**PRODUCT NAME : CURIO (CURE FOR I/O SUBSYSTEM INTEGRATION)**

(Combined 16-bit Ethernet Media Access Controller with Physical Layer Signalling (MACE), 16-bit SCSI Controller (53C94-Mode 1), Enhanced Serial Communications Controller (85C30) with LocalTalk capabilities and extended FIFOs, 79C30A Serial Bus Port Interface, and IEEE P1149.1 Test Access Port).

**PRODUCT NUMBER :** 79C950

**DOCUMENT NUMBER : 90-018-7661-OBS-003**

**PRODUCT LINE :** NPD - 802 Networking

**DATE :** 6-6-91

## OBJECTIVE SPECIFICATION - ADVANCED COMBO (CURIO)

## TABLE OF CONTENTS


1 GENERAL DESCRIPTION....7
1.1 INTRODUCTION....7
2 DISTINCTIVE CHARACTERISTICS....8
2.1 MACE SECTION....8
2.2 SCSI SECTION....8
2.3 ESCC SECTION....8
2.4 SBP INTERFACE SECTION....9
3 FUNCTIONAL DESCRIPTION....10
3.1 CURIO BLOCK DIAGRAM....10
3.1.1 Basic Block Diagram....10
3.1.2 Detailed Overview....11
3.2 PINOUT SUMMARY....12
3.2.1 Numerical Pin Assignment Table, 181 pin PGA....15
3.2.2 Pin Assignment, 168 pin PQFP....18
3.3 PIN DESCRIPTION....19
3.3.1 Ethernet Subsection....19
3.3.1.1 Attachment Unit Interface....19
3.3.1.2 Digital Attachment Interface™....20
3.3.1.3 MACE System Interface....25
3.3.2 SCSI Subsection....27
3.3.2.1 SCSI BUS....27
3.3.2.2 SCSI System Interface....27
3.3.3 16-Bit System Interface....28
3.3.4 Internal ESCC Subsection....29
3.3.4.1 Internal ESCC Interface....29
3.3.4.2 Internal ESCC System Interface....30
3.3.5 SBP Subsection....31
3.3.5.1 SBP Interface....31
3.3.5.2 External SCC Interface....31
3.3.5.3 SBP System Interface....31
3.3.6 8-Bit Host System Interface....32
3.3.7 IEEE P1149.1 TAP Interface....33
3.3.8 General Interface....34
3.4 MACE Functional description....35
3.4.1 Basic MACE Functions....35
3.4.1.1 Network Interfaces....36
3.4.1.2 System Interface....36
3.4.2 Detailed MACE Functions....37
3.4.2.1 Bus Interface Unit (BIU)....37
3.4.2.1.1 BIU to FIFO Data Path....37
3.4.2.1.2 BIU to Control and Status Register Data Path....39
3.4.2.2 FIFO Sub-system....39
3.4.2.3 Media Access Control (MAC)....40
3.4.2.4 Manchester Encoder/Decoder (MENDEC)....44
3.4.2.4.1 Attachment Unit Interface....44
3.4.2.4.2 Digital Attachment Interface™....44
3.4.2.5 General Purpose Serial Interface (GPSI) Extension....45
3.4.2.6 Slave Access Operation....46
3.4.2.6.1 Read Access....46
3.4.2.6.2 Write Access....48
3.4.2.7 Transmit Operation....48
3.4.2.7.1 Transmit FIFO Write....48
3.4.2.7.2 Transmit Function Programming....49
3.4.2.7.3 Automatic Pad Generation....50
3.4.2.7.4 Transmit FCS Generation....51

3.4.2.7.5 Transmit Status Information ....52
3.4.2.7.6 Transmit Exception Conditions....52
3.4.2.8 Receive Operation ....54
3.4.2.8.1 Receive FIFO Read....55
3.4.2.8.2 Receive Function Programming....55
3.4.2.8.3 Automatic Pad Stripping ....56
3.4.2.8.4 Receive FCS Checking....56
3.4.2.8.5 Receive Status Information....56
3.4.2.8.6 Receive Exception Conditions ....56
3.4.2.9 Loopback Operation....58
3.4.3 MACE User Accessible Registers ....58
3.4.3.1 Receive FIFO(RCVFIFO) ....59
3.4.3.2 Transmit FIFO (XMTFIFO) ....59
3.4.3.3 Transmit Frame Control (XMTFC)....60
3.4.3.4 Transmit Frame Status (XMTFS)....61
3.4.3.5 Transmit Retry Count (XMTRC) ....62
3.4.3.6 Receive Frame Control (RCVFC)....62
3.4.3.7 Receive Frame Status (RCVFS) ....62
3.4.3.7.1 RFS0 - Receive Message Byte Count (RCVCNT)....63
3.4.3.7.2 RFS1 - Receive Status (RCVSTS)....63
3.4.3.7.3 RFS2 - Runt Packet Count (RNTPC) ....64
3.4.3.7.4 RFS3 - Receive Collision Count (RCVCC)....64
3.4.3.8 FIFO Frame Count (FIFOFC)....65
3.4.3.9 Interrupt Register (IR)....65
3.4.3.10 Interrupt Mask Register (IMR)....67
3.4.3.11 Poll Register (PR)....67
3.4.3.12 BIU Configuration Control (BIUCC)....68
3.4.3.13 FIFO Configuration Control (FIFOCC)....69
3.4.3.14 MAC Configuration Control (MACCC) ....71
3.4.3.15 PLS Configuration Control (PLSCC)....72
3.4.3.16 PHY Configuration Control (PHYCC)....73
3.4.3.17 MACE Chip Identification Register (CHIPID [15-00]) ....73
3.4.3.18 Internal Address Configuration (IAC)....74
3.4.3.19 Logical Address Filter (LADRF [63-00]) ....74
3.4.3.20 Physical Address (PADR [47-00])....76
3.4.3.21 Missed Packet Count (MPC)....76
3.4.3.22 User Test Register (UTR) ....76
3.4.3.23 Reserved Test Register 1 (RTR1)....78
3.4.3.24 Reserved Test Register 2 (RTR2)....79
3.4.3.25 Register Table Summary....80
3.4.3.26 Register Bit Summary....81
3.4.3.26.1 16-Bit Registers ....81
3.4.3.26.2 8-Bit Registers....81
3.4.3.26.3 Receive Frame Status ....82
3.5 SCSI FUNCTIONAL DESCRIPTION....83
3.5.1 SCSI Bus Sequences....83
3.5.2 Host Command Sequences....85
3.5.3 Parity Detection and Generation....86
3.5.4 FIFO Threshold ....86
3.5.5 Burst Mode DMA....87
3.5.6 SCSI Bus Throughput....87
3.5.7 Data Alignment....88
3.5.8 Register Descriptions....89
3.5.8.1 DMA Counter....90
3.5.8.2 DMA Counter Read Address....90
3.5.8.3 FIFO Register....90
3.5.8.4 Command Register....91
3.5.8.5 Status Register ....92
3.5.8.6 Destination ID ....93
3.5.8.7 Interrupt Register....94

3.5.8.8 SELECT/ReSELECT Time-Out....95
3.5.8.9 Sequence Step....95
3.5.8.10 Synchronous Period....96
3.5.8.11 FIFO Flags / Sequence Step....96
3.5.8.12 Synchronous Offset....97
3.5.8.13 Configuration-1 Register....97
3.5.8.14 Clock Conversion....99
3.5.8.15 Test Mode Register....99
3.5.8.16 Configuration-2 Register....100
3.5.8.17 Vendor ID/Rev / Configuration-3 Register....101
3.5.8.18 FIFO Bottom Register....103
3.5.9 Command Execution....103
3.5.9.1 Initiator mode interrupts....103
3.5.9.2 Target mode interrupts....104
3.5.10 Command Set....106
3.5.10.1 Initiator Commands....108
3.5.10.2 Target Commands....110
3.5.10.3 Disconnected State Commands....112
3.5.10.4 Miscellaneous Commands....114
3.6 ESCC FUNCTIONAL DESCRIPTION....115
3.6.1 ESCC Block Diagram....115
3.6.2 ESCC Data Path....116
3.6.3 Detailed ESCC Description....116
3.6.3.1 Data Communications Capabililties....116
3.6.3.2 Asynchronous Modes....117
3.6.3.3 Synchronous Modes....117
3.6.3.4 SDLC Loop Mode....119
3.6.3.5 Baud Rate Generator....120
3.6.3.6 Digital Phase-Locked Loop....122
3.6.3.7 Crystal Oscillator....122
3.6.3.8 Data Encodlng....123
3.6.3.9 Auto Echo and Local Loopback....123
3.6.3.10 I/O Interface Capabilities....124
3.6.3.11 Polling....124
3.6.3.12 Interrupts....124
3.6.3.13 CPU/DMA Block Transfer....125
3.6.4 Programming Information....126
3.6.4.1 Read Registers....127
3.6.4.2 Write Registers....129
3.6.5 ESCC Timing....135
3.6.5.1 Read Cycle Timing....136
3.6.5.2 Write Cycle Timing....136
3.6.5.3 Interrupt Acknowledge Cycle Timing....136
3.6.6 Status FIFO Enhancements....138
3.6.6.1 FIFO Detail....139
3.6.6.2 Enable/Dlsable....140
3.6.6.3 Read Operation....140
3.6.6.4 Write Operation....140
3.6.6.5 Byte Counter Detail....140
3.6.7 LocalTalk Enhancements....143
3.6.7.1 ENRA, ENRB Register Format....143
3.6.7.2 LocalTalk protocol format....144
3.6.7.3 ESCC enhanced features....144
3.6.8 Extended Transmit and Receive Data FIFOs....145
3.7 SBP INTERFACE FUNCTIONAL DESCRIPTION....146
3.7.1 SBP Interface Block Diagram....147
3.7.2 SBP Interface Data Path....148
3.7.3 SBP Interface Detail Functional Description....148
3.7.3.1 Address Table....149
3.7.3.2 Command/Status Register (CSR) Format....150

3.7.3.3 SBP Interface Routing Commands....151
3.7.4 External SCC interface....151
3.8 IEEE P1149.1 (JTAG) PORT FUNCTIONAL DESCRIPTION....152
3.8.1 Boundary Scan Circuit....152
3.8.2 TAP FSM....152
3.8.3 Supported Instructions....153
3.8.4 Instruction Register and Decoding Logic....153
3.8.5 Data Register Array....153
3.8.6 The TAP Reset Pin, TRST#....154
4 ELECTRICAL SPECIFICATIONS....155
4.1 DC CHARACTERISTICS....155
4.2 AC CHARACTERISTICS....159
4.2.1 MACE AC Timing....159
4.2.1.1 Clock and Reset Timing....179
4.2.1.2 BIU Timing....180
4.2.1.3 AUI Timing....180
4.2.1.4 DAI™ Timing....182
4.2.1.5 GPSI Timing....183
4.2.2 SCSI AC Timing....184
4.2.3 ESCC AC Timing....195
4.2.4 SBP AC Timing....204
4.2.5 IEEE/JTAG P1149.1 Port AC Timing....207

# OBJECTIVE SPECIFICATION - ADVANCED COMBO (CURIO)

## 1 GENERAL DESCRIPTION

### 1.1 INTRODUCTION

The CURIO is a combined Ethernet controller (MACE), Small Computer Systems Interface controller (SCSI), Enhanced Serial Communications Controller (ESCC) with LocalTalk capabilities and extended FIFOs, together with a Serial Bus Port interface to the Am79C30A ISDN Controller (SBP Interface). The chip includes most of the functionality in the Am79C9416, the Am53C94 (Mode 1), the Am85C30 and the SBP interface requirement for the Am79C30A. An IEEE 1149.1 compliant test access port is also provided.

The Ethernet Media Access Controller (MACE) section embodies the Media Access Control (MAC) and Physical Signaling (PLS) sub-layers of the 802.3 protocol. The device provides the IEEE defined Attachment Unit Interface (AUI) for coupling to remote Media Attachment Units (MAUs) or on-board transceivers. The device also provides a Digital Attachment Interface™ (DAI™), bypassing the differential AUI interface.

The SCSI controller section is a high performance device conforming to the ANSI standard, X3.131-1986, for Small Computer Systems Interface. It is a software compatible super-set of the 53C90 with additional commands, registers, etc., (53C94, mode 1). It also includes on-chip 48 mA drivers for single-ended transmission. The SCSI controller will operate at sustained data transfer rates of up to 5 MegaBytes per second in synchronous mode and 5 MegaBytes per second in asynchronous mode.

The Enhanced Serial Communications Controller (ESCC) section consists of a high-speed, multi-protocol communications peripheral. It has a total of two independent, full duplex channels and functions as a serial-to-parallel and parallel-to-serial converter/controller. AMD's proprietary enhancements make the ESCC easier to interface to, with higher effectiveness in high-speed applications, by reducing software overhead and eliminating significant external glue logic. A special hardware circuit has been included for improving system performance when executing the LocalTalk protocol. In addition, the transmit and receive data FIFO depth has been extended to 8 bytes each.

The Serial Bus Port Interface (SBP) provides a direct connection path to an external Am79C30A ISDN terminal controller. It allows the three independent 64Kbps 79C30A TDM data channels to be multiplexed and demultiplexed to and from two SCC channels and the 8-bit system bus. Pipelined byte packing/unpacking registers are also provided. The two SCC channels can be selected either from the two internal ESCC channels, or from two external SCC channels.

The IEEE P1149.1 test access port eases the continuity test between the CURIO and other components in the system by providing boundary-scan test capability.

The dual bus system interface provides a 16-bit data conduit to and from an 802.3 network and a SCSI bus, as well as an 8-bit data conduit to and from the two on chip ESCC channels and the external ISDN controller via the SBP interface.

# 2 DISTINCTIVE CHARACTERISTICS

## 2.1 MACE SECTION
Supports ISO 8802-3 (IEEE/ANSI 802.3) and Ethernet standards.
Implements both 802.3 MAC and PLS sub-layer functions.
Synchronous host system interface.
Fully independent system and network clocks.
High speed 16-bit data path to/from host system.
Slave based Register Address access to all on board configuration/status registers and transmit/receive FIFOs. Alternative Direct FIFO read/write access for interface to simple DMA controllers.
Arbitrary byte alignment for host memory interface.
Little endian or big endian memory interface support.
Allows for the provision of both an Attachment Unit Interface (AUI) and a Digital Attachment Interface™ (DAI™).
Individual 128 byte transmit and receive FIFOs.
Runt packets (less than 64 bytes) are automatically flushed from the receive FIFO during normal operation.
The transmit FIFO retains data and does not require refilling for collision retries within the slot time (512 bit times).
Automatic padding and stripping of illegally short message frames.
External address matching and rejection support for bridge/router functions.
Dynamic transmit FCS generation programmable on a packet-by-packet basis.
Low power (sleep) mode for power critical applications.
Two internal and one external loopback capabilities.

## 2.2 SCSI SECTION
ANSI X3.131-1986 Compatible
On-chip 48 mA drivers
Software compatible with the 53C90 (53C94-Mode 1)
SCSI-2 Tagged Queuing
High speed 16-bit data path to/from the host system
Burst Mode
Up to 5 MegaBytes/second Asynchronous SCSI
Up to 5 MegaBytes/second Synchronous SCSI
Single-ended SCSI mode only

## 2.3 ESCC SECTION
Fast Data rates of up to 4 Mb/s
Two Independent Full-duplex Serial Channels
Asynchronous Mode Features include:
- Programmable stop bits, clock divider, character length and parity
- Break detection/divider
- Error detection for framing, overrun and parity

Synchronous Mode Features include:
- Supports IBM BISYNC, SDLC, SDLC Loop, HDLC and ADCCP Protocols

Programmable CRC generators and checkers
SDLC/HDLC support includes frame control, zero insertion and deletion, abort, and residue handling
Enhanced ESCC functions support high-speed frame reception using DMA
14-bit byte counter
10 X 19 SDLC/HDLC Frame Status FIFO
Independent Control on all channels
Enhanced operation does not allow special receive conditions to lock the three-byte DATA FIFO when the 10 X 19 FIFO is enabled
Local Loopback and Auto Echo Modes
Internal or External Character Synchronization
1 Mb/s FM Encoding Transmit and Receive capability using internal DPLL at 16 Mhz
Internal Synchronization between RxC to PCLK and TxC to PCLK
This allows the user to eliminate external synchronization hardware required by the older NMOS devices when transmitting or receiving data at the maximum rate of 1/4 PCLK frequency
Dedicated LocalTalk protocol state machine improves system software efficiency
8-byte deep enhanced transmit and receive data FIFOs
Separate INTerrupts for each channel
Separate DTR/REQ#, REQ# pins for each channel

**2.4 SBP INTERFACE SECTION**

Direct Connections to an external Am79C30A ISDN Subscriber Controller and a 2-channel Serial Communication Controller (Am8530 / Am85C30)
Software programmable routing between external ISDN data with:
1. the 8-bit System Bus Port,
2. the 2-channel internal ESCC, and
3. the 2-channel external SCC.

