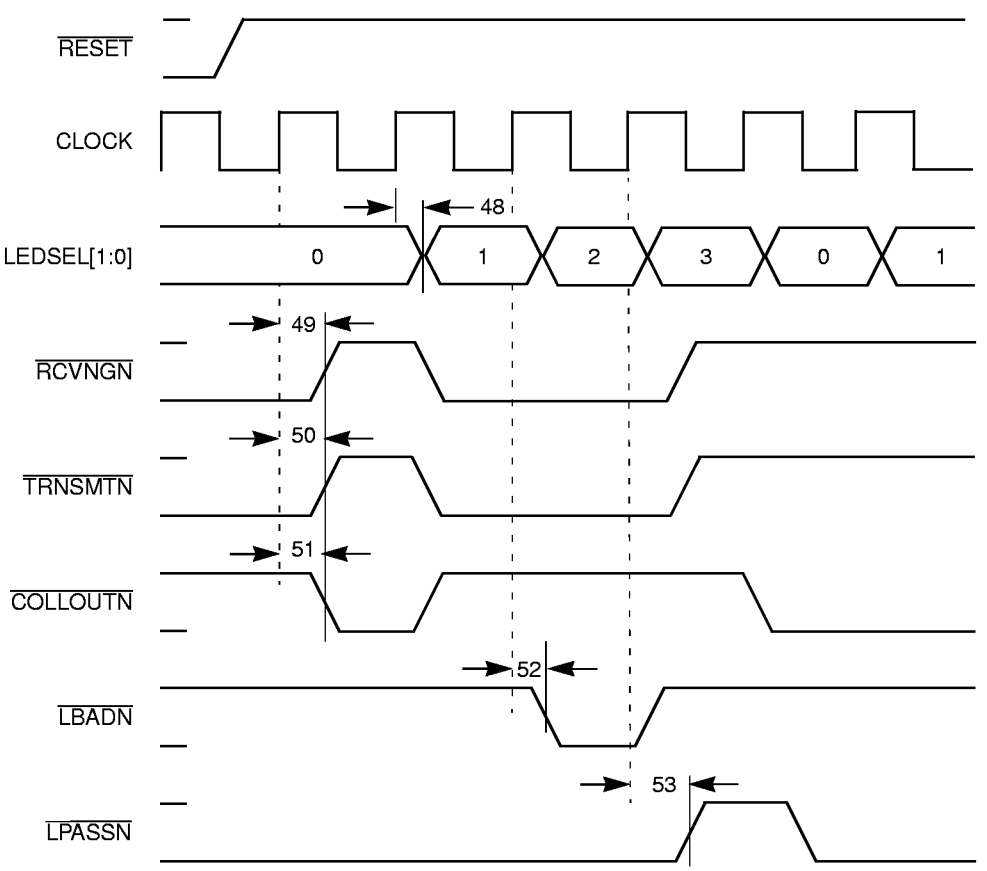
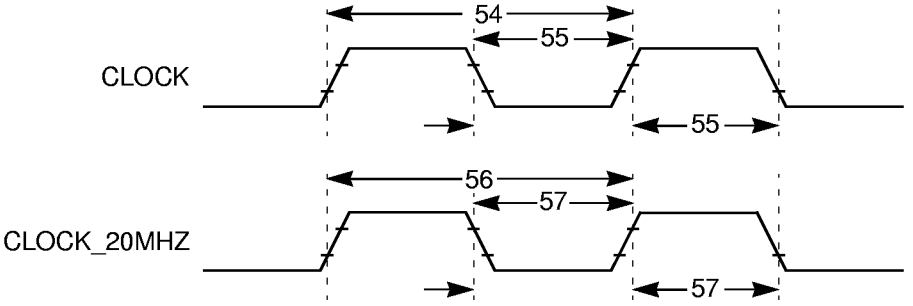