Automatic framing of the three ISDN oriented data channels to and from the Am79C30A
Automatic Serial and Parallel Format Conversions
Built in pipeline registers for easy synchronization

# 3 FUNCTIONAL DESCRIPTION

## 3.1 CURIO BLOCK DIAGRAM

### 3.1.1 Basic Block Diagram

**Am79C950 CURIO**

**Basic Block Diagram**

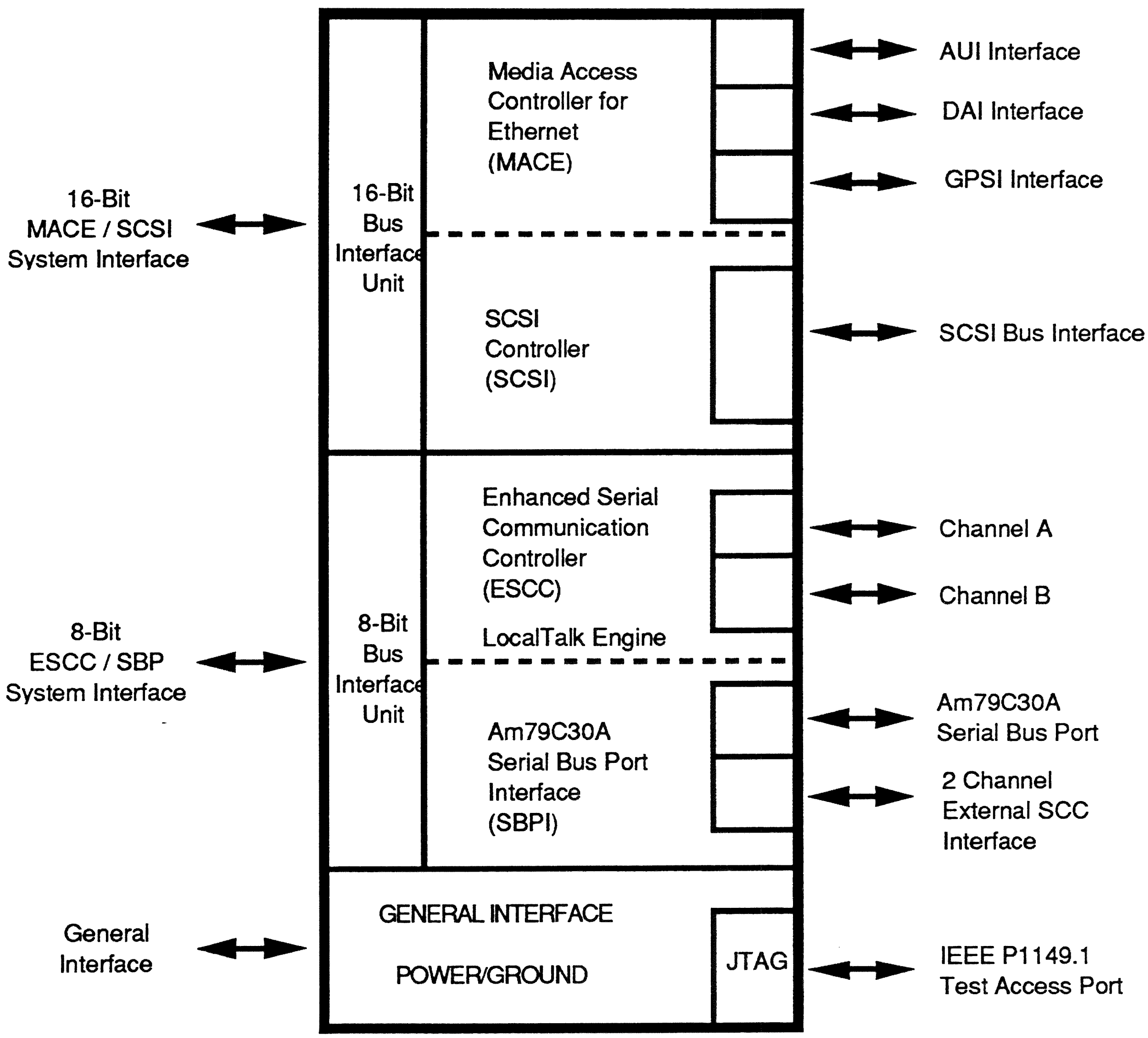
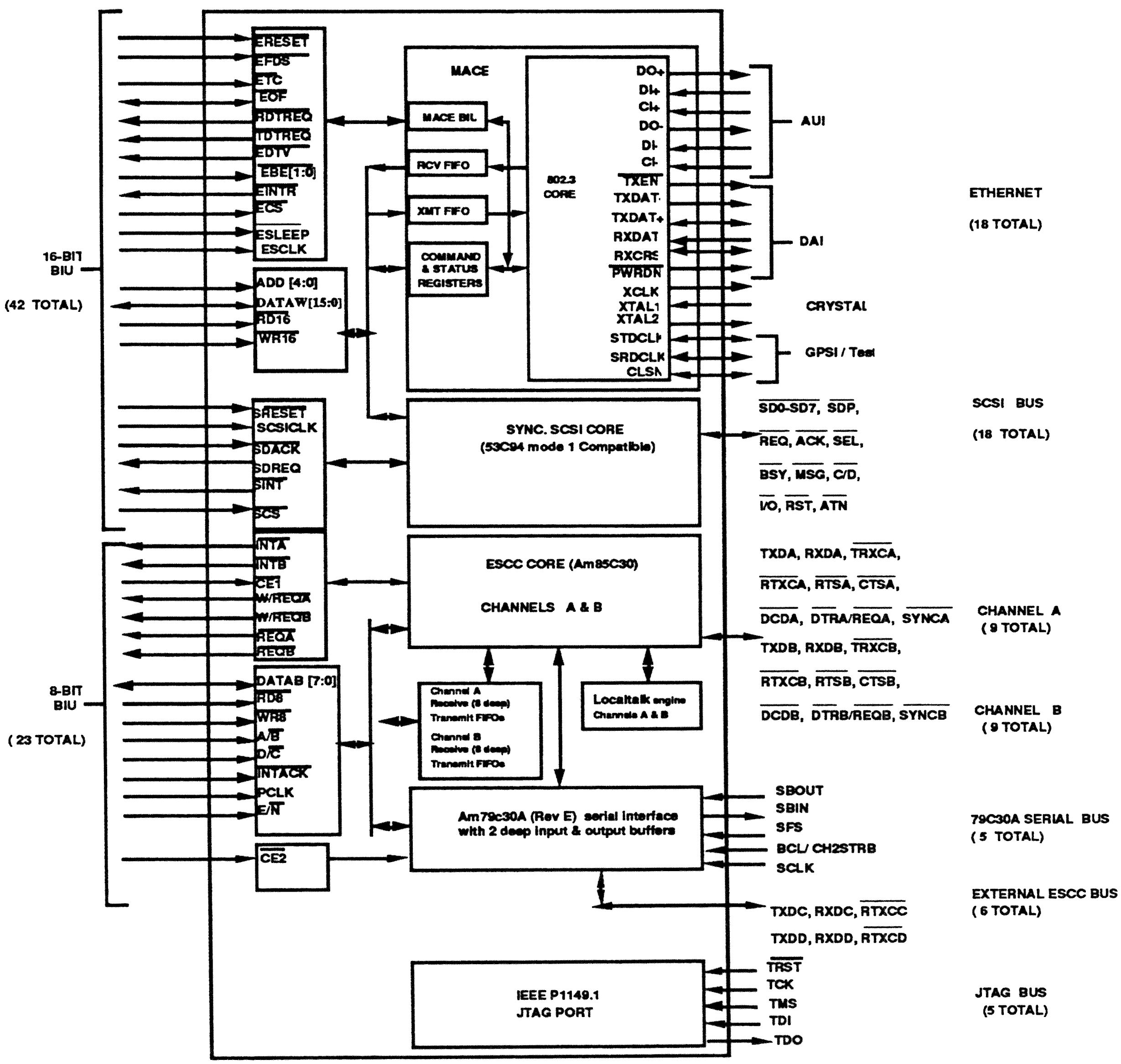



**Figure 1.**

### 3.1.2 Detailed Overview

**Am79C950 CURIO**
BASIC BLOCK OVERVIEW

AMD CONFIDENTIAL
DATE: 5/8/91
CURIO REV: 5.2
Author: Vinod Menon




**168-pin PQFP**

135 SIGNAL PINS

8 VCC, 2 AVCC
14 VSS, 2 AVSS

**Figure 2.**

## 3.2 PINOUT SUMMARY

### CURIO PIN SUMMARY

| PIN NAME | PIN FUNCTION | TYPE | POLARITY | PIN ASSIGNMENT |
|---|---|---|---|---|
| **MACE Attachment Unit Interface (AUI)** | | | | |
| DO+/DO- | Data Out | O | | |
| DI+/DI- | Data In | I | | |
| CI+/CI- | Control In | I | | |
| **MACE Digital Attachment Interface™ (DAI)** | | | | |
| TXEN# | Transmit Enable | O | Low | |
| TXDAT+ | Transmit Data + | O | High | |
| TXDAT- | Transmit Data - | O | Low | |
| RXDAT | Receive Data | I | High | |
| RXCRS | Receive Carrier Sense | I/O | High | |
| STDCLK | Serial Transmit Data Clock | I/O | | |
| SRDCLK | Serial Receive Data Clock | I/O | | |
| CLSN | Collision | I/O | High | |
| PWRDN# | Power Down | O | Low | |
| **MACE System Interface** | | | | |
| ERESET# | Ethernet Reset | I | Low | |
| EFDS# | Ethernet FIFO Data Select | I | Low | |
| ETC# | Timing Control | I | Low | |
| EOF# | End Of Frame | I/O | Low | |
| RDTREQ# | Receive Data Transfer Request | O | Low | |
| TDTREQ# | Transmit Data Transfer Request | O | Low | |
| EDTV# | Data Transfer Valid | O | Low | |
| EBE<1-0># | Upper,Lower byte enables | I | Low | |
| EINTR# | MACE Interrupt | O | Low | |
| ECS# | MACE Chip Select | I | Low | |
| ESLEEP# | MACE Sleep Mode Enable | I | Low | |
| ESCLK | ENET System Clock (MACE) | I | High | |
| **SCSI BUS** | | | | |
| SDI0<7-0>#, SDP# | SCSI I/O Data/Parity Bus | I/O | Low | |
| BSY# | SCSI BUSY | I/O | Low | |
| SEL# | SCSI SELECT | I/O | Low | |
| RST# | SCSI RESET | I/O | Low | |
| REQ# | SCSI REQUEST | I/O | Low | |
| ACK# | SCSI ACKNOWLEDGE | I/O | Low | |
| ATN# | SCSI ATTENTION | I/O | Low | |
| MSG# | SCSI MESSAGE | I/O | Low | |
| C/D# | COMMAND/DATA | I/O | Low | |
| I/O# | INPUT/OUTPUT | I/O | Low | |

**SCSI System Interface**

| | | | |
|---|---|---|---|
| SRESET# | SCSI Subsection Reset | I | Low |
| SCSICLK | SCSI System Clock | I | High |
| SDACK# | SCSI Data Transfer Acknowledge | I | Low |
| SDREQ | SCSI Data Transfer Request | O | High |
| SINT# | SCSI Interrupt | O | Low |
| SCS# | SCSI Chip Select | I | Low |

**16-Bit System Interface**

| | | | |
|---|---|---|---|
| ADD<4-0> | Address Bus | I | High |
| DATAW<15-0> | 16-bit Data Bus | I/O | High |
| RD16# | 16-bit Read Signal | I | Low |
| WR16# | 16-bit Write Signal | I | Low |

**Internal ESCC Interface**

| | | | |
|---|---|---|---|
| RTxCA# | Receive/Transmit Clock -Ch A | I | Low |
| RTxCB# | Receive/Transmit Clock - Ch B | I | Low |
| TRxCA# | Transmit/Receive Clock - Ch A | I/O | Low |
| TRxCB# | Transmit/Receive Clock - Ch B | I/O | Low |
| DCDA# | Data Carrier Detect- Ch A | I | Low |
| DCDB# | Data Carrier Detect- Ch B | I | Low |
| DTRA#/REQA# | Data Terminal Ready/ Req - Ch A | O | Low |
| DTRB#/REQB# | Data Terminal Ready/ Req - Ch B | O | Low |
| SYNCA# | Synchronization- Ch A | I/O | Low |
| SYNCB# | Synchronization- Ch B | I/O | Low |
| RxDA | Receive Data-Ch A | I | High |
| RxDB | Receive Data-Ch B | I | High |
| TxDA | Transmit Data-Ch A | O | High |
| TxDB | Transmit Data-Ch B | O | High |
| CTSA# | Clear to Send - Ch A | I | Low |
| CTSB# | Clear to Send - Ch B | I | Low |
| RTSA# | Request to Send - Ch A | O | Low |
| RTSB# | Request to Send - Ch B | O | Low |

**Internal ESCC System Interface**

| | | | |
|---|---|---|---|
| INTA# | ESCC Channel A Interrupt Request | O | Low |
| INTB# | ESCC Channel B Interrupt Request | O | Low |
| CE1# | ESCC Core Circuit Enable | I | Low |
| ~W/REQA# | Wait/Request- Ch A | O | Low |
| ~W/REQB# | Wait/Request- Ch B | O | Low |
| REQA# | Request - Ch A | O | Low |
| REQB# | Request - Ch B | O | Low |

**SBP Interface**

| | | | |
|---|---|---|---|
| SFS | Serial Frame Sync | I | High |
| CH2STRB | SBP Channel 2 Strobe | I | High |
| SCLK | Serial Data Clock | I | High |
| SBIN | Serial Data In to 79C30A | O | High |
| SBOUT | Serial Data Out from 79C30A | I | High |

**External ESCC Interface**

| | | | |
|---|---|---|---|
| TxDC | Transmit Data-Ch C | I | High |
| TxDD | Transmit Data-Ch D | I | High |
| RxDC | Receive Data-Ch C | O | High |
| RxDD | Receive Data-Ch D | O | High |
| RTxCC# | Receive/Transmit Clock-Ch C | O | Low |
| RTxCD# | Receive/Transmit Clock-Ch D | O | Low |

**SBP System Interface**

| | | | |
|---|---|---|---|
| CE2# | SBP Interface Circuit Enable | I | Low |

**8-Bit System Interface**

| | | | |
|---|---|---|---|
| DATAB<7-0> | 8-bit Data Bus | I/O | High |
| RD8# | 8-bit Read Signal | I | Low |
| WR8# | 8-bit Write Signal | I | Low |
| A/B# | Channel A/Channel B Select | I | Low |
| D/C# | Data/Control S elect | I | Low |
| INTACK# | Interrupt Acknowledge | I | Low |
| PCLK | Clock | I | |
| E/N# | ESCC Extended Address line | I | High |

**IEEE P1149.1 TAP Interface**

| | | | |
|---|---|---|---|
| TCK | Test Clock | I | High |
| TMS | Test Mode Select | I | High |
| TDI | Test Data In | I | High |
| TDO | Test Data Out | O | High |
| TRST# | Test Reset | I | Low |

**General Interface**

| | | | |
|---|---|---|---|
| XTAL1/XTAL2 | Crystal Input | I | |
| XCLK | Oscillator Output | O | High |

**Power, Ground Interface**

| | | | |
|---|---|---|---|
| DVDD | Digital Power (8 pins) | P | |
| DVSS | Digital Ground (14 pins) | P | |
| AVDD | Analog Power (2 pins) | P | |
| AVSS | Analog Ground (2 pins) | P | |

3.2.1 **Numerical Pin Assignment Table, 181 pin PGA. (6/29/91)**

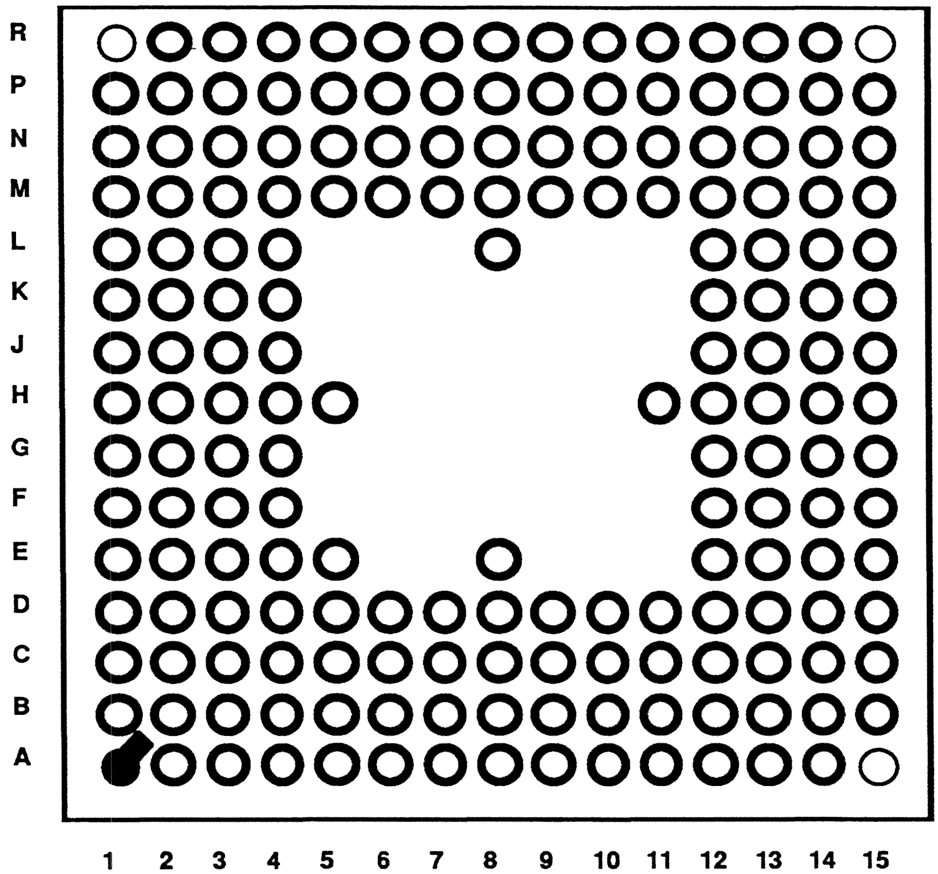
| PIN | SIGNALS | PIN | SIGNALS | PIN | SIGNALS | PIN | SIGNALS |
|---|---|---|---|---|---|---|---|
| B1 | | M5 | | P15 | | D11 | |
| C1 | | M6 | | N15 | | D10 | |
| D1 | W/REQA# | M7 | AVss | M15 | DATAW8 | D9 | |
| E1 | SYNCA# | R2 | DO- | L15 | DATAW9 | A14 | DVcc |
| F1 | RTxCA# | R3 | DO+ | K15 | DATAW10 | A13 | DVss |
| G1 | RxDA | R4 | AVcc | J15 | DATAW11 | A12 | REQA# |
| H1 | TRxCA# | R5 | DI- | H15 | DATAW12 | A11 | DTRA# |
| C2 | TxDA | R6 | DI+ | N14 | DATAW13 | A10 | RTSA# |
| D2 | SBOUT | R7 | CI- | M14 | DATAW14 | A9 | CTSA# |
| E2 | SBIN | P3 | CI+ | L14 | DATAW15 | B13 | DCDA# |
| F2 | SFS | P4 | AVcc | K14 | EOF# | B12 | PCLK |
| G2 | CH2STRB | P5 | DVcc | J14 | DVss | B11 | DCDB# |
| H2 | SCLK | P6 | EFDS# | H14 | DVcc | B10 | CTSB# |
| D3 | DVcc | P7 | EBE0# | M13 | SCSICLK | B9 | RTSB# |
| E3 | TxDC | N4 | EBE1# | L13 | SRESET# | C12 | REQB# |
| F3 | RxDC | N5 | DVcc | K13 | SDACK# | C11 | DTRB# |
| G3 | DVcc | N6 | ESCLK | J13 | SDREQ | C10 | DVss |
| H3 | RTxCC# | N7 | TDTREQ# | H13 | SINT# | C9 | TxDB |
| E4 | TxDD | M8 | RDTREQ# | L12 | SCS# | D8 | TRxCB# |
| F4 | RxDD | M9 | ADD0 | K12 | MSG# | D7 | RxDB |
| G4 | RTxCD# | M10 | ADD1 | J12 | C/D# | D6 | RTxCB# |
| H4 | DVss | M11 | ADD2 | H12 | DVss | D5 | SYNCB# |
| H5 | DVss | L8 | ADD3 | H11 | I/O# | E8 | DVss |
| J4 | TDO | M12 | ADD4 | G12 | ACK# | D4 | W/REQB# |
| K4 | TRST# | N8 | RD16# | F12 | REQ# | C8 | D/C# |
| L4 | TMS | N9 | WR16# | E12 | SDP# | C7 | CE1# |
| M4 | TCK | N10 | DVss | D12 | DVss | C6 | A/B# |
| J3 | TDI | N11 | ECS# | G13 | SD0# | C5 | E/N# |
| K3 | DVcc | N12 | ERESET# | F13 | SD1# | C4 | WR8# |
| L3 | PWRDN# | N13 | ESLEEP# | E13 | DVcc | C3 | RD8# |
| M3 | XCLK | P8 | EINTR# | D13 | SD2# | B8 | DVcc |
| N3 | TxDAT+ | P9 | EDTV# | C13 | SD3# | B7 | DATAB7 |
| J2 | TxDAT- | P10 | ETC# | G14 | DVss | B6 | DATAB6 |
| K2 | TxEN# | P11 | DATAW0 | F14 | SD4# | B5 | DATAB5 |
| L2 | RxDAT | P12 | DVss | E14 | SD5# | B4 | DATAB4 |
| M2 | RxCRS | P13 | DATAW1 | D14 | SD6# | B3 | DATAB3 |
| N2 | CLSN | P14 | DATAW2 | C14 | SD7# | B2 | DATAB2 |
| P2 | STDCLK | R8 | DATAW3 | B14 | DVss | A8 | DVss |
| J1 | SRDCLK | R9 | DVcc | G15 | RST# | A7 | DATAB1 |
| K1 | | R10 | DATAW4 | F15 | BSY# | A6 | DATAB0 |
| L1 | | R11 | DATAW5 | E15 | SEL# | A5 | INTB# |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| M1 | XTAL1 | R12 | DATAW6 | D15 | ATN# | A4 | INTA# |
| N1 | AVss | R13 | DATAW7 | C15 | DVss | A3 | INTACK# |
| P1 | XTAL2 | R14 | DVss | B15 | | A2 | CE2# |
| R1 | | R15 | | A15 | | A1 | |

* reference pin: E5.