Figure 7.9
Clock Timing




## 7.2 Electrical Requirements

This section specifies the electrical requirements for the L64381. Five tables list electrical data in the following categories:

- Absolute Maximum Ratings (Table 7.2)
- Recommended Operating Conditions (Table 7.3)
- Capacitance (Table 7.4)
- DC Characteristics (Table 7.5)

- Pin Description Summary (Table 7.6)

Table 7.2
Absolute Maximum Ratings

| Symbol | Parameter | Limits[1] | Unit |
|---|---|---|---|
| $V_{DD}$ | DC Supply | -0.3 to +7 | V |
| $V_{IN}$ | Input Voltage | -0.3 to $V_{DD}$ +0.3 | V |
| $I_{IN}$ | DC Input Current | ±10 | mA |
| $T_{STG}$ | Storage Temperature Range, Plastic | -40 to +125 | °C |

1. Referenced to $V_{SS}$.

Table 7.3
Recommended Operating Conditions

| Symbol | Parameter | Limits | Unit |
|---|---|---|---|
| $V_{DD}$ | DC Supply, Commercial | +4.75 to +5.25 | V |
| $T_C$ | Case Temperature | 0 to 85 | ° C |

The thermal characteristics of the 208-pin Plastic Quad Flat Pack (PQFP) are defined as follows:

| | | |
|---|---|---|
| $\theta_{jc}$ = 7.5 °C/Watt | $\theta_{jA}$ = 29.5 °C/Watt | Still Air |
| | $\theta_{jA}$ = 25.8 °C/Watt | 200 lfpm |
| | $\theta_{jA}$ = 24.3 °C/Watt | 400 lfpm |
| | $\theta_{jA}$ = 23.0 °C/Watt | 600 lfpm |

$j_c$ = junction-to-case $j_A$ = junction-to-ambient

The L64381 will operate from 0 to 65.29 °C (110 - 1.84 X 24.3 = 65.29) as the ambient temperature range, with an air flow of 400 lfpm and without a heat sink. Also, the junction temperature should not exceed 110 °C for long-term reliability.

Table 7.4
Capacitance

| Symbol | Parameter[1] | Min | Typ | Max | Unit |
|---|---|---|---|---|---|
| $C_{IN}$ | Input Capacitance | 2.5 | | | pF |
| $C_{OUT}$ | Output Capacitance[2] | 2.0 | | | pF |
| $C_{IO}$ | I/O Bus Capacitance | 2.5 | | | pF |

1. Measurement conditions are $V_{IN}$ = 5.0 V, $T_A$ = 25 ˚C, and clock frequency = 1 MHz.
2. Output using single buffer structure (excluding package).