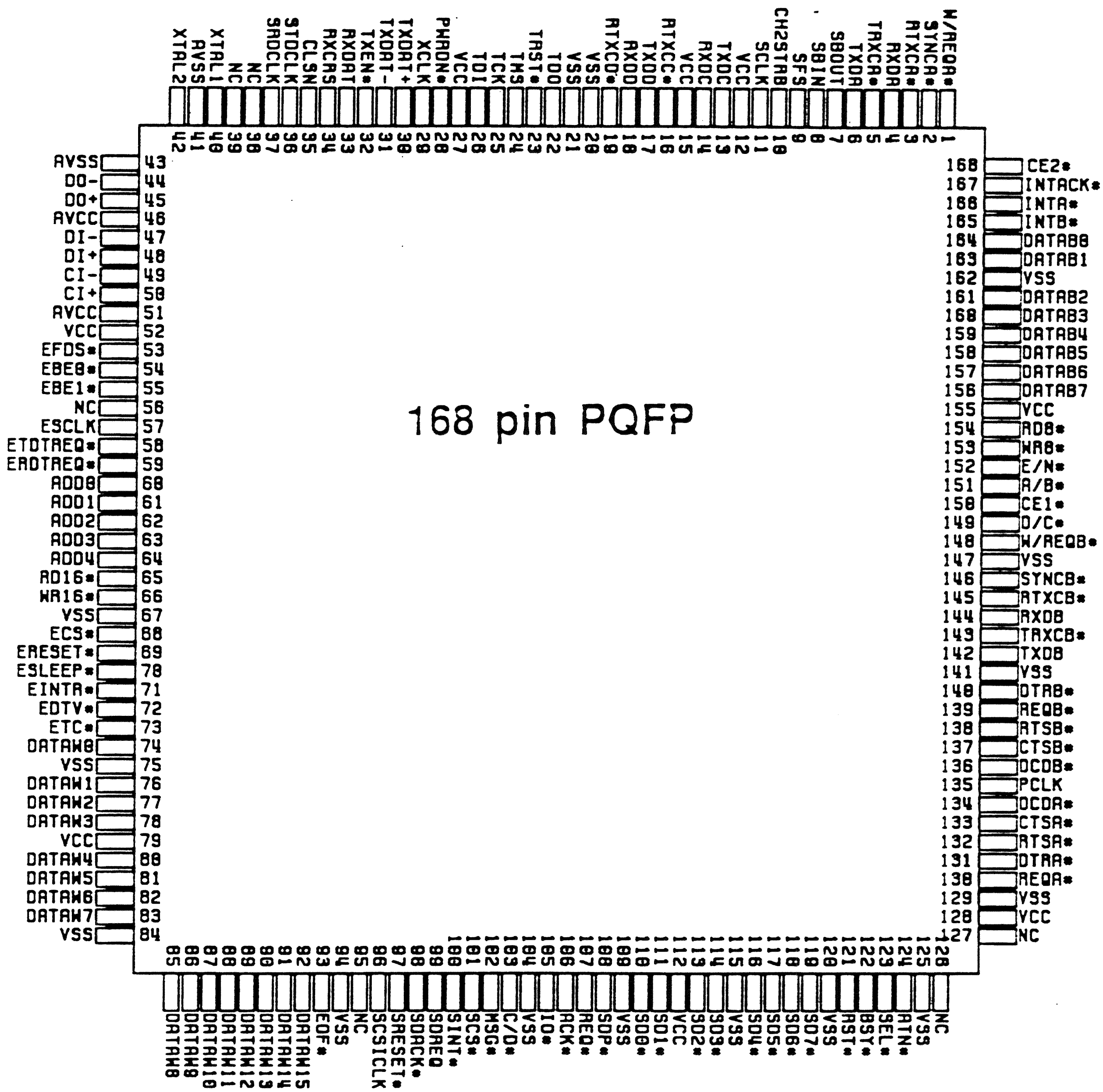
**Table 3.**

Bottom View




181-Pin Pin Grid Array

## 3.2.2 Pin Assignment, 168 pin PQFP. (6/29/91)



## 3.3 PIN DESCRIPTION

### 3.3.1 Ethernet Subsection

#### 3.3.1.1 Attachment Unit Interface (AUI)

DO+/DO- Data Out Output
A differential output pair from the MACE for transmitting Manchester encoded data to the network. Operates at pseudo ECL levels.

DI+/DI- Data In Input
A differential input pair to the MACE for receiving Manchester encoded data from the network. Operates at pseudo ECL levels.

CI+/CI- Control In Input
A differential input pair, indicating to the MACE that a collision has been detected on the network media, indicated by the CI± inputs being exercised with 10MHz pattern of sufficient amplitude and duration. Operates at pseudo ECL levels.

### 3.3.1.2 **Digital Attachment Interface™** (DAI™)

TXEN#/TXEN Transmit Enable Output

When the AUI port is selected (PORTSEL [1-0] = 00), an output indicating that valid Manchester encoded data is being output on the AUI DO± differential output. When the DAI™ port is selected (PORTSEL [1-0] = 10), indicates that Manchester data is being output on the DAI™ TXDAT± complementary outputs. Active low when the DAI™ is selected, active high when the AUI is selected. Operates at TTL levels.

| ESLEEP# | PORTSEL 1 0 | ENSTS | SIATST | INTERFACE DESCRIPTION | PIN FUNCTION |
|---|---|---|---|---|---|
| 0 | XX | X | X | Sleep Mode | High Impedance |
| 1 | 0 0 | 1 | 0 | AUI | TXEN Output |
| 1 | 0 1 | 1 | 0 | Reserved Mode | TXEN Output |
| 1 | 1 0 | 1 | 0 | DAI | TXEN# Output |
| 1 | 1 1 | 1 | 0 | GPSI | TXEN Output |
| 1 | XX | 0 | 0 | Status Disabled | High Impedance |
| 1 | XX | X | 1 | SIA Test Mode | TXEN Input |

TXDAT+/ TXDAT- Transmit Data Output

A complementary pair providing Manchester encoded data output from the MACE, for transmitting data to a local external network transceiver. During valid transmission (indicated by TXEN# low), a logical "1" is indicated by the TXDAT+ pin being in the high state, and TXDAT- in the low state; and logical "0" is indicated by the TXDAT+ pin being in the low state, and TXDAT- in the high state. During idle (TXEN# high), TXDAT+ will be in the high state, and TXDAT- in the low state. Operates at TTL levels. The operational of TXDAT+ is defined in the table below.

| ESLEEP# | PORTSEL 1 0 | ENSTS | SIATST | INTERFACE DESCRIPTION | PIN FUNCTION |
|---|---|---|---|---|---|
| 0 | XX | X | X | Sleep Mode | High Impedance |
| 1 | 0 0 | 1 | 0 | AUI | High Impedance |
| 1 | 0 1 | 1 | 0 | Reserved Mode | High Impedance |
| 1 | 1 0 | 1 | 0 | DAI | TXDAT+ Output |
| 1 | 1 1 | 1 | 0 | GPSI | TXDAT+ Output |
| 1 | XX | 0 | 0 | Status Disabled | High Impedance |
| 1 | XX | X | 1 | SIA Test Mode | TXDAT Input |

TXDAT+ Configuration

The operation of TXDAT- is defined in the table below.

| ESLEEP# | PORTSEL 1 0 | ENSTS | SIATST | INTERFACE DESCRIPTION | PIN FUNCTION |
|---|---|---|---|---|---|
| 0 | XX | X | X | Sleep Mode | High Impedance |
| 1 | 0 0 | 1 | 0 | AUI | High Impedance |
| 1 | 0 1 | 1 | 0 | Reserved Mode | High Impedance |
| 1 | 1 0 | 1 | 0 | DAI | TXDAT- Output |
| 1 | 1 1 | 1 | 0 | GPSI | TXDAT- Output |
| 1 | XX | 0 | 0 | Status Disabled | High Impedance |
| 1 | XX | X | 1 | SIA Test Mode | High Impedance |

TXDAT- Configuration

Note : PORTSEL and ENSTS are located in the PLS Configuration Control register. SIATST is located in Reserved Test Register 1.

RXDAT Receive Data Input

When the DAI™ port is selected (PORTSEL [1-0] = 10), the Manchester encoded data input to the integrated clock recovery and Manchester decoder of the MACE, from an external network transceiver. When the GPSI port is selected (PORTSEL [1-0] =11), the NRZ decoded data input to the MAC core of the MACE, from an external Manchester encoder/decoder. Operates at TTL levels.

| ESLEEP# | PORTSEL 1 0 | ENSTS | SIATST | INTERFACE DESCRIPTION | PIN FUNCTION |
|---|---|---|---|---|---|
| 0 | XX | X | X | Sleep Mode | High Impedance |
| 1 | 0 0 | 1 | 0 | AUI | High Impedance |
| 1 | 0 1 | 1 | 0 | Reserved Mode | High Impedance |
| 1 | 1 0 | 1 | 0 | DAI | RXDAT Input |
| 1 | 1 1 | 1 | 0 | GPSI | RXDAT Input |
| 1 | XX | 0 | 0 | Status Disabled | High Impedance |
| 1 | XX | X | 1 | SIA Test Mode | RXDAT Output |

RXDAT Configuration

RXCRS Receive Carrier Sense Input/Output

When the AUI port is selected (PORTSEL [1-0] = 00), an output indicating that the DI± input pair is receiving valid Manchester encoded data from the external transceiver which meets the signal amplitude and pulse width requirements. RXCRS will be asserted high for the entire duration of the receive message. When the DAI™ port is selected (PORTSEL [1-0] = 10), an input signaling the MACE that a receive carrier condition has been detected on the network, and valid Manchester encoded data is being presented to the MACE on the RXDAT line. When the GPSI port is selected (PORTSEL [1-0] = 11), an input signalling the internal MAC core that valid NRZ data is being presented on the RXDAT input. Operates at TTL levels.

| ESLEEP# | PORTSEL 1 0 | ENSTS | SIATST | INTERFACE DESCRIPTION | PIN FUNCTION |
|---|---|---|---|---|---|
| 0 | XX | X | X | Sleep Mode | High Impedance |
| 1 | 0 0 | 1 | 0 | AUI | RXCRS Output |
| 1 | 0 1 | 1 | 0 | Reserved Mode | RXCRS Output |
| 1 | 1 0 | 1 | 0 | DAI | RXCRS Input |
| 1 | 1 1 | 1 | 0 | GPSI | RXCRS Input |
| 1 | XX | 0 | 0 | Status Disabled | High Impedance |
| 1 | XX | X | 1 | SIA Test Mode | RXCRS Output |

RXCRS Configuration

Note : PORTSEL and ENSTS are located in the PLS Configuration Control register. SIATST is located in Reserved Test Register 1.

STDCLK Serial Transmit Data Clock Input/Output

When using the AUI or DAI™ port, STDCLK is an output operating at one half the crystal or XCLK frequency. STDCLK is the encoding clock for Manchester data transferred to the output of either the AUI DO± pair or the DAI™ TXDAT± pair. When using the GPSI port, STDCLK is an input at the network data rate, provided by the external Manchester encode/decoder, to strobe out the NRZ data presented on the TXDAT+ output.

| SLEEP# | PORTSEL 1 0 | ENSTS | SIATST | INTERFACE DESCRIPTION | PIN FUNCTION |
|---|---|---|---|---|---|
| 0 | XX | X | X | Sleep Mode | High Impedance |
| 1 | 0 0 | 1 | 0 | AUI | STDCLK Output |
| 1 | 0 1 | 1 | 0 | Reserved Mode | STDCLK Output |
| 1 | 1 0 | 1 | 0 | DAI | STDCLK Output |
| 1 | 1 1 | 1 | 0 | GPSI | STDCLK Input |
| 1 | XX | 0 | 0 | Status Disabled | High Impedance |
| 1 | XX | X | 1 | SIA Test Mode | STDCLK Output |

STDCLK Configuration

Note : PORTSEL and ENSTS are located in the PLS Configuration Control register. SIATST is located in Reserved Test Register 1.

SRDCLK Serial Receive Data Clock Input/Output

The Serial Receive Data output is synchronous to this clock, which runs at the 10MHz receive data clock frequency. The pin is only configured as an input when the SIATST is asserted.

| SLEEP# | PORTSEL 1 0 | ENSTS | SIATST | INTERFACE DESCRIPTION | PIN FUNCTION |
|---|---|---|---|---|---|
| 0 | XX | X | X | Sleep Mode | High Impedance |
| 1 | 0 0 | 1 | 0 | AUI | SRDCLK Output |
| 1 | 0 1 | 1 | 0 | Reserved Mode | SRDCLK Output |
| 1 | 1 0 | 1 | 0 | DAI | SRDCLK Output |
| 1 | 1 1 | 1 | 0 | GPSI | SRDCLK Input |
| 1 | XX | 0 | 0 | Status Disabled | High Impedance |
| 1 | XX | X | 1 | SIA Test Mode | SRDCLK Output |

SRDCLK Configuration

Note : PORTSEL and ENSTS are located in the PLS Configuration Control register. SIATST is located in Reserved Test Register 1.

CLSN Collision Input/Output

An external indication that a collision condition has been detected by the Medium Attachment Unit (MAU), and that signals from two or more nodes are present on the network. When the AUI port is selected (PORTSEL [1-0] = 00), CLSN will be activated when the CI± input pair is receiving a collision indication from the external transceiver which meets the signal amplitude and pulse width requirements. CLSN will be asserted high for the entire duration of the collision detection, but will not be asserted during the SQE Test message following a transmit message on the AUI. When the DAI™ port is selected (PORTSEL [1-0] = 10), CLSN will be asserted high when simultaneous transmit and receive activity is detected (logically detected when RXCRS and TXEN# are both active). When the GPSI port is selected (PORTSEL [1-0] = 11), an input from the external Manchester encoder/decoder signaling the MACE that a collision condition has been detected on the network, and any receive frame in progress should be aborted. Operates at TTL levels.

| SLEEP# | PORTSEL 1 0 | ENSTS | SIATST | INTERFACE DESCRIPTION | PIN FUNCTION |
|---|---|---|---|---|---|
| 0 | XX | X | X | Sleep Mode | High Impedance |
| 1 | 0 0 | 1 | 0 | AUI | CLSN Output |
| 1 | 0 1 | 1 | 0 | Reserved Mode | CLSN Output |
| 1 | 1 0 | 1 | 0 | DAI | CLSN Output |
| 1 | 1 1 | 1 | 0 | GPSI | CLSN Input |
| 1 | XX | 0 | 0 | Status Disabled | High Impedance |
| 1 | XX | X | 1 | SIA Test Mode | CLSN Output |

CLSN Configuration

Note : PORTSEL and ENSTS are located in the PLS Configuration Control register. SIATST is located in Reserved Test Register 1.

PWRDN# Power Down Output

An output from the MACE to indicate the network port in use, as programmed by the PORTSEL[1-0] bits. Active (low) when the AUI port is selected. Inactive (high) when the DAI™ is selected.

| ESLEEP# | PORTSEL 1 0 | ENSTS | SIATST | INTERFACE DESCRIPTION | PIN FUNCTION |
|---|---|---|---|---|---|
| 0 | XX | X | X | Sleep Mode | High Impedance |
| 1 | 0 0 | X | 0 | AUI | Low |
| 1 | 0 1 | X | 0 | Reserved Mode | High |
| 1 | 1 0 | X | 0 | DAI | High |
| 1 | 1 1 | X | 0 | Reserved | |
| 1 | XX | X | 1 | SIA Test Mode | High Impedance |

PWRDN# Configuration

### 3.3.1.3 MACE System Interface

ERESET# Ethernet Reset Input

This signal clears the MACE subsection logic. ERESET# can be asynchronous to ESCLK, but must be asserted for a minimum of 15 ESCLK cycles.

EFDS# - Ethernet FIFO Data Select Input

Ethernet FIFO Data Select allows direct access to the transmit or receive FIFO without use of the address bus. EFDS# must be activated in conjunction with RD16#. When the MACE samples RD16# as low and EFDS# low, a read cycle from the receive FIFO will be initiated. When the MACE samples WR16# and EFDS# low, a write cycle to the transmit FIFO will be initiated. The ECS# line should be inactive (high) when FIFO access is requested using the EFDS# pin. If the MACE samples both ECS# and EFDS# as active simultaneously, no cycle will be executed, and EDTV# will remain inactive.

ETC# Timing Control Input

The Timing Control input conditions the minimum number of System Clocks (ESCLK) cycles taken to read or write the internal registers and FIFOs. ETC# can be used as a wait state generator, to allow additional time for data to be presented by the host during a write cycle, or allow additional time for the data to be latched in a read cycle. ETC# has an internal (ESLEEP# disabled) pull up.

| ETC# | Number of Clocks |
|---|---|
| 1 | 2 |
| 0 | 3 |

EOF# End Of Frame Input/Output/3-state

End Of Frame will be asserted by the MACE as the last byte/word of information is read from the receive FIFO. This will indicate the completion of the frame status field for the receive message. End Of Frame must be asserted low to the MACE, as the last byte/word of the frame is written into the transmit FIFO.