Table 7.5
DC Characteristics

| Symbol | Parameter | Condition[1] | Min | Typ | Max | Units |
|---|---|---|---|---|---|---|
| $V_{IL}$ | Voltage Input Low | | – | – | 0.8 | V |
| $V_{IH}$ | Voltage Input High | | 2.0 | – | – | V |
| $V_{OH}$ | Voltage Output High | $I_{OH}$ = -4 mA | 2.4 | 4.5 | – | V |
| | | $I_{OH}$ = -8 mA | 2.4 | 4.5 | – | V |
| | | $I_{OH}$ = -24 mA | 2.4 | 4.5 | – | V |
| $V_{OL}$ | Voltage Output Low | $I_{OL}$ = 4 mA | – | 0.2 | 0.4 | V |
| | | $I_{OL}$ = 8 mA | – | 0.2 | 0.4 | V |
| | | $I_{OL}$ = 24 mA | – | 0.2 | 0.4 | V |
| $I_{IL}$ | Current Input Leakage | $V_{DD}$ = Max, $V_{IN}$ = $V_{DD}$ or $V_{SS}$ | -10 | ±1 | 10 | µA |
| $I_{OZ}$ | Current 3-State Output Leakage | $V_{DD}$ = Max, $V_{OUT}$ = $V_{SS}$ or $V_{DD}$ | -10 | ±1 | 10 | µA |
| $I_{IPU}$ | Current Input Pull-up | $V_{IN}$ = $V_{SS}$ | -35 | -115 | -214 | µA |
| $I_{IPD}$ | Current 3-State Output w/Pull-up | $V_{IN}$ = $V_{SS}$ | -35 | -115 | -222 | µA |
| $I_{OSP4}$ | Current P-Channel Output Short Circuit (4 mA Output Buffers)[2] | $V_{DD}$ = Max, $V_{OUT}$ = $V_{SS}$ | -117 | -75 | -40 | mA |
| $I_{OSP8}$ | Current P-Channel Output Short Circuit (8 mA Output Buffers)[2] | $V_{DD}$ = Max, $V_{OUT}$ = $V_{SS}$ | -234 | -150 | -80 | mA |
| $I_{OSP24}$ | Current P-Channel Output Short Circuit (24 mA Output Buffers)[2] | $V_{DD}$ = Max, $V_{OUT}$ = $V_{SS}$ | -702 | -450 | -240 | mA |
| $I_{OSN4}$ | Current N-Channel Output Short Circuit (4 mA Output Buffers)[2] | $V_{DD}$ = Max, $V_{OUT}$ = $V_{DD}$ | 37 | 90 | 140 | mA |
| $I_{OSN8}$ | Current N-Channel Output Short Circuit (8 mA Output Buffers)[2] | $V_{DD}$ = Max, $V_{OUT}$ = $V_{DD}$ | 54 | 180 | 280 | mA |
| $I_{OSN24}$ | Current N-Channel Output Short Circuit (24 mA Output Buffers)[2] | $V_{DD}$ = Max, $V_{OUT}$ = $V_{DD}$ | 222 | 540 | 840 | mA |
| $I_{DD}$ | Quiescent Supply Current | $V_{IN}$ = $V_{DD}$ or $V_{SS}$ | – | 15 | – | µA |
| $I_{CC}$ | Dynamic Supply Current | VDD = Max, f = 33 MHz | – | 275 | 350 | mA |

1. Specified at $V_{DD}$ equals 5 V ±5% over the specified case temperature range. The maximum power dissipation of the L64381 is 1.84 W.

2. Not more than one output may be shorted at a time for a maximum duration of one second.

Table 7.6
Pin Description
Summary

| Mnemonic | Description | Type | Drive (mA) | Active |
|---|---|---|---|---|
| ABORT_IN | Abort In | Input | – | Low |
| ABORT_OUT | Abort Out | Output | 4 | Low |
| ADRS[5:0] | Register Select Address | Input | – | – |
| ADRS_STROBE | Address Strobe | Input | – | Low |
| AI_FCS_IN | Auto-Insert FCS | Input | – | Low |
| BUF_AVAIL | Buffer Available | Output | 4 | Low |
| BYTE_VALID[3:0] | Valid Bytes | Bidirectional | 6 | – |
| CHIP_SELECT | Chip Select | Input | – | Low |
| CLOCK | System Clock | Input | – | – |
| CLOCK_20MHZ | Internal Clock | Input | – | – |
| COLLIN[3:0] | Negative Collision Threshold | CMOS Input | – | – |
| COLLIP[3:0] | Positive Collision Threshold | CMOS Input | – | – |
| COLLOUTN | Collision Out | Output | 4 | Low |
| CREF[3:0] | Collision Reference | Input | – | – |
| DATA[31:0] | L64381 Data | Bidirectional | 6 | – |
| DATAI[3:0] | Serial Data In | CMOS Input | – | – |
| DATAN[3:0] | Negative Manchester-Encoded Data | Output | 24 | – |
| DATAP[3:0] | Positive Manchester-Encoded Data | Output | 24 | – |
| DATATN[3:0] | Data Threshold Negative | CMOS Input | – | – |
| DATATP[3:0] | Data Threshold Positive | CMOS Input | – | – |
| DREF[3:0] | Data Threshold Reference | Input | – | – |
| EOP | End of Packet | Bidirectional | 4 | Low |
| INTERRUPT | Interrupt | Output | 4 | Low |
| LBADN | Link Inverted | Output | 4 | Low |
| LEDSEL[1:0] | LED Select | Output | 4 | – |
| LP1 | Phase Detector | Output | – | – |
| LP2 | VCO | Input | – | – |
| LPASSN | Link Test Status | Output | 4 | Low |
| PDATA[15:0] | Processor Data Bus | Bidirectional | 4 | – |
| PKT_AVAIL | Packet Available | Output | 4 | Low |
| PLLAGND | PLL Analog Ground | Output | – | – |
| PLLVDD | PLL Power | Input | – | – |
| PLLVSS | PLL Digital Ground | Input | – | – |
| PORT_BUSY | Port Status | Output | 4 | Low |
| PORT_NO[1:0] | Port Number | Input, PD | – | – |