RDTREQ# Receive Data Transfer Request Output

Receive Data Transfer Request indicates there is data in the receive FIFO to be read. When RDTREQ# is asserted there will be a minimum of 16 bytes to be read except at the end of the frame, in which case EOF# will be asserted. RDTREQ# can be programmed to request receive data transfer when 16, 32 or 64 bytes are available in the receive FIFO, by programming the Receive FIFO Watermark (RCVFW bits) in the FIFO Configuration Control register. Note that unless Runt Packet Accept is enabled (RPA bit) in the User Test Register, at least 64 bytes of packet information must be received prior to the initial assertion of RDTREQ# for the received packet. RDTREQ# will be asserted only when Enable Receive (ENRCV) is set in the MAC Configuration Control register.

TDTREQ# Transmit Data Transfer Request Output

Transmit Data Transfer Request indicates there is room in the transmit FIFO for more data. TDTREQ# is asserted when there are a minimum of 16 bytes empty in the transmit FIFO. TDTREQ# can be programmed to request transmit data transfer when 16, 32 or 64 bytes are available in the transmit FIFO, by programming the Transmit FIFO Watermark (XMTFW bits) in the FIFO Configuration Control register. TDTREQ# will be asserted only when Enable Transmit (ENXMT) is set in the MAC Configuration Control register.

EDTV# Data Transfer Valid Output

When asserted, indicates that the read or write operation has completed successfully. The absence of EDTV# at the termination of a host access cycle on the MACE indicates that the data transfer was unsuccessful. The latching or strobing of read or write data can be synchronized to the ESCLK input rather then using this signal.

$EBE_{0-1}$# Byte Enable Input

Used to indicate the active portion of the data transfer to or from the internal FIFOs. For word (16-bit) transfers, both $EBE_0$ and $EBE_1$ should be activated by the external host/controller. Single byte transfers are performed by identifying the active data bus byte and activating only one of the two signals. The function of the $EBE_{0-1}$ pins is programmed using the BSWP bit (BIU Configuration Control register, bit 6). $EBE_{0-1}$# are not required for accesses to internal MACE registers.

EINTR# Interrupt Output / 3-state

An attention signal that indicates that one or more of the following status flags are set: XMTINT, RCVINT, MPCO, CERR or BABL. Each of the interrupts can be individually masked. No interrupt conditions can take place in the MACE after a hardware ERESET#.

ECS# MACE Chip Select Input

Used to access the MACE FIFOs and internal registers locations using the address bus. The FIFOs may alternatively be directly accessed without supplying the FIFO address, by using the EFDS# and RD16#, WR16# pins.

ESLEEP# Sleep Mode Input

The Sleep Mode input allows the MACE to be placed in a power saving mode. All outputs will be placed in an inactive or high impedance state. Clock inputs to the MACE can be suspended. On removal of ESLEEP#, the MACE will go through an internally generated hardware ERESET# sequence. The MACE internal registers must be re-initialized on removal of the Sleep Mode.

ESCLK MACE Clock Output
The system clock input controls the operational frequency of the slave interface to the MACE and the internal processing of frames. ESCLK is unrelated to the 20MHz clock frequency required for the 802.3/Ethernet interface. The frequency range is currently 5 MHz - 25 MHz.

### 3.3.2 **SCSI Subsection**

#### 3.3.2.1 **SCSI BUS**

SDIO<7-0>#,SDP# SCSI I/O Data/Parity Bus Input/Output
48 mA, open drain SCSI single-ended Data/Parity Input/Output bus. These pins are active low SCSI Data signals.

BSY# SCSI BUSY Input/Output
48mA open drain SCSI I/Os.

SEL# SCSI SELECT Input/Output
48mA open drain SCSI I/O.

RST# SCSI RESET Input/Output
48mA open drain SCSI I/O. This command will cause the SCSI Controller to driver RST# true for 25-40uS, depending on SCSICLK frequency and SCSICLK Conversion Factor. See Miscellaneous Commands.

REQ# SCSI REQUEST Input/Output
48mA open drain SCSI I/O. Asserted only in Target mode.

ACK# SCSI ACKNOWLEDGE Input/Output
48mA open drain SCSI I/O. Driven in Initiator mode only.

ATN# SCSI ATTENTION Input/Output
48mA output Schmitt trigger input. In Initiator mode, is asserted when the controller detects an incoming parity error, or may be asserted by certain SCSI CORE commands. In Target mode, this signal is an input. Hysterisis is nominally 400mV centered at 1.4 Volt.

MSG# SCSI MESSAGE Input/Output
Bi-directional SCSI phase signal. $I_{OL}$ is 48mA output in Target mode, and Schmitt trigger input in Initiator mode. The I/O hysterisis is nominally 400mV centered at 1.4 Volt.

C/D# COMMAND/DATA Input/Output
Bi-directional SCSI phase signal. $I_{OL}$ is 48mA output in Target mode, and Schmitt trigger input in Initiator mode. The I/O hysterisis is nominally 400mV centered at 1.4 Volt.

I/O# INPUT/OUTPUT Input/Output
Bi-directional SCSI phase signal. $I_{OL}$ is 48mA output in Target mode, and Schmitt trigger input in Initiator mode. The I/O hysterisis is nominally 400mV centered at 1.4 Volt.

#### 3.3.2.2 **SCSI System Interface**

SRESET# SCSI Subsection Reset Input
This signal clears the internal SCSI controller logic. It can be asynchronous to SCSICLK.

SCSICLK SCSI System Clock Input
Square wave clock input which generates internal chip timing. The maximum frequency is 25MHz. The minimum frequency for asynchronous SCSI is 10MHz. The minimum frequency for synchronous transmission rate is equal to the SCSICLK period divided by the value in the Synchronous Transfer Period Register. The asynchronous transmission rate is indirectly affected by the SCSICLK period. See section 3.5.6 on SCSI Bus Throughput.

SDACK# SCSI Data Transfer Acknowledge Input
Active low SCSI DMA Acknowledge from the DMA controller. SDACK# accesses the FIFO only, while SCS# accesses any register including the FIFO. SCS# and SDACK# must never be true simultaneously. Furthermore, since MACE and SCSI sections cannot be enabled at the same time, it follows that only one out of the 4 signals, EFDS#, ECS#, SDACK# and SCS# can be active at any moment.

SDREQ SCSI Data Transfer Request Output
Active high SCSI DMA Request signal to the DMA controller. SDREQ will remain true as long as either the FIFO contains at least one word to send to memory during DMA read, or has room for one more word in the FIFO during DMA write.

SINT# SCSI Interrupt Output
Open drain SCSI Interrupt signal to the microprocessor. It is latched on the rising edge of SCSICLK. It may be cleared by reading the Interrupt Register or SRESET# or a SCSI software Reset (but not by a SCSI Reset). This output cannot be masked by the user.

SCS# SCSI Chip Select Input
This input enables 8-bit access to SCSI core registers during read or write. SCS# uses the address inputs to access any register and including the FIFO, while SDACK# accesses only the FIFO. SCS# and SDACK# must never be active simultaneously. Furthermore, since MACE and SCSI sections cannot be enabled at the same time, it follows that only one out of the 4 signals, EFDS#, ECS#, SDACK# and SCS# can be active at any moment.

### 3.3.3 16-Bit System Interface

| | | |
|---|---|---|
| ADD<4-0> | Address Bus | Input |
| Ethernet: | ADD is used to access the internal registers and FIFOs for read or write operations. | |
| SCSI: | These pins are address bits used to access the internal registers. Although shared with the Ethernet subsection, access timing involved is independent. | |
| DATAW<15-0> | Data Bus | Input /Output/3-state |
| Ethernet: | DATAW Bus contains read and write data to and from internal registers and the transmit and receive FIFOs. | |
| SCSI: | Bidirectional active high bus with internal pull-ups. The processor accesses internal registers on the lower 8 bits, while the DMA accesses the FIFO using all 16 bits. Although shared with the Ethernet subsection, access timing involved is independent. | |
| RD16# | Read Control Signal | Input |
| Ethernet: | Indicates a READ operation during a MACE register or FIFO access. | |
| SCSI: | Active low register Read signal. This input allows internal registers to drive the data bus when either SCS# or SDACK# are true. Although shared with the Ethernet subsection, access timing involved is independent. However, RD16# and WR16# should not be asserted simultaneously. | |
| WR16# | Write Control Signal | Input |
| Ethernet: | Indicates a WRITE operation during a MACE register or FIFO access. | |
| SCSI: | Active low register Write signal. This input causes data to be written into the SCSI controller's internal registers when SCS# is true. When SDACK# is true, data is written into the internal FIFO. Although shared with the Ethernet subsection, access timing involved is independent. However, RD16# and WR16# should not be asserted simultaneously. | |

### 3.3.4 Internal ESCC Subsection

#### 3.3.4.1 Internal ESCC Interface

RTxCA#, RTxCB# Receive/Transmit Clocks Inputs

These pins can be programmed in several different modes of operation. In each channel, RTxC# may supply the receive clock, the transmit clock, the clock for the baud rate generator, or the clock for the Digital Phase-Locked Loop. These pins can also be programmed for use with the respective SYNC# pins as a crystal oscillator. The receive clock may be 1, 16, 32 or 64 times the data rate in asynchronous modes.

TRxCA#, TRxCB# Transmit/Receive Clocks Input/Output

These pins can be programmed in several different modes of operation. TRxC# may supply the receive clock or the transmit clock in the input mode or supply the output of the Digital Phase-Locked Loop, the crystal oscillator, the baud rate generator, or the transmit clock in the output mode.

DCDA#, DCDB# Data Carrier Detect Inputs

These pins function as receiver enables if they are programmed for Auto Enables; otherwise, they may be used as general purpose input pins. Both pins are Schmitt-triggered buffered to accommodate slow rise-time signals. The SCC detects pulses on these pins and can interrupt the CPU on both logic level transitions.

DTRA#/REQA#, DTRB#/REQB# Data Terminal Ready Outputs
These outputs follow the inverted state programmed into the DTR bit in WR5. They can also be used as general purpose outputs or as Request Lines for a DMA controller.

SYNCA#, SYNCB# Synchronization Input/Output
These pins can act either as inputs, outputs, or part of the crystal oscillator circuit. In the Asynchronous Receive mode (crystal oscillator option not selected), these pins are inputs similar to CTS# and DCD#. In this mode, transitions on these lines affect the state of the Synchronous/Hunt status bits in Read Register 0 but have no other function.

In External Synchronization mode with the crystal oscillator not selected, these lines also act as inputs. In this mode, SYNC# must be driven LOW two receive clock cycles after the last bit in the synchronbous character is recieved. Character assembly begins on the rising edge of the receive clock immediately preceding the activation of SYNC#.

In the Internal Synchronization mode (Monosync and Bisync) with the crystal oscillator not selected, these pins act as outputs and are active only during the part of the receive clock cycle in which synchronous characters are recognized. The synchronous condition is not latched, so these outputs are active each time a synchronization pattern is recognized (dregardless of character boundaries). In SDLC mode, these pins act as outputs and are valid on receipt of a flag.

RxDA, RxDB Receive Data Inputs
These input signals receive serial data at standard TTL levels.

TxDA, TxDB Transmit Data Outputs
These output signals transmit serial data at standard TTL levels.

RTSA#, RTSB# Request to Send Outputs
When the Request to Send (RTS) bit in Write Register 5 is set, the RTS# signal goes LOW. When the RTS bit is reset in the asynchronous mode and Auto Enable is on, the signal goes HIGH after the transmitter is empty. In synchronous mode or in asynchronous mode with Auto Enable off, the RTS# pin strictly follows the state of the RTS bit. Both pins can be used as general-purpose outputs.

REQA#, REQB# Transmit Request for Channel A/B Outputs
Request lines for a DMA controller.

### 3.3.4.2 Internal ESCC System Interface

INTA#, INTB# Interrupt Request Output
Interrupt Request (open drain, active low). This signal is activated when the ESCC activates an interrupt.

CE1# Circuit Enable Input

Circuit Enable for the ESCC Subsection. This signal selects the ESCC for a read or write operation.

W/REQA#, W/REQB# Wait/Request Outputs
Wait/Request (open-drain when programmed for a Wait function, driven HIGH or LOW when programmed for a Request function). These dual-purpose outputs may be programmed as Request lines for a DMA controller or as Wait lines to synchronize the CPU to the ESCC data rate. The reset state is Wait.

### 3.3.5 SBP Subsection

#### 3.3.5.1 SBP Interface

SFS Serial Frame Synchronization Input
SFS is an 8 kHz signal which identifies the beginning of each frame by a low to high transition. The 192kbps data stream on SBIN and SBOUT is referenced to SFS.

CH2STRB SBP Channel 2 Strobe Input
This signal is active during the 8-bit times of the second 64kbps data channel.

SCLK Serial Data Clock Input
This signal carries the 192kHz data clock. Input and Output data will assumed to be valid at the rising edge of this signal.

SBIN Serial Data In Output
This pin outputs data to the SBIN input of the external 79C30A device.

SBOUT Serial Data Out Input
This pin receives data from the SBOUT output of the external 79C30A device.

#### 3.3.5.2 External SCC Interface

TxDC, TxDD Transmit Data Inputs
Transmit Data inputs from External SCC channels C and D.

RxDC, RxDD Receive Data Outputs
Receive Data inputs to External SCC channels C and D.

RTxCC#, RTxCD# Receive/Transmit Clocks Outputs
Receive / Transmit Clocks for External SCC channels C and D.

#### 3.3.5.3 SBP System Interface

CE2# Circuit Enable Input
Circuit Enable for the SBP Interface subsection (active low). This signal selects the SBP Interface for a read or write operation.

### 3.3.6 8-Bit Host System Interface

| | | |
|---|---|---|
| DATAB<7-0> | Data Bus | Input/Output/3-state |
| ESCC: | These lines carry data and commands to and from the ESCC. | |
| SBP: | This bus is used to transfer parallel data between the SBP Interface internal data buffers, as well as the command/status register. | |
| RD8# | Read Control Signal | Input |
| ESCC: | This signal indicates a read operation. During the Interrupt Acknowledge cycle, this signal gates the interrupt vector onto the bus if the ESCC is the highest priority device requesting an interrupt. | |
| SBP: | This signal enables the data inside the SBP Interface internal registers to be read through the DATAB bus. | |
| WR8# | Write Control Signal | Input |
| ESCC: | When the ESCC is selected, this signal indicates a write operation. The coincidence of RD8# and WR8# is interpreted as a reset. | |
| SBP: | This signal enables data on the DATAB bus to be written into the SBP Interface internal data registers, as well as the command/status register. | |
| A/B# | | Input |
| ESCC: | Channel A/ Channel B Select. This signal selects the channel in which the read or write operation occurs. | |
| SBP: | This signal, in conjunction with D/C#, RD8# and WR8#, selectively access the internal registers inside the SBP Interface subsection. | |
| D/C# | | Input |
| ESCC: | Data/Control Select. This signal defines the type of information transferred to or from the ESCC. A HIGH means data is transferred; a LOW indicates a command. | |
| SBP: | This signal, in conjunction with A/B#, RD8# and WR8#, selectively access the internal registers inside the SBP Interface subsection. | |
| INTACK# | | Input |
| ESCC: | Interrupt Acknowledge (active low). This signal indicates an active Interrupt Acknowledge cycle. When RD8# becomes active, the ESCC places an interrupt vector on the data bus. INTACK# is latched by the rising edge of PCLK. | |
| SBP: | | Not Applicable |
| PCLK | 8-bit Subsection System Clock | Input |
| E/N# | Extended Address Line | Input |
| ESCC: | When this pin is LOW, all internal ESCC registers are accessible. When this pin is HIGH, only the VERSION register and the LocalTalk/Extended FIFO enhancement register are accessible to the user. The address and definition of these registers are as follows: | |

VERSION register: It is located at address 3 of the enhanced address space. i.e., in order to access the VERSION register WR0 should contain the following 8 bit [7:0] code: "xxxx0011" and the E/N# line should be driven HIGH. (D/C# pin should be driven LOW.)

The content of the CURIO VERSION register follows the format: VERSION [7:0] = [ 1 1 x x x x x x ]; the lower 6 bits are used to indicate the rev ID of this CURIO circuit.

Write accesses to this register are IGNORED.

The LocalTalk/Extended FIFO enhancement register is located at address 2 of the enhanced address space. i.e., in order to access this register WR0 should contain the following 8 bit [7:0] code: "xxxx0010" and the E/N# pin should be driven HIGH. ( D/C# pin should be driven LOW). Both write and read accesses are recognized.

If the E/N# pin is driven HIGH, accesses to locations 0,1,4-15 are NOT permitted and will be read as zeros. (These locations are reserved for future extensions).

### 3.3.7 IEEE P1149.1 TAP Interface

TCK Test Clock Input

The clock input for the boundary scan test mode operation. TCK can operate from 1MHz to 10MHz.

TMS Test Mode Select Input

A serial input bit stream used to define the specific boundary scan test to be executed. The TMS input is sampled at the rising edge of TCK. This signal is internally pull-up through a 100Kohm resistor to Vcc.

TDI Test Data Input Input

The test data input path to the CURIO. Data is sampled at the rising edge of TCK.

TDO Test Data Out Output

The test data output path from the CURIO. The TDO data output changes at the falling edge of TCK.

TRST# Test Reset Input

Assertion of TRST# causes the IEEE P1149.1 TAP to be reset. This signal is internally pulled-up through a 100Kohm resistor to Vcc.

### 3.3.8 General Interface

| | | |
|---|---|---|
| XTAL1/XTAL2 | Crystal Input | Input |

The crystal frequency determines the network data rate. The network data rate is one-half of the crystal frequency. XTAL1 may alternatively be driven using an external TTL level source, in which case XTAL2 must be left unconnected.

| | | |
|---|---|---|
| XCLK | Crystal Clock | Output |

XCLK is derived from the crystal oscillator. The frequency of XCLK is equal to the crystal or XTAL1 frequency.

## 3.4 MACE FUNCTIONAL DESCRIPTION

The MACE can be connected to an 802.3 network via one of three network interfaces. The Attachment Unit Interface (AUI) provides an IEEE compliant differential interface to a remote MAU or an on-board transceiver. The Digital Attachment Interface™ (DAI™) can connect to local transceiver devices for 10BASE2, 10BASE-T or 10BASE-F connections. Additionally, a General Purpose Serial Interface (GPSI) is supported, which effectively bypasses the integrated Manchester encoder/decoder, and allows direct access to/from the integral 802.3 Media Access Controller (MAC) to provide support for external encoding/decoding schemes. The interface in use is determined by the PORTSEL [1-0] bits in the PLS Configuration Control register.

Designed with a slave type interface, the MACE circuit is intended to function as a standalone peripheral device to the host CPU, a DMA controller or an intelligent I/O processor. From the software stand point, it operates as a data conduit to and from the 802.3 network through register based function programming and interrupt driven event handling. The internal MAC controller executes the CSMA/CD communication algorithm without host intervention.

### 3.4.1 BASIC MACE FUNCTIONS

**MACE BLOCK DIAGRAM**

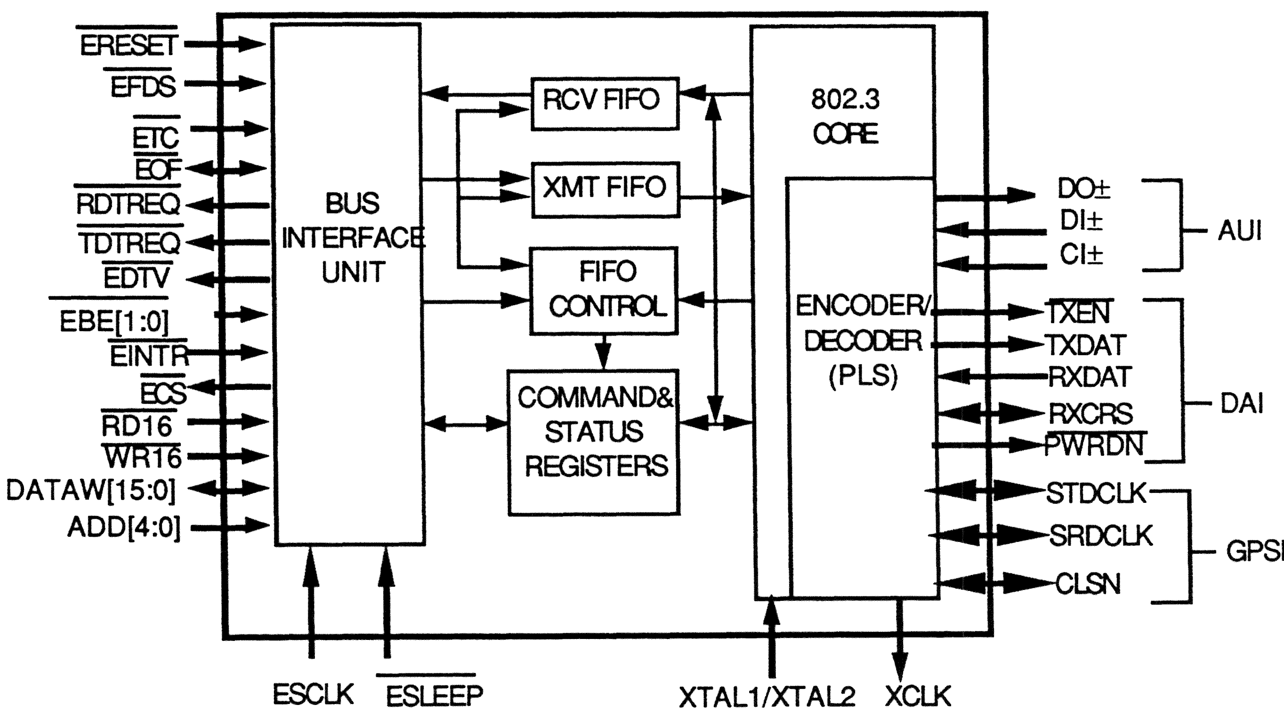



Figure A.1

3.4.1.1 Network Interfaces

The MACE subsection can be connected to an 802.3 network via one of two network interfaces. The Attachment Unit Interface (AUI) provides an IEEE compliant differential interface to a remote MAU or an on-board transceiver. The Digital Attachment Interface™ (DAI™) can connect to local transceiver devices for 10BASE2, 10BASE-T or 10BASE-F connections. The interface in use is determined by the PORTSEL [1-0] bits in the PLS Configuration Control register.

3.4.1.2 System Interface

The MACE is a register based peripheral. All transfers to and from the device, including data, are performed using a simple memory or I/O read or write commands. Access to all registers, including the transmit and receive FIFOs, is performed with identical read or write timing. All information on the system interface is synchronous to the system clock (ESCLK), which allows simple external logic to be designed to interrogate the device status and control the network data flow.

The receive and transmit FIFOs can be read or written by driving the appropriate address lines and asserting ECS# and RD16# or WR16# as appropriate. An alternative FIFO access mechanism allows use of the EFDS# and the RD16# or WR16# lines, without attention to the address lines ($ADD_{4-0}$). The state of the RD16#, WR16# lines in conjunction with the EFDS# input determines whether the receive FIFO is read (RD16# low) or the transmit FIFO written (WR16# low). The MACE system interface permits interleaved transmit and receive bus transfers to take place, allowing the transmit FIFO to be filled ("primed") while a frame is being received from the network and/or read from the receive FIFO.

In receive operation, the MACE asserts Receive Data Transfer Request (RDTREQ#) when the FIFO contains adequate data. For the first indication of a new receive frame, 64 bytes must be received, assuming normal operation. Once the initial 64 byte threshold has been reached, RDTREQ# assertion and de-assertion is dependent on the programming of the Receive FIFO Watermark (RCVFW bits in the BIU Configuration Control register). The RDTREQ# can be programmed to activate when there are 16, 32 or 64 bytes of data available in the FIFO. Enable Receive (ENRCV bit in MAC Configuration Control register) must be set to assert RDTREQ#. If the Runt Packet Accept feature is invoked (RPA bit in User Test Register), RDTREQ# will be asserted for receive frames of less that 64 bytes on the basis of internal and/or external address match only. When RPA is set, RDTREQ# will be asserted when the entire frame has been received or when the initial 64 byte threshold has been exceeded.

Note that the receive FIFO may not contain 64 data bytes at the time RDTREQ# is asserted, if the automatic pad stripping feature has been enabled (ASTRP RCV bit in the Receive Frame Control register) and a minimum length packet with pad is received. The MACE will check for the minimum received length from the network, strip the pad characters, and pass only the data frame through the receive FIFO.

In transmit operation, the MACE asserts Transmit Data Transfer Request (TDTREQ#) dependent on the programming of the Transmit FIFO Watermark (XMTFW bits in the BIU Configuration Control register). TDTREQ# will be permanently asserted when the transmit FIFO is empty. The TDTREQ# can be programmed to activate when there are 16, 32 or 64 bytes of space available in the transmit FIFO. Enable Transmit (ENXMT bit in MAC Configuration Control register) must be set to assert TDTREQ#. Write cycles to the Transmit FIFO will not return EDTV# if ENXMT is disabled, and no data will be written. The MACE will commence the preamble sequence once the Transmit Start Point (XMTSP bits in BIU Configuration Control register) threshold is reached in the transmit FIFO.

The transmit FIFO data will not be overwritten until at least 512 data bits have been transmitted onto the network. If a collision occurs within the slot time (512 bit time) window, the MACE will generate a jam sequence before ceasing the transmission, the transmit FIFO will be reset to point at the start of the transmit data field, and the message will be retried after the random back-off interval has expired.

### 3.4.2 Detailed MACE Functions

#### 3.4.2.1 Bus Interface Unit (BIU)

The BIU performs the interface between the host or system bus and the Transmit and Receive FIFOs, as well as all on board control and status registers. The BIU can be configured to accept data presented in either little-endian or big-endian format, minimizing the external logic required to access the MACE integral FIFOs and registers. In addition, the BIU directly supports 8-bit transfers and incorporates features to simplify interfacing to 32-bit systems using external latches.

Externally, the FIFOs appear as two independent registers located at individual addresses. The remainder of the internal registers occupy 30 additional addresses consecutively, and appear as 8-bits wide.

All regularly accessed registers, including the FIFOs, transmit and receive control and status information, and interrupts are located within a block of 8 contiguous addresses.

##### 3.4.2.1.1 BIU to FIFO Data Path

The BIU operates assuming that the 16-bit data path to/from the internal FIFOs is configured as two independent byte paths, activated by the Byte Enable signals $EBE_0$# and $EBE_1$#.

$EBE_0$# and $EBE_1$# are only used during accesses to the 16-bit wide Transmit and Receive FIFOs. After ERESET#, the BSWP bit will be cleared. FIFO accesses to the MACE will operate assuming an Intel type memory convention (most significant byte of a word stored in the higher addressed byte). Word data transfers to/from the FIFOs over the DATAW<15-0> lines will have the least significant byte located on $DATAW_{0-7}$ (activated by $EBE_0$#) and the most significant byte located on DATAW<15-8> (activated by $EBE_1$#).

FIFO data can be read or written using either byte and/or word operations.

If byte operation is required, read/write transfers can be performed on either the upper or lower data bus by asserting the appropriate byte enable. For instance with BSWP = 0, reading from or writing to DATAW<15-8> is accomplished by asserting $EBE_1$#, and allows the data stream to be read from or written to the appropriate FIFO in byte order (byte 0, byte 1,....byte n). It is equally valid to read or write the data stream using DATAW<7-0> and by asserting $EBE_0$#. For BSWP = 1, reading from or writing to DATAW<15-8> is accomplished by asserting $EBE_0$#, and allows the byte stream to be transferred in byte order.

When word operations are required, BSWP ensures that the byte ordering of the target memory is compatible with the 802.3 requirement to send/receive the data stream in byte ascending order. With BSWP = 0, the data transferred to/from the FIFO assumes that byte "n" will be on $DATAW_{0-7}$ (activated by $EBE_0$#) and byte "n+1" will be on $DATAW_{8-15}$ (activated by $EBE_1$#). With BSWP = 1, the data transferred to/from the FIFO assumes that byte "n" will be presented on $DATAW_{8-15}$ (activated by $EBE_0$#), and byte "n+1" will be on $DATAW_{0-7}$ (activated by $EBE_1$#).

There are some additional special cases to the above generalized rules, which are as follows :
(a) When performing byte read operations, both halves of the data bus are driven with identical data, effectively allowing the user to arbitrarily read from either the upper or lower data bus, when only one of the byte enables is activated.
(b) When byte write operations are performed, the Transmit FIFO latency is affected. See the "FIFO Sub-system" section for additional details.
(c) If a word read is performed when the last data byte is read for a receive frame (when the MACE activates the EOF# signal), such as when the message contained an odd number of bytes but the host requested a word operation by asserting both $EBE_0$# and $EBE_1$#, the MACE will present one valid and one non-valid byte on the data bus. The placement of valid data for the data byte is dependent on the target memory architecture. Regardless of BSWP, the single valid byte will be read from the $EBE_0$# memory bank. If BSWP = 0, $EBE_0$# corresponds to DATAW<7-0>; if BSWP = 1, $EBE_0$# corresponds to DATAW<15-8>.
(d) If a byte read is performed when the last data byte is read for a receive frame (when the MACE activates the EOF# signal), then the same byte will be presented on both the upper and lower byte of the data bus, regardless of which byte enable was activated (as is the case for all byte read operations).
(d) When writing the last byte in a transmit message to the transmit FIFO, the portion of the data bus that the last byte is transferred over is irrelevant, providing the appropriate byte enable is used. For BSWP = 0, data can be presented on DATAW<7-0> using $EBE_0$# or DATAW<15-8> using $EBE_1$#. For BSWP = 1, data can be presented on DATAW<7-0> using $EBE_1$# or DATAW<15-8> using $EBE_0$#.

| $EBE_0$ | $EBE_1$ | BSWP | DATAW<7-0> | DATAW<15-8> |
|---|---|---|---|---|
| 0 | 0 | 0 | n | n+1 |
| 0 | 1 | 0 | n | n |
| 1 | 0 | 0 | n | n |
| 1 | 1 | 0 | X | X |
| 0 | 0 | 1 | n+1 | n |
| 0 | 1 | 1 | n | n |
| 1 | 0 | 1 | n | n |
| 1 | 1 | 1 | X | X |

Table A.2 : Byte Alignment For FIFO Read Operations

| $EBE_0$ | $EBE_1$ | BSWP | DATAW<7-0> | DATAW<15-8> |
|---|---|---|---|---|
| 0 | 0 | 0 | n | n+1 |
| 0 | 1 | 0 | X | n |
| 1 | 0 | 0 | n | X |
| 1 | 1 | 0 | X | X |
| 0 | 0 | 1 | n+1 | n |
| 0 | 1 | 1 | X | n |
| 1 | 0 | 1 | n | X |
| 1 | 1 | 1 | X | X |

Table A.3 : Byte Alignment For FIFO Write Operations

3.4.2.1.2 BIU to Control and Status Register Data Path

Registers 2-31 perform data transfers on both bytes of the data bus, regardless of the programming of BSWP. <7-0><15-8>All accesses on registers 2-31 are independent of the $EBE_0$# and $EBE_1$# pins.

| $EBE_{0\#}$ | $EBE_{1\#}$ | BSWP | DATAW<7-0> | DATAW<15-8> |
|---|---|---|---|---|
| X | X | 0 | READ DATA | READ DATA |
| X | X | 1 | READ DATA | READ DATA |

Table A.4.1: Byte Alignment For Register Read Operations

| $EBE_{0\#}$ | $EBE_{1\#}$ | BSWP | DATAW<7-0> | DATAW<15-8> |
|---|---|---|---|---|
| X | X | 0 | WRITE DATA | X |
| X | X | 1 | X | WRITE DATA |

Table A.4.2 : Byte Alignment For Register Write Operations

3.4.2.2 FIFO Sub-system

The MACE provides two independent 128 byte FIFOs for receive and transmit operations. The FIFO sub-system contains both the FIFOs, and the control logic to handle normal and exception related conditions.

The Transmit and Receive FIFOs interface on the network side with the serializer/de-serializer in the MAC engine. The BIU provide access to and from the FIFOs from the host system to enable the movement of data to and from the network via the FIFOs.

Internally, the FIFOs appear to the BIU as independent 16-bit wide registers. Bytes or words can be written to the Transmit FIFO, or read from the Receive FIFO. The BIU will ensure correct byte ordering dependent on the target host system, as determined by the programming of the BSWP bit in the BIU Configuration Control register.

However, when writing bytes to the XMTFIFO, certain restrictions apply which will have a direct influence on the latency the FIFO is able to provide to the host system. When a byte is written to the word FIFO location, the entire word location is used. The unused byte is marked as a "hole"in the XMTFIFO. These "holes" are skipped during the serialization process performed by the MAC engine, as the bytes are unloaded from the FIFO.

For instance, assume the Transmit FIFO Watermark (XMTFW) is set for 32 write cycles. If the host writes byte wide data to the FIFO, the TDTREQ# will de-assert once 32 bytes are present in the FIFO. Transmission will not commence until 64-bytes or the "End-of-Frame" are available in the FIFO, so transmission would not start, and TDTREQ# would remain de-asserted. Hence for byte wide data transfers, XMTFW should be programmed to the 8 or 16 write cycle limit, or the host should ensure that sufficient data will be written to the FIFO after TDTREQ# has been dropped, to guarantee that the transmission will commence. A third alternative is to program the Transmit Start Point (XMTSP) in the BIU Configuration Control register to below the 64-byte default. This will impose a lower latency to the host system, from the point of view that the MACE will require additional data to ensure the FIFO does not underflow during the transmit process, versus using the default XMTSP value. Note that if 64 single byte writes are executed on the XMTFIFO, and the XMTSP is set to 64-bytes, the transmission will commence, and all 64-bytes of information will be accepted by the transmit FIFO.

As a second example. assume again that the XMTFW is programmed for 32 write cycles. If the host writes word wide data to the FIFO, the TDTREQ# will de-assert once the 32 writes have executed on the XMTFIFO, at which point there will be 64-bytes present. TDTREQ# will not re-assert until the transmission of the packet has commenced and the possibility of a collision within the "slot time" is removed (512 bits have been transmitted without a collision indication). At this point there will be only 8 bytes of date remaining in the FIFO (8 bytes of preamble/SFD plus 56 bytes of data have been transmitted), which corresponds to 6.4µs of latency before a FIFO underrun could occur. This is considerably less than the possible 51.2µs the system may have been assuming.

The number of write cycles that the host uses to write the packet into the Transmit FIFO will also directly influence the amount of space utilized by the transmit message. If the number of write cycles (n) required to transfer a packet to the Transmit FIFO is even, the number of bytes used in the Transmit FIFO will be 2n. If the number of write cycles required to transfer a packet to the Transmit FIFO is odd, the number of bytes used in the transmit FIFO will be 2n+2. This is due to the “End Of Frame” indication in the FIFO always being placed at the end of a 4-byte boundary. As an example, a 32-byte message written as bytes will use 64-bytes of space in the Transmit FIFO, whereas a 65-byte message written as 32 words and 1 byte (33 cycles) would use 68-bytes

### 3.4.2.3 Media Access Control (MAC)

The Media Access Control engine is the heart of the MACE, incorporating the essential protocol requirements for operation of a compliant Ethernet/802.3 node, and providing the interface between the FIFO sub-system and the Manchester Encoder/Decoder (MENDEC).

The MAC engine is fully compliant to Section 4 of ISO/IEC 8802-3 (ANSI/IEEE Standard 1990 Second edition) and ANSI/IEEE 802.3 (1985).

The MAC engine provides enhanced features, programmed through the Transmit Frame Control register, designed to minimize host supervision and pre or post message processing. These include the ability to disable retries after a collision, dynamic FCS generation on a packet-by-packet basis, and automatic pad field insertion and deletion to enforce minimum frame size attributes.

The two primary attributes of the MAC engine are:

(1) Transmit and receive message data encapsulation.
- (i) Framing (frame boundary delimitation, frame synchronization).
- (ii) Addressing (source and destination address handling).
- (iii) Error detection (physical medium transmission errors).

(2) Media access management.
- (i) Medium allocation (collision avoidance).
- (ii) Contention resolution (collision handling).

(1) Transmit and receive message data encapsulation.

(i) Framing (frame boundary delimitation, frame synchronization).

(ii) Addressing (source and destination address handling).

(iii) Error detection (physical medium transmission errors).

(2) Media Access Management.
The basic requirement for all stations on the network is to provide fairness of channel allocation. The 802.3/Ethernet protocols define a media access mechanism which permits all stations to access the channel with equality. Any node can attempt to contend for the channel by waiting for a predetermined time (Inter Packet Gap interval) after the last activity, before transmitting on the media. The channel is a bus or multidrop communications medium (with various topological configurations permitted) which allows a single station to transmit and all other stations to receive. If two nodes simultaneously contend for the channel, their signals will interact causing loss of data, defined as a collision. It is the responsibility of the MAC to attempt to avoid and recover from a collision, to guarantee data integrity for the end-to-end transmission to the receiving station.

(i) Medium Allocation (collision avoidance)
The MAC engine implements the optional receive two part deferral algorithm, with an first part inter-frame-spacing time of 6.0µs. The second part of the inter-frame-spacing interval is therefore 3.6µs. In addition, transmit two part deferral is implemented as an option which can be disabled using the DXMT2PD bit in the MAC Configuration Control register. Two part deferral after transmission is useful for ensuring that severe IPG shrinkage cannot occur in specific circumstances, causing a transmit message to follow a receive message so closely, as to make them indistinguishable.