(Sheet 1 of 2)

Table 7.6 (Cont.)
Pin Description
Summary

| Mnemonic | Description | Type | Drive (mA) | Active |
|---|---|---|---|---|
| PREADY | Ready | Output | 4 | Low |
| PREEN[3:0] | Negative Pre-emphasis | Output | 24 | – |
| PREEP[3:0] | Positive Pre-emphasis | Output | 24 | – |
| P_TESTN | Global 3-State | CMOS Input, PU | – | Low |
| RCVNGN | Receiver Status | Output | 4 | Low |
| READ | Read | Input | – | Low |
| READ_OUT_PKT | Read Data from Receive FIFO | Input | – | Low |
| RESET | Reset | Input | – | Low |
| SOP | Start of Packet | Bidirectional | 4 | Low |
| STROBE | Strobe | Input | – | Low |
| TCK | JTAG Test Clock | Input | – | – |
| TDI | JTAG Test Data In | Input, PU | – | – |
| TDO | JTAG Test Data Out | Output | 4 | – |
| TMS | JTAG Test Mode Select | Input, PU | – | – |
| TP/AUI[3:0] | Twisted-Pair/AUI Select | Input | – | – |
| TRNSMTN | Transmitter Status | Output | 4 | Low |
| TRSTN | JTAG Test Reset | Input, PU | – | Low |
| WRITE_IN_PKT | Write Data to Transmit FIFO | Input | – | Low |

(Sheet 2 of 2)

## 7.3 Special VDD and VSS Connections

To minimize EMI and digital noise coupling, the L64381 provides separate power and ground for each port. The output buffers driving the DATAP[3:0], DATAN[3:0], PREEP[3:0], and PREEN[3:0] pins of each port have their own VDD and VSS pins, and are thus decoupled from the VDD and VSS pins for the remaining chip output buffers. Table 7.7 lists the VDD and VSS pins for each port.

Table 7.7
VDD/VSS Pin
Numbers Per Port

| Port Number | VDD Pin Number | VSS Pin Number |
|---|---|---|
| 0 | 139 | 140 |
| 1 | 120 | 119 |
| 2 | 107 | 108 |
| 3 | 101 | 102 |

## 7.4 Packaging

The L64381 is available in a 208-pin Plastic Quad Flat Pack (PQFP) package.

### 7.4.1 Ordering Information

Table 7.8 lists the L64381 according to order number and package type.