The IEEE 802.3 Standard (ISO/IEC 8802-3 1990) requires that the CSMA/CD MAC monitors the medium for traffic by watching for carrier activity. When carrier is detected, the media is considered busy, and the MAC should defer to the existing message.

The IEEE 802.3 Standard also allows optional two part deferral after a receive message.

**See ANSI/IEEE Std 802.3-1990 Edition, 4.2.3.2.1:**

"NOTE : It is possible for the PLS carrier sense indication to fail to be asserted during a collision on the media. If the deference process simply times the interFrame gap based on this indication it is possible for a short interFrame gap to be generated, leading to a potential reception failure of a subsequent frame. To enhance system robustness the following optional measures, as specified in 4.2.8, are recommended when interFrameSpacingPart1 is other than zero:

(1) Upon completing a transmission, start timing the interpacket gap, as soon as transmitting and carrierSense are both false.

(2) When timing an interFrame gap following reception, reset the interFrame gap timing if carrierSense becomes true during the first 2/3 of the interFrame gap timing interval.During the final 1/3 of the interval the timer shall not be reset to ensure fair access to the medium. An initial period shorter than 2/3 of the interval is permissible including zero."

The MACE will perform the two part deferral algorithm as specified in Section 4.2.8 (Process Deference). The Inter Packet Gap (IPG) timer will start timing the 9.6µs InterFrameSpacing after the receive carrier is de-asserted. During the first part deferral (InterFrameSpacingPart1 - IFS1) the MACE will defer any pending transmit frame and respond to the receive message. The IPG counter will be reset to zero continuously until the carrier de-asserts, at which point the IPG counter will resume the 9.6µs count once again. Once the IFS1 period of 6.0µs has elapsed, the MACE will begin timing the second part deferral (InterFrameSpacingPart2 - IFS2) of 3.6µs. Once IFS1 has completed, and IFS1 has commenced, the MACE will not defer to a receive packet if a transmit packet is pending. This means that the MACE will not attempt to receive the receive packet, since it will start to transmit, and generate a collision at 9.6µs. The MACE will guarantee to complete the preamble (64-bit) and jam (32-bit) sequence before ceasing transmission and invoking the random backoff algorithm.

During the time period immediately after a transmission has been completed, the external transceiver (in the case of a standard AUI connected device), should generate the SQE Test message (a nominal 10MHz burst of 5-15 BT duration) on the CI± pair (within 0.6 - 1.6µs after the transmission ceases). During the time period in which the SQE Test message is expected the MACE will not respond to receive carrier sense.

**See ANSI/IEEE Std 802.3-1990 Edition, 7.2.4.6 (1)):**

"At the conclusion of the output function, the DTE opens a time window during which it expects to see the *signal_quality_error* signal asserted on the Control In circuit. The time window begins when the CARRIER_STATUS becomes CARRIER_OFF. If execution of the output function does not cause CARRIER_ON to occur, no SQE test occurs in the DTE. The duration of the window shall be at least 4.0µs but no more than 8.0µs. During the time window the Carrier Sense Function is inhibited."

The MACE implements a carrier sense "blinding" period within 0 - 4.0µs from de-assertion of carrier sense after transmission. This effectively means that when transmit two part deferral is enabled (DXMT2PD in the MAC Configuration Control register is cleared) the IFS1 time is from 4µs to 6µs after a transmission. However, since IPG shrinkage below 4µs will not be encountered on correctly configured networks, and since the fragment size will be larger than the 4µs blinding window, then the IPG counter will be reset by a worst case IPG shrinkage/fragment scenario and the MACE will defer its transmission. The MACE will not restart the carrier sense "blinding" period if carrier is detected within the 4.0 - 6.0μs portion of IFS1, but will restart timing of the entire IFS1 period.

(ii) Contention resolution (collision handling).
Collision detection is performed and reported to the MAC engine either by the integrated Manchester Encoder/Decoder (MENDEC), or by use of an external function (e.g. Serial Interface Adaptor, Am7992B) utilizing the GPSI.

If a collision is detected before the complete preamble/SFD sequence has been transmitted, the MACE will complete the preamble/SFD before appending the jam sequence. If a collision is detected after the preamble/SFD has been completed, but prior to 512 bits being transmitted, the MACE will abort the transmission, and append the jam sequence immediately. The jam sequence is a 32-bit all zeroes pattern.

The MACE will attempt to transmit a frame a total of 16 times (initial attempt plus 15 retries) due to normal collisions (those within the slot time). Detection of collision will cause the transmission to be re-scheduled, dependent on the backoff time that the MACE computes. Each collision which occurs during the transmission process will cause the value of XMTRC in the Transmit Retry Count register to be updated. If a single retry was required, the ONE bit will be set in the Transmit Frame Status. If more than one retry was required, the MORE bit will be set, and the exact number of attempts can be determined (XMTRC+1). If all 16 attempts experienced collisions, the RTRY bit will be set, and the transmit message will be flushed from the FIFO, either by resetting the FIFO (if no "End-of-Frame" tag exists) or by moving the FIFO read pointer to the next free location (If an "End-of-Frame" tag is present).

If a collision is detected after 512 bit times have been transmitted, the collision is termed a late collision. The MACE will abort the transmission, append the jam sequence and set the LCOL bit in the Transmit Frame Status. No retry attempt will be scheduled on detection of a late collision, and the FIFO will be flushed.

The IEEE 802.3 Standard requires use of a "truncated binary exponential backoff" algorithm which provides a controlled pseudo random mechanism to enforce the collision backoff interval, before re-transmission is attempted..

**See ANSI/IEEE Std 802.3-1990 Edition, 4.2.3.2.5:**

"At the end of enforcing a collision (jamming), the CSMA/CD sublayer delays before attempting to re-transmit the frame. The delay is an integer multiple of slotTime. The number of slot times to delay before the nth re-transmission attempt is chosen as a uniformly distributed random integer r in the range:

$0 \leq r \leq 2^k$

where

$k = \min(n,10)$."

The MACE implements a random number generator, configured to ensure that nodes experiencing a collision, will not have their retry intervals track identically, causing retry errors.

The MACE provides an alternative algorithm, which suspends the counting of the slot time/IPG during the time that receive carrier sense is detected. This aids in networks where large numbers of nodes are present, and numerous nodes can be in collision. It effectively accelerates the increase in the backoff time in busy networks, and allows nodes not involved in the collision to access the channel whilst the colliding nodes await a reduction in channel activity. Once channel activity is reduced, the nodes resolving the collision time-out their slot time counters as normal.

If a receive message suffers a collision, it will be either a runt, in which case it will be deleted in the Receive FIFO, or it will be marked as a recieve late collision, using the CLSN bit in the Recieve Frame Status register. All frames which suffer a collision within the slot time will be deleted in the Receive FIFO without requesting host intervention, providing that the RCVFW bits (FIFO Configuration Control) are programmed to require 64 bytes are received prior to the assertion of RDTREQ#. Runt packets which suffer a collision will be aborted regardless of the state of the RPA bit (User Test Register).

### 3.4.2.4 Manchester Encoder/Decoder (MENDEC)

The integrated Manchester Encoder/Decoder provides the PLS (Physical Layer Signalling) functions required for a fully compliant IEEE 802.3 station. The MENDEC block contains the AUI and DAI™ interfaces, both of which transfer data to appropriate transceiver devices in Manchester encoded format. The MENDEC provides the encoding function for data to be transmitted on the network using the high accuracy on-board oscillator, driven by either the crystal oscillator or an external CMOS level compatible clock. The MENDEC also provides the decoding function from data received from the network. The MENDEC contains a Power On Reset (POR) circuit, which ensures that all analog portions of the MACE are forced into their correct state during power up, and prevents erroneous data transmission and/or reception during this time.

#### 3.4.2.4.1 Attachment Unit Interface (AUI)

The AUI is the PLS (Physical Layer Signalling) to PMA (Physical Medium Attachment) interface which effectively connects the DTE to the MAU. The differential interface provided by the MACE is fully compliant to Section 7 of ISO 8802-3 (ANSI/IEEE 802.3).

After the MACE initiates a transmission it will expect to see data "looped-back" on the DI± pair (AUI port selected). This will internally generate a "carrier sense", indicating that the integrity of the data path to and from the MAU is intact, and that the MAU is operating correctly. This "carrier sense" signal must be asserted within 9 bit times after the first transmitted bit on DO± (when using the AUI port). If "carrier sense" does not become active in response to the data transmission, or becomes inactive before the end of transmission, the loss of carrier (LCAR) error bit will be set in the Transmit Frame Status (bit 7) after the packet has been transmitted.

#### 3.4.2.4.2 Digital Attachment Interface™ (DAI™)

The Digital Attachment Interface™ is a simplified electrical attachment specification which allows MAUs which do not require the DC isolation between the MAU and DTE (e.g. devices compatible with the 10BASE-T and 10BASE-F Draft documents) to be implemented. All data transferred across the DAI™ is Manchester Encoded. Decoding and encoding is performed by the MENDEC.

The DAI™ will accept receive data on the basis that the RXCRS input is active, and will take the data presented on the RXDAT input as valid Manchester data. Transmit data is sent to the external transceiver by the MACE asserting TXEN# and presenting complimentary data on the TXDAT± pair. During idle, the MACE will assert the TXDAT+ line high, and the TXDAT- line low, while TXEN# is maintained inactive (high). The MACE implements "logical collision detection" and will use the simultaneous assertion of TXEN# and RXCRS to internally detect a collision condition, take appropriate internal action (such as abort the current transmit or receive activity), and provide external indication using the CLSN pin. Any external transceiver utilized for the DAI™ interface must not loop back the transmit date (presented by the MACE) on the TXDAT± pins to the RXDAT pin. Neither should the transceiver assert the RXCRS pin when transmitting data to the network. Duplication of these functions by the external transceiver (unless the MACE is in the external loop back test configuration) will cause false collision indications to be detected.

In order to provide an integrity test of the connectivity between the MACE an the external transceiver similar to the SQE Test Message provided as a part of the AUI functionality, the MACE can be programmed to operate the DAI™ port in an external loopback test. In this case, the external transceiver is assumed to loopback the TXDAT± data stream to the RXDAT pin, and assert RXCRS in response to the TXEN# request. When in the external loopback mode of operation (programmed by LOOP [1-0] = 01), the MACE will not internally detect a collision condition. The external transceiver is assumed to take action to ensure that this test will not disrupt the network. This type of test is intended to be operated for a very limited period (e.g. after power up), since the transceiver is assumed to be located physically close to the MACE and with minimal risk of disconnection (e.g. connected via printed circuit board traces).

Note that when the DAI™ port is selected, LCAR errors will not occur, since the MACE will internally loop back the transmit data path to the receiver. This loop back function must not be dupliacated by a transceiver which is externally connected via the DAI™ port, since this will result in a condition where a collision is generated during any transmit activity.

### 3.4.2.5 General Purpose Serial Interface (GPSI) Extension

The GPSI Extension provides the additional signals necessary to present an interface consistent with the non encoded data functions observed to/from a LAN controller such as the Am7990 Local Area Network Controller for Ethernet (LANCE). Combined with some of the pins from the DAI™ port, the GPSI Extension replicates this type of interface.

The GPSI allows use of an external Manchester encoder/decoder, such as the Am7992B Serial Interface Adapter (SIA). In addition, it allows the MACE to be used as a MAC sublayer engine in a repeater based on the Am79C980 Integrated Multiport Repeater (IMR). Simple connection to the IMR Expansion Bus allows the MAC to view all packet data passing through a number of interconnected IMRs, allowing statistics and network management information to be collected.

The GPSI functional pins are duplicated as follows:

| FUNCTION | TYPE | LANCE PIN | MACE PIN |
|---|---|---|---|
| Receive Data | I | RX | RXDAT |
| Receive Clock | I | RCLK | SRDCLK |
| Receive Carrier Sense | I | RENA | RXCRS |
| Collision | I | CLSN | CLSN |
| Transmit Data | O | TX | TXDAT+ |
| Transmit Clock | I | TCLK | STDCLK |
| Transmit Enable | O | TENA | TXEN |

Table 2-17 : Pin Configuration for GPSI Function

3.4.2.6 Slave Access Operation

Internal register accesses are based on a 2 or 3 ESCLK cycle duration, dependent on the state of the ETC# input pin. ETC# must be externally tied low to force the MACE to perform 3 cycle accesses. ETC# is internally pulled high if left unconnected, to configure the 2 cycle access by default.

All register accesses are byte wide with the exception of the data path to and from the internal FIFOs.

Data exchanges to/from register locations will take place over the appropriate half of the data bus to suit the host memory organization (as programmed by the BSWP bit in the BIU Configuration Control register).

The $EBE_0$#, $EBE_1$# and EOF# signals are provided to allow control of the data flow to and from the FIFOs. Byte read operations from the receive FIFO cause data to be duplicated on both the upper and lower bytes of the data bus. Byte write operations to the transmit FIFO must use the $EBE_0$# and $EBE_1$# inputs to define the active data byte to the MACE.

3.4.2.6.1 Read Access

Details of the read access timing are located in section 4.2.

ETC# can be dynamically changed on a cycle by cycle basis to program the slave cycle execution for 2 (ETC# = high) or 3 (ETC# = low) ESCLK cycles. ETC# must be stable by the falling edge of ESCLK in S0 at the start of a cycle, and should only be changed in S0 in a multiple cycle burst.

A read cycle is initiated when either ECS# or EFDS# is sampled low on the falling edge of ESCLK at S0. EFDS# and ECS# must be asserted exclusively. If they are active simultaneously when sampled, the MACE will not execute any read or write cycle.

If ECS# is low, a Register Address read will take place. The state of the $ADD_{0-4}$ will be used to commence decoding of the appropriate internal register/FIFO.

If EFDS# is low, a FIFO Direct read will take place from the receive FIFO. The state of the $ADD_{0-4}$ bus is irrelevant for the FIFO Direct mode.

With either the ECS# or EFDS# input active, the state of the $ADD_{0-4}$ (for Register Address reads), RD16# (low to indicate a read cycle), $EBE_0$ and $EBE_1$ will also be latched on the falling edge of ESCLK at S0.

From the falling edge of ESCLK in S1, the MACE will drive data on DATAW<15-0> and activate the EDTV# output (providing the read cycle completed successfully). If the cycle read the last byte/word of data for a specific frame from the FIFO, the MACE will also assert the EOF# signal. DATAW<15-0>, EDTV# and EOF# will be guaranteed valid and can be sampled on the falling edge of ESCLK at S2.

If the Register Address mode is being used to access the FIFO, once EOF# is asserted during the last byte/word read for the frame, the Receive Frame Status can be read in one of two ways. The Register Address mode can be continued, by placing the appropriate address (00110b) on the address bus and executing 4 read cycles (ECS# active) on the Receive Frame Status location. In this case, additional Register Address read requests from the Receive FIFO will be ignored, and no EDTV# returned, until all 4 bytes of the Receive Frame Status register have been read. Alternatively, a FIFO Direct read can be performed, which will effectively route the Receive Frame Status through the Receive FIFO location. This mechanism is explained in more detail below.

If the FIFO Direct mode is used, the Receive Frame Status can be read directly from the FIFO by continuing to execute read cycles (by keeping EFDS# low and RD16# low) after EOF# is asserted indicating the last byte/word read for the frame. Each of the 4 bytes of Receive Frame Status will appear on both halves of the data bus, as if the actual Receive Frame Status register were being accessed. Alternatively, the status can be read as normal using the Register Address mode, by placing the appropriate address (00110b) on the address bus and executing 4 read cycles (ECS# active).

Either the FIFO Direct or Register Address modes can be interleaved at any time to read the Receive Frame Status, although this is considered unlikely due to the additional overhead it requires. In either case, no additional data will be read from the Receive FIFO until the Receive Frame Status has been read; as 4 bytes appended to the end of the packet when using the FIFO Direct mode, or as 4 bytes from the Receive Frame Status location when using the Register Address mode.

EOF# will only be driven by the MACE when reading received packet data from the FIFO. At all other times, including reading the Receive Frame Status using the FIFO Direct mode, the MACE will place EOF# in a high impedance state.

RDTREQ# should be sampled on the falling edge of ESCLK. The assertion of RDTREQ# is programmed by RCVFW, and the de-assertion is modified dependent on the state of the RCVBRST bit (both in the FIFO Configuration Control register). See the section "Receive FIFO Read" for additional details.

3.4.2.6.2 Write Access

Details of the read access timing are located in section 4.2.

Write cycles are executed in a similar from to the read cycle previously described, but with the WR16# input low, and the host responsible to provide the data with sufficient set up to the falling edge of ESCLK after S2.

After a FIFO write, TDTREQ# should be sampled on or after the falling edge of ESCLK after S3 of the FIFO write. The state of TDTREQ# at this time will reflect the state of the FIFO.

After going active (low), TDTREQ# will remain low for 2 or more FIFO writes.

The minimum high (inactive) time of TDTREQ# is one ESCLK cycle. When EOF# is written to the transmit FIFO, TDTREQ# will go inactive for a minimum of one ESCLK cycle.

3.4.2.7 Transmit Operation

The transmit operation and features of the MACE are controlled by programmable options. These options are programmed through the BIU, FIFO and MAC Configuration Control registers.

Parameters controlled by the MAC Configuration Control register are generally programmed only once, during initialization, and are therefore static during the normal operation of the MACE (see the Media Access Control section for a detailed description). The features controlled by the FIFO Configuration Control register and the Transmit Frame Control register can be re-programmed if the MACE is not transmitting.

3.4.2.7.1 Transmit FIFO Write

The Transmit FIFO is accessed by performing a host generated write sequence on the MACE. See section 4.2 for details of the write access timing.

There are two fundamentally different access methods to write data into the FIFO. Using the Register Address mode, the FIFO can be addressed using the $ADD_{0-4}$ lines, (address 00001b), initiating the cycle with the ECS# and WR16# signals. The FIFO Direct mode allows write access to the Transmit FIFO without use of the address lines, and using only the EFDS# and WR16# lines. If the MACE detects both signals active, it will not execute a write cycle. The write cycle timing for the Register Address or Direct FIFO modes are identical. EFDS# and ECS# should be mutually exclusive.

The data stream to the transmit FIFO is written using multiple byte and/or word writes. ECS# or EFDS# does not have to be returned inactive to commence execution of the next write cycle. If ECS/EFDS# is detected low at the falling edge of S0, a write cycle will commence. Note that EOF# must be asserted by the host/controller during the last byte/word transfer.

3.4.2.7.2 Transmit Function Programming

The Transmit Frame Control register allows programming of dynamic transmit attributes. Automatic transmit features such as retry on collision, FCS generation/transmission and pad field insertion can all be programmed, to provide flexibility in the (re-)transmission of messages.