Table 7.8
L64381 Ordering Information

| Order Number | Clock Frequency (MHz) | Package Type | Operating Range |
|---|---|---|---|
| L64381 | 33 | 208-pin PQFP | Commercial |

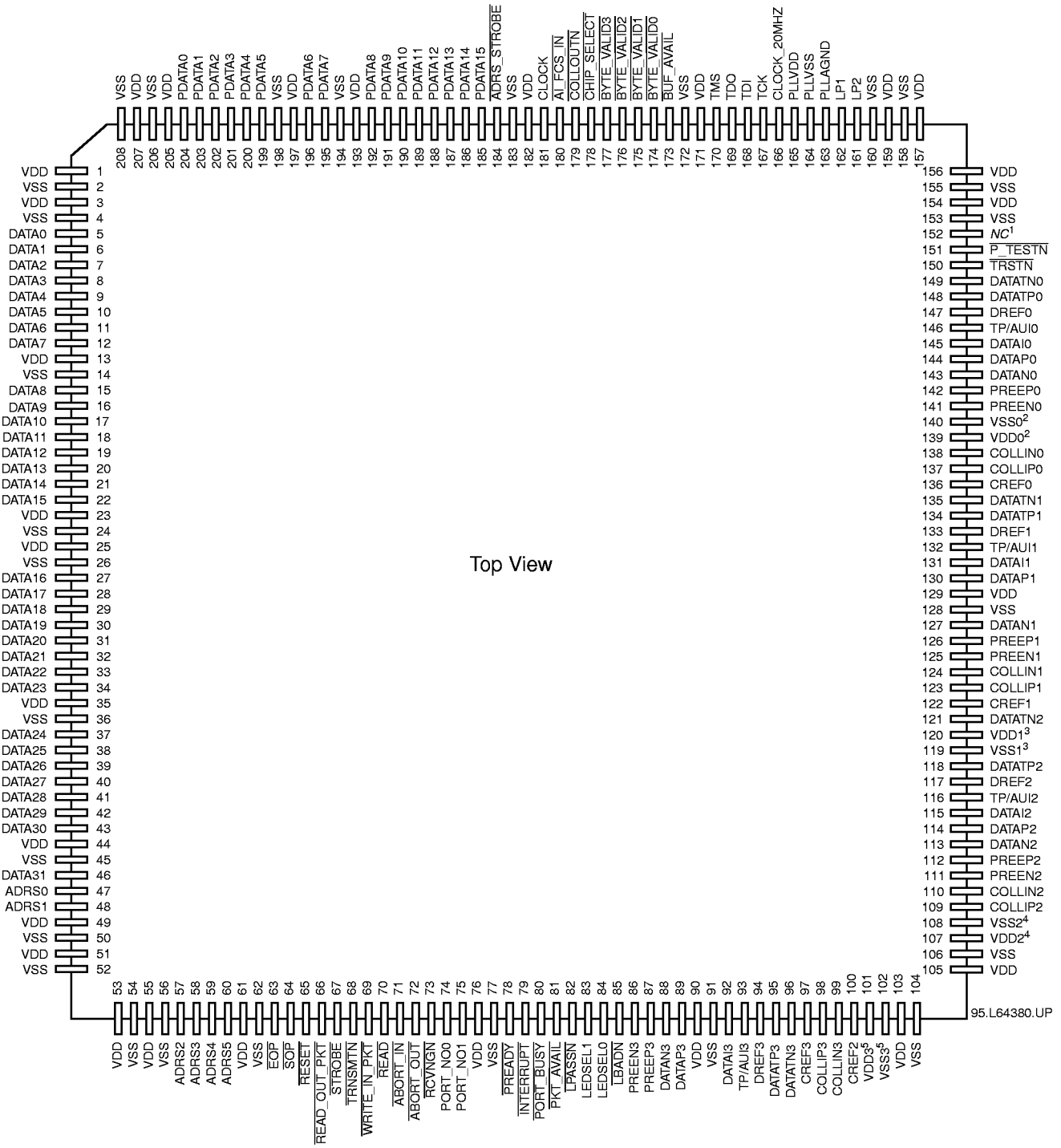
### 7.4.2 Package Information

This subsection provides three types of information for each package type: an alphabetical pin list (Table 7.9), a pinout (Figure 7.10), and a mechanical drawing (Figure 7.11).