The disable retry on collision (DRTRY bit) and automatic pad field insertion (APAD XMT bit) features should not be changed whilst data remains in the transmit FIFO. Writing to either the DRTRY or APAD XMT bits in this case may have unpredictable results. These bits are not internally latched or protected. When writing to the Transmit Frame Control register the DRTRY and APAD XMT bits should be programmed consistently. Once the transmit FIFO is empty, DRTRY and DXMTFCS can be reprogrammed.

This can be achieved with no risk of transmit data loss or corruption, by clearing ENXMT after the packet data for the current frame has been completely loaded. The transmission will complete normally and the activation of the EINTR# can be used to determine if the transmit frame has completed (XMTINT will be set in the Interrupt Register). Once the Transmit Frame Status has been read, APAD XMT and/or DRTRY can be changed, and ENXMT set to restart the transmit process with the new parameters.

APAD XMT is sampled if there are less than 60 bytes in the transmit packet when the last bit of the last byte is transmitted. If APAD XMT is set, a pad field of pattern "11111111" is added until the minimum frame size of 64 bytes (excluding preamble and SFD) is achieved. If APAD XMT is clear, no pad field insertion will take place and runt packet transmission is possible. When APAD XMT is enabled, the DXMTFCS feature is over-ridden, and the 4 byte FCS will be added to the transmitted packet unconditionally.

The disable FCS generation/transmission feature can be programmed dynamically on a packet by packet basis. The current state of the DXMTFCS bit is internally latched on the last write to the transmit FIFO, when the EOF# indication is asserted by the host/controller.

The programming of static transmit attributes are distributed between the BIU, FIFO and MAC Configuration Control registers.

The point at which transmission begins in relation to the number of bytes of a frame in the FIFO is controlled by the XMTSP bits in the BIU Configuration Control register. Depending on the bus latency of the system, XMTSP can be set to ensure that the transmit FIFO does not underflow before more data is written to the FIFO. When the entire frame is in the FIFO, transmission of preamble will commence regardless of the value in XMTSP. The default value of XMTSP is 64 bytes after ERESET#.

The point at which TDTREQ# is asserted in relation to the number of empty bytes present in the transmit FIFO is controlled by the XMTFW bits in the FIFO Configuration Control register. TDTREQ# will be asserted when one of the following conditions is true :

( i ) The number of bytes free in the Transmit FIFO relative to the current "Saved Read Pointer" value is greater than or equal to the threshold set by the XMTFW (16, 32 or 64 bytes). The "Saved Read Pointer" is the first byte of the current transmit frame, either in progress or awaiting channel availability.

(ii) The number of bytes free in the Transmit FIFO relative to the current "Read Pointer" value is greater than or equal to the threshold set by the XMTFW (16, 32 or 64 bytes). The "Read Pointer" becomes available only after a minimum of 64 byte frame length has been transmitted on the network (8 bytes of preamble plus 56 bytes of data), and points to the current byte of the transmit frame in progress.

Depending on the bus latency of the system, XMTFW can be set to ensure that the transmit FIFO does not underflow before more data is written into the FIFO. When the entire frame is in the FIFO, TDTREQ# will remain asserted if sufficient bytes remain empty. The default value of XMTFW is 64 bytes after ERESET#. Note that if the XMTFW is set below the 64 byte limit, the transmit latency for the host to service the MACE is effectively increased, since TDTREQ# will occur earlier in the transmit sequence, and more bytes will be present in the transmit FIFO when the TDTREQ# is de-asserted.

The transmit operation of the MACE can be halted at any time by clearing the ENXMT bit (bit 1) in the MAC Configuration Control register. Note that any complete transmit frame that is in the Transmit FIFO and is currently in progress will complete, prior to the transmit function halting. Transmit frames in the FIFO which have not commenced will not be started. Transmit frames which have commenced but which have not been fully transferred into the Transmit FIFO will be aborted, in one of two ways. If less that 512 bits have been transmitted onto the network the transmission will be terminated immediately, generating a runt packet which can be deleted at the receiving station. If greater than 512 bits have been transmitted, the messages will have the current CRC inverted and appended at the next byte boundary, to guarantee an error is detected at the receiving station. This ensures that packets will not be generated with potential undetected data corruption.

### 3.4.2.7.3 Automatic Pad Generation

Transmit frames can be automatically padded to extend them to 64 data bytes (excluding preamble). This allows the minimum frame size of 64 bytes (512 bits) for 802.3/Ethernet to be guaranteed, with no software intervention from the host/controlling process. APAD XMT = 1 enables the automatic padding feature. The pad is placed between the length field and FCS field in the 802.3 frame. FCS is always added if the frame is padded, regardless of the state of DXMTFCS. The transmit frame will be padded by bytes with the value of FFh. The default value of APAD XMT will enable auto pad generation after ERESET#.

It is the responsibility of upper layer software to correctly define the actual length field in the message, to correspond to the total number of LLC Data bytes containd in the message (length field as defined in the IEEE 802.3 standard). This value is not used by the MACE to compute the actual number of pad bytes to be inserted. The MACE will append pad bytes dependent on the actual byte count contained in the message.

| PREAMBLE 1010....1010 | SYNCH 10101011 | DEST. ADDR. | SRCE. ADDR. | LENGTH | LLC DATA | PAD | FCS |
|---|---|---|---|---|---|---|---|
| 56 BITS | 8 BITS | 6 BYTES | 6 BYTES | 2 BYTES | 46-1500 BYTES | | 4 BYTES |

MACE 802.3 FRAME.MD2

The Ethernet specification makes no use of the LLC pad field, and assumes that minimum length messages will be at least 64 bytes in length. Since the Type field is used instead of the Length field, stripping cannot be achieved by a receiving MACE, since it utilizes the Length field to accomplish this. For this reason, Ethernet frames should not use the APAD XMT feature.

| PREAMBLE 1010....1010 | SYNCH 11 | DEST. ADDR. | SRCE. ADDR. | TYPE | DATA | FCS |
|---|---|---|---|---|---|---|
| 62 BITS | 8 BITS | 6 BYTES | 6 BYTES | 2 BYTES | 46-1500 BYTES | 4 BYTES |

MACE E'NET FRAME.MD2

3.4.2.7.4 Transmit FCS Generation

Automatic generation and transmission of FCS for a transmit frame depends on the value of DXMTFCS (Disable Transmit FCS) when the EOF# is asserted indicating the last byte/word of data for the transmit frame is being written to the FIFO. The action of writing the last data byte/word of the transmit frame, latches the current contents of the Transmit Frame Control register, and therefore determines the programming of DXMTFCS for the transmit frame. When DXMTFCS = 0 the transmitter will generate and append the FCS to the transmitted frame. If the automatic padding feature is invoked (APAD XMT in Transmit Frame Control), the FCS will be appended regardless of the state of DXMTFCS. Note that the calculated FCS is transmitted most significant bit first. The default value of DXMTFCS is 0 after ERESET#.

3.4.2.7.5 Transmit Status Information

Although multiple transmit frames can be queued in the transmit FIFO, the MACE will not permit loss of Transmit Frame Status information. The Transmit Frame Status can only be maintained internally for a maximum of two frames. The MACE will therefor not commence a third transmit frame, until the status from the first frame is read. Once the Transmit Frame Status for the first transmit packet is read, the MACE will autonomously commence the next transmit frame, providing that a transmit frame is pending, the XMTSP threshold has been exceeded, the network medium is free, and the IPG time has elapsed.

3.4.2.7.6 Transmit Exception Conditions

Exception conditions for frame transmission fall into two distinct categories; those which are the result of normal network operation, and those which occur due to abnormal network and/or host related events.

Normal events which may occur and which are handled autonomously by the MACE are basically collisions within the slot time with automatic retry. The MACE will ensure that collisions which occur within 512 bit times from the start of transmission (including preamble) will be automatically retried with no host intervention. The transmit FIFO ensures this by guaranteeing that data contained within the FIFO will not be overwritten until at least 64 bytes (512 bits) of data have been successfully transmitted onto the network. This criteria will be met, regardless of whether the transmit frame was the first (or only) frame in the FIFO, or if the transmit frame was queued pending completion of the preceding frame.

If 16 total attempts (initial attempt plus 15 retries) have been made to transmit the frame, the MACE will abandon the transmit process for the particular frame, report a Retry Error (RTRY) in the Transmit Frame Status, and set the XMTINT bit in the Interrupt Register, causing an external EINTR# providing the interrupt is unmasked.

Once the XMTINT condition has been externally recognized, the Transmit Frame Status should be read, which will indicate that the RTRY error occurred. The read operation on the Transmit Frame Status will update the FIFO read and write pointers. If no "End-of-Frame" write (EOF# pin assertion) had occurred during the FIFO write sequence, the entire transmit path will be reset (which will update the transmit FIFO watermark with the current XMTFW value in the FIFO Configuration Control register). If a whole frame does reside in the FIFO, the read pointer will be moved to the start of the next frame or free location in the FIFO, and the write pointer will be unaffected.

After a RTRY error, all further packet transmission will be suspended until the Transmit Frame Status is read, regardless of whether additional packet data exists in the FIFO to be transmitted. Receive FIFO read operations are not impaired.

Packets experiencing 16 unsuccessful attempt to transmit will not be re-tried. Recovery from this condition must be performed by upper layer software.

Abnormal network conditions include :
(a) Loss of carrier.
(b) Late collision.
(c) SQE Test Error.
These should not occur on a correctly configured 802.3 network, but will be reported if the network has been incorrectly configured or a fault condition exists.

(a) A loss of carrier condition will be reported if the MACE cannot observe receive activity whist it is transmitting on the AUI port. After the MACE initiates a transmission it will expect to see data "looped-back" on the DI± pair. This will internally generate a "carrier sense", indicating that the integrity of the data path to and from the MAU is intact, and that the MAU is operating correctly. This "carrier sense" signal must be asserted within TBD bit times after the first transmitted bit on DO± (when using the AUI port). If "carrier sense" does not become active in response to the data transmission, or becomes inactive before the end of transmission, the loss of carrier (LCAR) error bit will be set in the Transmit Frame Status (bit 7) after the packet has been transmitted. The packet will not be re-tried on the basis of an LCAR error.

When the DAI™ port is selected, LCAR errors will not occur, since the MACE will internally loop back the transmit data path to the receiver. The loop back feature must not be performed by the external transceiver.

(b) A late collision will be reported if a collision condition exists or commences 64 byte times (512 bit times) after the transmit process was initiated (first bit of preamble commenced). The MACE will abandon the transmit process for the particular frame, report a Late Collision (LCOL) in the Transmit Frame Status, and set the XMTINT bit in the Interrupt Register, causing an external EINTR# providing the interrupt is unmasked.

Once the XMTINT condition has been externally recognized, the Transmit Frame Status should be read, which will indicate that the LCOL error occurred. The action of reading the Transmit Frame Status will cause the XMTFIFO read and write pointers to be updated. If no "End-of-Frame" write (EOF# pin assertion) had occurred during the FIFO write sequence, the entire transmit path will be reset (which will update the transmit FIFO watermark with the current XMTFW value in the FIFO Configuration Control register). If the whole frame did reside in the FIFO, the read pointer will be moved to the location immediately following the "End-of-Frame" flag, to point to the start of the next frame or free location in the FIFO; the write pointer will be unaffected.

After an LCOL error, all further packet transmission will be suspended until the Transmit Frame Status is read, regardless of whether additional packet data exists in the FIFO to be transmitted. Receive FIFO operations are unaffected.

Packets experiencing a late collision will not be re-tried. Recovery from this condition must be performed by upper layer software.

(c) During the inter packet gap time following the completion of a transmitted message, the AUI CI± pair is asserted by some transceivers as a self-test. When the AUI port has been selected, the integral Manchester Encoder/Decoder will expect the SQE Test Message (nominal 10MHz sequence) to be returned via the CI± pair, within a 20 network bit time period after DI± goes inactive. If the CI± input is not asserted within the 20 network bit time period following the completion of transmission, then the MACE will set the CERR bit (bit 5) in the Interrupt

Register. The EINTR# line will be activated if the corresponding mask bit CERRM = 0. The CERR bit will not be set if the DAI™ has been selected. The external transceiver in this case is not required to support such a test feature.

Host related transmit exception conditions include :
(a) Overerflow caused by excessive writes to the Transmit FIFO (EDTV# will not be issued if the Transmit FIFO is full).
(b) Underflow caused by lack of host writes to the transmit FIFO.
(c) Not reading current Transmit Frame Status.

(a) The host may continue to write to the transmit FIFO after the TDTREQ# has been de-asserted, and can safely do so on the basis of knowledge of the number of free bytes remaining (set by XMTFW in the FIFO Configuration Control register). If however the host system continues to write data to the point that no additional FIFO space exists, the MACE will not return the EDTV# signal and hence will effectively not acknowledge acceptance of the data. It is the host's responsibility to ensure that the data is re-presented at a future time when space exists in the transmit FIFO, and to track the actual data written into the FIFO.

(b) If the host fails to respond to the TDTREQ# from the MACE before the Transmit FIFO is emptied, a FIFO underrun will occur. The MACE will in this case terminate the network transmission in an orderly sequence. If less that 512 bits have been transmitted onto the network the transmission will be terminated immediately, generating a runt packet. If greater than 512 bits have been transmitted, the message will have the current CRC inverted and appended at the next byte boundary, to guarantee an error is detected at the receiving station. The MACE will report this condition to the host by de-asserting the TDTREQ# pin, and setting both the XMTSV bit (in the Transmit Frame Control) and the XMTINT bit (in the Interrupt Register), and will activate the EINTR# pin providing the corresponding XMTINTM bit (in the Interrupt Mask Register) is cleared. It is the host's responsibility to determine if the EINTR# is prematurely set by this condition during the transmit process.

(c) The MACE will internally store the Transmit Frame Status for up to two packets. If the host fails to read the Transmit Frame Status and both internal entries become occupied, the MACE will not commence any subsequent transmit frames to prevent overwriting of the internally stored values. This will occur regardless of the number of bytes written to the transmit FIFO.

### 3.4.2.8 Receive Operation

The receive operation and features of the MACE are controlled by programmable options. These options are programmed through the BIU, FIFO and MAC Configuration Control registers.

Parameters controlled by the MAC Configuration Control register are generally programmed only once, during initialization, and are therefore static during the normal operation of the MACE (see the Media Access Control section for a detailed description). The features controlled by the FIFO Configuration Control register and the Receive Frame Control register can be programmed without performing a reset on the part. The host is responsible for ensuring that no data is present in the receive FIFO when re-programming the receive attributes.

3.4.2.8.1 Receive FIFO Read

The Receive FIFO is accessed by performing a host generated read sequence on the MACE. See section 3.4.2.5.1 on Read Access, and section 4.2 for details of the read access timing.

Note that EOF# will be asserted by the MACE during the last data byte/word transfer.

3.4.2.8.2 Receive Function Programming

Automatic pad field stripping can be programmed using the Receive Frame Control register, to provide flexibility in the reception of messages. ASTRP RCV must be static when the receive function is enabled (ENRCV = 1). The receiver should be disabled before (re-) programming this feature.

The programming of static receive attributes are distributed between the BIU, FIFO and MAC Configuration Control registers.

All receive frames can be accepted by setting the PROM bit (bit 7) in the MAC Configuration Control register. When PROM is set, the MACE will attempt to receive all messages, subject to minimum frame enforcement.

The point at which RDTREQ# is asserted in relation to the number of bytes of a frame that are present in the receive FIFO is controlled by the RCVFW bits in the FIFO Configuration Control register. RDTREQ# will be asserted when one of the following conditions is true :

(i) There are at least 64 bytes in the Receive FIFO.

(ii) The received packet has passed the 64 byte minimum criteria, and the number of bytes in the Receive FIFO is greater than or equal to the threshold set by the RCVFW (16 or 32 bytes).

(iii) A receive packet has completed, and part or all of it is present in the Receive FIFO.

Note that if the RCVFW is set below the 64 byte limit, the MACE will still require 64 bytes of data to be received before the initial assertion of RDTREQ#. Subsequently, RDTREQ# will be asserted at any time the RCVFW threshold is exceeded. The only time that the RDTREQ# will be asserted when there are not at least an initial 64 bytes of data in the FIFO is when either the APAD STRP function has been invoked, and the pad is automatically stripped from a minimum length packet; or when the RPA bit has been set in the User Test Register, and a runt packet has been received.

Depending on the bus latency of the system, RCVFW can be set to ensure that the receive FIFO does not overflow before more data is read from the FIFO. When the entire frame is in the FIFO, RDTREQ# will be asserted regardless of the value in RCVFW. The default value of RCVFW is 64 bytes after ERESET#.

The receive operation of the MACE can be halted at any time by clearing the ENRCV bit (bit 0) in the MAC Configuration Control register. Note that any receive frame currently in progress will be accepted normally, and the MACE will disable the receive process once the message has completed.

3.4.2.8.3 Automatic Pad Stripping

During reception of a frame the pad field can be stripped automatically. ASTRP RCV = 1 enables the automatic pad stripping feature. The pad field will be stripped from receive frames and not passed through the FIFO, thus preserving FIFO space for additional frames. The FCS field will also be stripped, since it is computed at the transmitting station based on the data and pad field characters, and will be invalid for a receive frame that has the pad characters stripped.

The number of bytes to be stripped is calculated from the embedded length field (as defined in the IEEE 802.3 definition) contained in the packet. The length indicates the actual number of LLC data bytes contained in the message.

3.4.2.8.4 Receive FCS Checking

Reception and checking of the received FCS is performed automatically by the MACE. Note that if the Automatic Pad Stripping feature is enabled, the received FCS will be verified against the value computed for the incoming bit stream including pad characters, but it will not be passed through the receive FIFO to the host. If a FCS error is detected, this will be reported by the FCS bit (bit 4) in the Receive Frame Status.

3.4.2.8.5 Receive Status Information

The EOF# indication signals that the last byte/word of data has been passed from the FIFO for the specific frame. This will be accompanied by a RCVINT indication in the the Interrupt Register signalling that the Receive Frame Status has been updated, and must be read. The Receive Frame Status is a single location which must be read 4 times to allow the 4 bytes of status information associated with each frame to be read. Further data read operations from the Receive FIFO using the Register Address mode, will be ignored by the MACE (indicated by the MACE not returning EDTV#) until all 4 bytes of the Receive Frame Status have been read. Alternatively, the FIFO Direct access mode may be used to read the Receive Frame Status through the Receive FIFO. In either case, the 4 byte total must be read before additional receive data can be read from the Receive FIFO. However, the RDTREQ# indication will continue to reflect the state of the receive FIFO as normal, regardless of whether the Receive Frame Status has been read. EDTV# will not be returned when a read operation is performed on the Receive Frame Status location and no valid status is present or ready.

Note that the Receive Frame Status can be read using either the Register Address or FIFO Direct modes. For details, see the section "Receive FIFO Read" for additional details.