Table 7.9
Alphabetical Pin List for
the 208-Pin PQFP

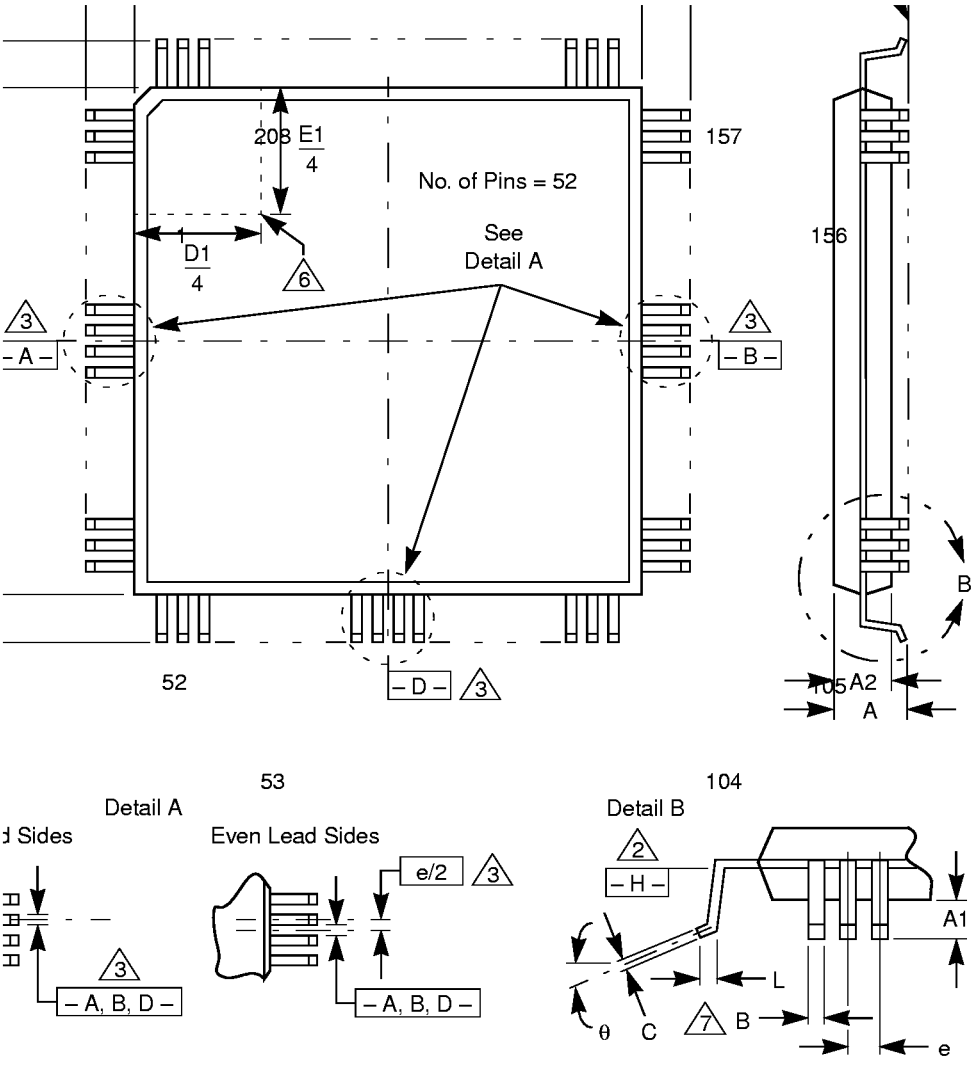
| Signal | Pin | Signal | Pin | Signal | Pin | Signal | Pin |
|---|---|---|---|---|---|---|---|
| ABORT_IN | 71 | DATA21 | 32 | PDATA9 | 191 | VDD | 61 |
| ABORT_OUT | 72 | DATA22 | 33 | PDATA10 | 190 | VDD | 76 |
| ADRS0 | 47 | DATA23 | 34 | PDATA11 | 189 | VDD | 90 |
| ADRS1 | 48 | DATA24 | 37 | PDATA12 | 188 | VDD3 | 101 |
| ADRS2 | 57 | DATA25 | 38 | PDATA13 | 187 | VDD | 103 |
| ADRS3 | 58 | DATA26 | 39 | PDATA14 | 186 | VDD | 105 |
| ADRS4 | 59 | DATA27 | 40 | PDATA15 | 185 | VDD2 | 107 |
| ADRS5 | 60 | DATA28 | 41 | PKT_AVAIL | 81 | VDD1 | 120 |
| ADRS_STROBE | 184 | DATA29 | 42 | PLLAGND | 163 | VDD | 129 |
| AI_FCS_IN | 180 | DATA30 | 43 | PLLVDD | 165 | VDD0 | 139 |
| BUF_AVAIL | 173 | DATA31 | 46 | PLLVSS | 164 | VDD | 154 |
| BYTE_VALID0 | 174 | DATAI0 | 145 | PORT_BUSY | 80 | VDD | 156 |
| BYTE_VALID1 | 175 | DATAI1 | 131 | PORT_NO0 | 74 | VDD | 157 |
| BYTE_VALID2 | 176 | DATAI2 | 115 | PORT_NO1 | 75 | VDD | 159 |
| BYTE_VALID3 | 177 | DATAI3 | 92 | PREADY | 78 | VDD | 171 |
| CHIP_SELECT | 178 | DATAN0 | 143 | PREEN0 | 141 | VDD | 182 |
| CLOCK | 181 | DATAN1 | 127 | PREEN1 | 125 | VDD | 193 |
| CLOCK_20MHZ | 166 | DATAN2 | 113 | PREEN2 | 111 | VDD | 197 |
| COLLIN0 | 138 | DATAN3 | 88 | PREEN3 | 86 | VDD | 205 |
| COLLIN1 | 124 | DATAP0 | 144 | PREEP0 | 142 | VDD | 207 |
| COLLIN2 | 110 | DATAP1 | 130 | PREEP1 | 126 | VSS | 2 |
| COLLIN3 | 99 | DATAP2 | 114 | PREEP2 | 112 | VSS | 4 |
| COLLIP0 | 137 | DATAP3 | 89 | PREEP3 | 87 | VSS | 14 |
| COLLIP1 | 123 | DATATN0 | 149 | P_TEST | 151 | VSS | 24 |
| COLLIP2 | 109 | DATATN1 | 135 | RCVNGN | 73 | VSS | 26 |
| COLLIP3 | 98 | DATATN2 | 121 | READ | 70 | VSS | 36 |
| COLLOUTN | 179 | DATATN3 | 96 | READ_OUT_PKT | 66 | VSS | 45 |
| CREF0 | 136 | DATATP0 | 148 | RESET | 65 | VSS | 50 |
| CREF1 | 122 | DATATP1 | 134 | SOP | 64 | VSS | 52 |
| CREF2 | 100 | DATATP2 | 118 | STROBE | 67 | VSS | 54 |
| CREF3 | 97 | DATATP3 | 95 | TCK | 167 | VSS | 56 |
| DATA0 | 5 | DREF0 | 147 | TDI | 168 | VSS | 62 |
| DATA1 | 6 | DREF1 | 133 | TDO | 169 | VSS | 77 |
| DATA2 | 7 | DREF2 | 117 | TMS | 170 | VSS | 91 |
| DATA3 | 8 | DREF3 | 94 | TP/AUI0 | 146 | VSS3 | 102 |
| DATA4 | 9 | EOP | 63 | TP/AUI1 | 132 | VSS | 104 |
| DATA5 | 10 | INTERRUPT | 79 | TP/AUI2 | 116 | VSS | 106 |
| DATA6 | 11 | LBADN | 85 | TP/AUI3 | 93 | VSS2 | 108 |
| DATA7 | 12 | LEDSEL0 | 84 | TRNSMTN | 68 | VSS1 | 119 |
| DATA8 | 15 | LEDSEL1 | 83 | TRSTN | 150 | VSS | 128 |
| DATA9 | 16 | LP1 | 162 | WRITE_IN_PKT | 69 | VSS0 | 140 |
| DATA10 | 17 | LP2 | 161 | VDD | 1 | VSS | 153 |
| DATA11 | 18 | LPASSN | 82 | VDD | 3 | VSS | 155 |
| DATA12 | 19 | PDATA0 | 204 | VDD | 13 | VSS | 158 |
| DATA13 | 20 | PDATA1 | 203 | VDD | 23 | VSS | 160 |
| DATA14 | 21 | PDATA2 | 202 | VDD | 25 | VSS | 172 |
| DATA15 | 22 | PDATA3 | 201 | VDD | 35 | VSS | 183 |
| DATA16 | 27 | PDATA4 | 200 | VDD | 44 | VSS | 194 |
| DATA17 | 28 | PDATA5 | 199 | VDD | 49 | VSS | 198 |
| DATA18 | 29 | PDATA6 | 196 | VDD | 51 | VSS | 206 |
| DATA19 | 30 | PDATA7 | 195 | VDD | 53 | VSS | 208 |
| DATA20 | 31 | PDATA8 | 192 | VDD | 55 | *NC* | 152 |

Figure 7.10
208-Pin PQFP Pinout




1. NC pins are not connected.
2. Can be decoupled as separate power and ground for Port 0.
3. Can be decoupled as separate power and ground for Port 1.
4. Can be decoupled as separate power and ground for Port 2.
5. Can be decoupled as separate power and ground for Port 3.

Figure 7.11
208-Pin PQFP
Mechanical Drawing




.08

| Dimension | | mm |
|---|---|---|
| A | Max | 4.10 |
| A1 | Min | 0.25 |
| A2 | Min | 3.20 |
| | Nom | 3.40 |
| | Max | 3.60 |
| B | Min | 0.17 |
| | Nom | 0.23 |
| | Max | 0.27 |
| C | Min | 0.13 |
| | Nom | 0.16 |
| | Max | 0.20 |
| D | Min | 30.40 |
| | Nom | 30.60 |
| | Max | 30.80 |
| D1 | Min | 27.90 |
| | Nom | 28.00 |
| | Max | 28.10 |
| e | BSC | 0.50 |
| E | Min | 30.40 |
| | Nom | 30.60 |
| | Max | 30.80 |
| E1 | Min | 27.90 |
| | Nom | 28.00 |
| | Max | 28.10 |
| L | Min | 0.45 |
| | Nom | 0.60 |
| | Max | 0.75 |
| θ | Min | 0° |
| | Max | 7° |

1. Total number of pins is 208. Controlling dimension is millimeter. Drawing is not to scale.
2. Datum plane – H – is located at mold parting line and is coincident with the bottom of the leads, where the lead exits the plastic body.
3. Datums - A – B and – D – to be determined at datum plane – H –.
4. To be determined at seating plane – C –.
5. Dimensions D1 and E1 do not include mold protrusion. Allowable protrusion is 0.25 mm per side. These dimensions do not include mold mismatch and are determined at – H –.
6. Details of Pin 1 identifier are optional but must be located within the zone indicated.
7. Dimension B does not include dambar protrusion. Allowable dambar protrusion shall be 0.08 mm total in excess of the B dimension at maximum material condition. Dambar cannot be located on the lower radius or the foot. Minimum spacing between adjacent leads to be 0.07 mm.
8. For board layout and manufacturing, you may obtain engineering drawings from your LSI Logic marketing representative by requesting the outline drawing for package code UP.

MD92.UPe