3.4.2.8.6 Receive Exception Conditions

Exception conditions for frame reception fall into two distinct categories; those which are the result of normal network operation, and those which occur due to abnormal network and/or host related events.

Normal events which may occur and which are handled autonomously by the MACE are basically collisions within the slot time. The MACE will ensure that collisions which occur within 512 bit times from the start of reception (excluding preamble) will be automatically deleted from the receive FIFO with no host intervention, providing that the Runt Packet Accept (RPA bit in the User Test Register) feature has not been invoked and that the Receive FIFO Watermark (RCVFW bits in the FIFO Configuration Control register) is set at its default value of 64 bytes. The receive FIFO will delete any packet which is subject to a collision before 64 bytes are received. This criteria will be met, regardless of whether the receive frame was the first (or only) frame in the FIFO, or if the receive frame was queued behind a previously received message.

Abnormal network conditions include :
(a) FCS errors.
(b) Framing errors.
(c) Dribbling bits.
(d) Late collision.
These should not occur on a correctly configured 802.3 network, but will be reported if the network has been incorrectly configured or a fault condition exists.

Host related receive exception conditions include :
(a) Underflow caused by excessive reads from the Receive FIFO (EDTV# will not be issued if the receive FIFO is empty).
(b) Overflow caused by lack of host reads from the Receive FIFO.
(c) Missed packets due to lack of host reads from the Receive FIFO and/or the Receive Frame Status.

(a) Successive read operations from the Receive FIFO after the final byte of data/status has been read, will cause the EDTV# pin to remain deasserted during the read operation, indicating that no valid data is present. There will be no adverse effect on the FIFO.

(b) Data present in the Receive FIFO from packets which completed before the overflow condition occurred, can be read out by accessing the Receive FIFO normally. Once this data (and the associated Receive Frame Status) has been read, the EOF# indication will be asserted by the MACE immediately after the first read operation takes place from the Receive FIFO, for the packet which suffered the overflow. If there were no other packets in the FIFO when the overflow occurred, the EOF# will be asserted on the second read from the FIFO. In either case, the EOF# indication will be accompanied by an EINTR# indication from the MACE, providing that the RCVINTM bit in the Interrupt Mask Register is not set. If the Register Address mode is being used, the host is required to access the Receive Frame Status location using 4 separate read cycles. Access to the receive FIFO will be ignored by the MACE until all 4 bytes of the Receive Frame Status have been read. EDTV# will not be returned if a receive FIFO read is attempted. If the FIFO Direct mode is being used, the host can read the Receive Frame Status through the Receive FIFO, but the host must be aware that the subsequent 4 cycles will yield the receive status bytes, and not data from the same or a new packet. Only the OFLO bit will be valid in the Receive Frame Status, other error/status indication and the RCVCNT fields are invalid.

While the receive FIFO is in the overflow condition, it is “deaf” to additional receive data on the network. However, the MACE internal address detect logic continues to operate and counts the number of internal address matches that have been “missed” while the receive FIFO is in this state. The Missed Packet Count (MPC) is an 8 bit count (in register 24) that maintains the number of internal address matches detected. The MPC counter will wrap around when the maximum count of 255 is reached, setting the MPCO (Missed Packet Count Overflow) bit in the Interrupt Register. MPCOM (Missed Packet Count Overflow Mask) in the Interrupt Mask

Register allows the EINTR# event to be masked, and will be set (the interrupt will be masked) after hardware ERESET#.

(c) Failure to read packet data out of the Receive FIFO will eventually cause an overflow condition. The FIFO will maintain any completed packet, which can be read by the host at its convenience, but packet data on the network will not be received, regardless of destination address, until the overflow is cleared by reading the remaining FIFO data and error condition out. The MACE will increment the MPC register to indicate that a message which would have been normally passed to the host was dropped due to the error condition.

### 3.4.2.9 Loopback Operation

During loopback, the FCS logic can be allocated to the receiver by setting RCVFCSE = 1 in the User Test Register. This permits both the transmit and receive FCS operations to be verified during the loopback process. The state of RCVFCSE is only valid during loopback operation.

If RCVFCSE = 0, the MACE will calculate and append the FCS to the transmitted message. The receive message passed to the host will therefore contain an additional 4 bytes of FCS. The Receive Frame Status will indicate the result of the loopback operation and the RCVCNT.

If RCVFCSE = 1, the last four bytes of the transmit message must contain the FCS computed for the transmit date preceding it. The MACE will transmit the data without addition of an FCS field, and the FCS will be calculated and verified at the receiver.

The loopback facilities of the MACE allow full operation to be verified without disturbance to the network. Loopback operation is also affected by the state of the Loopback Control bits (LOOP [0-1]) in the User Test Register. This affects whether the internal MENDEC is considered part of the internal loopback path.

Whe in the loopback mode(s), the multicast address detection feature of the MACE, programmed by the contents of the Logical Address Filter (LADR [0-63]) can only be tested when RCVFCSE = 1, allocating the CRC generator to the receiver.

### 3.4.3 MACE User Accessible Registers

The following registers are provided for operation of the MACE. All registers are 8-bits wide unless otherwise stated. Note that all reserved register bits should be written as zero.

3.4.3.1 Receive FIFO(RCVFIFO) (REG ADDR 0)

| RCVFIFO [15-0] |
|---|

This register provides a 16-bit data path from the Receive FIFO. Reading this register will read one word/byte from the Receive FIFO. The RCVFIFO should only be read when Receive Data Transfer Request (RDTREQ#) is asserted. If the RCVFIFO location is read before 64 bytes are available in the FIFO, EDTV# will not be returned. Once the 64 byte threshold has been achieved and RDTREQ# is asserted, the de-assertion of RDTREQ# does not prevent additional data from being read from the FIFO, but indicates the number of additional bytes which are present, before the FIFO is emptied, and subsequent reads will not return EDTV#(see the section "FIFO Subsystem" for additional details). Write operations to this register will have no effect.

Byte transfers from the receive FIFO are supported, and will be fully aligned to the target memory architecture, defined by the BSWP bit in the BIU Configuration Control register. The Byte Enable inputs ($EBE_{0-1}$#) will define which half of the data bus should be used for the transfer. The external host/controller will be informed that the last byte/word of data in a receive frame is being read from the FIFO, when the MACE asserts the EOF# output.

3.4.3.2 Transmit FIFO (XMTFIFO) (REG ADDR 1)

| XMTFIFO [15-0] |
|---|

This register provides a 16-bit data path to the Transmit FIFO. Byte/word data written to this register will be placed in the Transmit FIFO. The XMTFIFO can be written at any time the Transmit Data Transfer Request (TDTREQ#) is asserted. The de-assertion of TDTREQ# does not prevent data being written to the XMTFIFO, but indicates the number of additional write cycles which can take place, before the FIFO is filled, and subsequent writes will not return EDTV#(see the section "FIFO Subsystem" for additional details). Read operations to this register will have no effect.

Byte transfers to the transmit FIFO are supported, and accept data from the source memory architecture to ensure the correct byte ordering for transmission, defined by the BSWP bit in the MAC Configuration Control register. The Byte Enable inputs ($EBE_{0-1}$#) will define which half of the data bus should be used for the transfer. The use of byte transfers have implications on the latency time provided by the FIFO (see the section "FIFO Subsystem" for additional details). The external host/controller must indicate the last byte/word of data in a transmit frame is being written to the FIFO, using the EOF# input.

3.4.3.3 Transmit Frame Control (XMTFC) (REG ADDR 2)

The Transmit Frame Control register is latched internally on the last write to the transmit FIFO for each individual packet, when EOF# is asserted. This permits automatic transmit padding and FCS generation on a packet-by-packet basis.

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| DRTRY | RES | RES | RES | DXMTFCS | RES | RES | APAD XMT |

| | | |
|---|---|---|
| Bit 7 | DRTRY | Disable Retry. When DRTRY is set, the MACE will provide a single transmission attempt for the packet, all further retries will be suspended. In the case of a collision during the attempt, a Retry Error will be reported in the Transmit Status. With DRTRY cleared, the MACE will attempt up to 15 retries (16 attempts total) before indicating a Retry Error. DRTRY is cleared by hardware ERESET#. DRTRY is not internally latched or protected, and sampled only when EOF# is asserted. |
| Bit 6-4 | RES | Reserved. Written and read as zeroes. |
| Bit 3 | DXMTFCS | Disable Transmit FCS. When DXMTFCS = 0 the transmitter will generate and append an FCS to the transmitted frame. When DXMTFCS = 1, no FCS will be appended to the transmitted frame. The value of DXMTFCS for each frame is programmed when EOF# is asserted to transfer the last byte/word for the transmit packet to the FIFO. DXMTFCS is cleared by hardware ERESET#. DXMTFCS is not internally latched or protected, and sampled only when EOF# is asserted. |
| Bit 2-1 | RES | Reserved. |
| Bit 0 | APAD XMT | Auto Pad Transmit. APAD XMT enables the automatic padding feature. Transmit frames will be padded to extend them to 64 bytes including FCS. The FCS is calculated for the entire frame including pad, and appended after the pad field. APAD XMT is set by hardware ERESET#. APAD XMT is not internally latched or protected, and sampled only when EOF# is asserted. |

3.4.3.4 Transmit Frame Status (XMTFS) (REG ADDR 3)

The Transmit Frame Status is valid after XMTINT is signaled, and providing that XMTSV is set within the register. The register is read only, and is cleared when XMTSV is set and a read operation is performed. Note that if XMTSV is not set, the values in this register can change at any time, including during a read operation.

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| XMTSV | UFLO | LCOL | ONE | MORE | DEFER | LCAR | RTRY |

| | | |
|---|---|---|
| Bit 7 | XMTSV | Transmit Status Valid. Transmit Status Valid indicates that this status is valid for the last frame transmitted. |
| Bit 6 | UFLO | Underflow. Indicates that the transmit FIFO emptied before the end of frame was reached. The transmitted frame is truncated at that point. |
| Bit 5 | LCOL | Late Collision. Indicates that a collision occurred after the slot time of the channel elapsed. The TINT bit will be set in the Interrupt Register, and TDTREQ# will be de-asserted. The MACE does not retry after a late collision. |
| Bit 4 | ONE | One. Indicates that exactly one retry was needed to transmit the frame. |
| Bit 3 | MORE | More. Indicates that more than one retry was needed to transmit the frame. |
| Bit 2 | DEFER | Defer. Indicates that MACE had to defer transmission of the frame. This condition results if the channel is busy when the MACE is ready to transmit. |
| Bit 1 | LCAR | Loss of Carrier. Indicates that the carrier became false during a transmission. The MACE does not retry upon Loss of Carrier. |
| Bit 0 | RTRY | Retry Error. Indicates that a total of 16 unsuccessful attempts were made to transmit the frame, and further attempts have been aborted. |

3.4.3.5 Transmit Retry Count (XMTRC) (REG ADDR 4)

The Transmit Retry Count should be read before the Transmit Frame Status register. Reading the Transmit Frame Status with XMTSV set will cause the XMTRC value to be reset. This register is read only.

| RES | RES | RES | RES | XMTRC [3-0] |
|---|---|---|---|---|

| | | |
|---|---|---|
| Bit 7-4 | RES | Reserved. Read as zeroes. |
| Bit 3-0 | XMTRC [3-0] | Transmit Retry Count. Contains the count of the number of retry attempts made by the MACE to transmit the current transmit packet. The value of the counter will be zero if the first transmission attempt was successful, and a maximum of 15 if all retry attempts were utilized. RTRY will be set in Transmit Frame Status if all 16 attempts were unsuccessful. |

3.4.3.6 Receive Frame Control (RCVFC) (REG ADDR 5)

| RES | RES | RES | RES | RES | M/R# | RES | ASTRPRCV |
|---|---|---|---|---|---|---|---|

| | | |
|---|---|---|
| Bit 7-3 | RES | Reserved. Written and read as zeroes. |
| Bit 2 | M/R# | Match/Reject. This function of this bit is not implemented in the CURIO. |
| Bit 1 | RES | Reserved. Written and read as zeroes. |
| Bit 0 | ASTRP RCV | Auto Strip Receive. ASTRP RCV enables the automatic pad stripping feature. The pad and FCS fields will be stripped from receive frames and not placed in the FIFO. ASTRP RCV is set by activation of the ERESET# pin or the SWRST bit. |

3.4.3.7 Receive Frame Status (RCVFS) (REG ADDR 6)

| RCVFS [31-00] |
|---|

The Receive Frame Status is a single byte location which must be read by 4 read cycles to obtain the 4 bytes (32-bits) of status associated with each receive frame. Receive Frame Status can be read using either the Register Direct or FIFO Direct access modes.

In Register Direct mode, access to the receive FIFO will be denied until all 4 status bytes for the completed frame have been read from the Receive Frame Status location. In FIFO Direct mode, the Receive Frame Status is read through the receive FIFO location, by continuing to execute 4 read cycles after the completion of packet data (and assertion of EOF#). The Receive Frame Status can be read using either mode, or a combination of both modes, however each status byte will be presented only once regardless of access method. Other register reads and/or writes can be interleaved at any time, during the Receive Frame Status sequence.

The Receive Frame Status consists of the following 4 bytes of information:

RFS0 Receive Message Byte Count (RCVCNT) [7-0]
RFS1 Receive Status, Receive Message Byte Count (RCVCNT) [11-8]
RFS2 Runt Packet Count (RNTPC) [7-0]
RFS3 Receive Collision Count (RCVCC) [7-0]

3.4.3.7.1 RFS0 - Receive Message Byte Count (RCVCNT)

| RCVCNT [7:0] |
|---|

| | | |
|---|---|---|
| Bit 7-0 | RCVCNT [7:0] | The Receive Message Byte Count indicates the number of whole bytes in the received message from the network. RCVCNT is 12 bits long, and valid only when there are no errors reported in the receive status. RCVCNT [10:8] correspond to bits 3-0 in RFS1 of the Receive Frame Status.RCVCNT [11-0} will be invalid when OFLO is set. |

3.4.3.7.2 RFS1 - Receive Status (RCVSTS)

| OFLO | CLSN | FRAM | FCS | RCVCNT [10:8] |
|---|---|---|---|---|

| | | |
|---|---|---|
| Bit 7 | OFLO | Overflow flag. Indicates that the receive FIFO over flowed due to the inability of the host/controller to read data fast enough to keep pace with the receive serial bit stream and the latency provided by the receive FIFO itself. OFLO is indicated on the receive frame that caused the overflow condition; complete frames in the FIFO are not affected. |
| Bit 6 | CLSN | Collision Flag. Indicates that the receive operation suffered a collision during reception of the frame. If CLSN is set, it indicates that the receive frame suffered a late collision, since collision detected within the slot time will be automatically deleted in the Receive FIFO. CLSN will not be set if OFLO is set. |

| | | |
|---|---|---|
| Bit 5 | FRAM | Framing Error flag. Indicates that the received frame contained a non-integer multiple of bytes and a FCS error. If there was no FCS error then FRAM will not be set. FRAM is not valid during internal loopback. FRAM will not be set if OFLO is set. |
| Bit 4 | FCS | FCS Error flag. Indicates that there is a FCS error in the frame. The receive FCS is computed and checked normally when ASTRP RCV = 1, but is not passed to the host. FCS will not be set if OFLO is set. |
| Bit 3-0 | RCVCNT [11:8] | The Receive Message Byte Count indicates the number of whole bytes in the received message from the network. RCVCNT is 12 bits long, and valid only when there are no errors reported in the receive status. RCVCNT [7:0] correspond to bits 7-0 in RFS0 of the Receive Frame Status. RCVCNT [11-0} will be invalid when OFLO is set. |

3.4.3.7.3 RFS2 - Runt Packet Count (RNTPC)

| RNTPC [7-0] |
|---|

| | | |
|---|---|---|
| Bit 7-0 | RNTPC [7-0] | The Runt Packet Count indicates the number of runt packets received, addressed to this node, since the last successfully received packet. The value does not roll over after 255 runt packets have been detected, and will remain frozen at the maximum count. |

3.4.3.7.4 RFS3 - Receive Collision Count (RCVCC)

| RCVCC [7-0] |
|---|

| | | |
|---|---|---|
| Bit 7-0 | RCVCC | The Receive Collision Count indicates the number of collisions detected on the network since the last successfully received packet. The value does not roll over after 255 collisions have been detected, and will remain frozen at the maximum count. |

3.4.3.8 FIFO Frame Count (FIFOFC)(REG ADDR 7)

| RCVFC [3-0] | XMTFC [3-0] |
|---|---|

Bit 7-4 RCVFC [3-0] Receive Frame Count. The (read only) count of the frames in the receive FIFO. A frame is counted when the last byte is put in the FIFO. The counter is decremented when the last byte of the frame is read. If the RCVFC reaches its maximum value of 15, additional receive frames will be ignored, and the Missed Packet Count (MPC) register will be incremented for frames which match the internal address(es) of the MACE.

Bit 3-0 XMTFC [3-0] Transmit Frame Count. The (read only) count of the frames in the Transmit FIFO. A frame is counted when the last byte is put in the FIFO. The counter is decremented when XMTSV (in the Transmit Frame Status and Poll Register) is set and the Transmit Frame Status read access is performed.

3.4.3.9 Interrupt Register (IR)(REG ADDR 8)

All status bits are set upon occurrence of an event and cleared when read. The resister is read only. In addition all status bits are cleared on hardware ERESET#. Bit assignments for the register are as follows:

| RES | BABL | CERR | RES | RES | MPCO | RCVINT | XMTINT |
|---|---|---|---|---|---|---|---|

Bit 7 RES Reserved. Read as zeroes.

Bit 6 BABL Babble Error. BABL is the transmitter time-out error. It indicates that the transmitter has been on the channel longer than the time required to send the maximum packet. It will be set after 1519 bytes (or greater) have been transmitted. The MACE will continue to transmit until the current packet transmission is over EINTR# is driven if the corresponding mask bit BABLM = 0.

BABL is READ/CLEAR only, and is set by the MACE and reset when read. Writing has no effect. It is also cleared by asserting ERESET#.

| | | |
|---|---|---|
| Bit 5 | CERR | Collision Error. CERR indicates that the CI± input pair to the MACE failed to activate within 20 network bit times after the transmission ended. The Signal Quality Error Test Message (SQE Test Message) after transmission is a transceiver test feature, and utilized only if the AUI port is in use. No such feature is invoked for the DAI™ port. The EINTR# line will be activated if the corresponding mask bit CERRM = 0.<br><br>CERR is READ/CLEAR only. It is set by the MACE and reset when read. Writing has no effect. It is also cleared by asserting ERESET#. |
| Bit 4-3 | RES | Reserved. Read as zeroes. |
| Bit 2 | MPCO | Missed Packet Count Overflow. Indicates that the Missed Packet Count register rolled over at a value of 255 missed frames. Missed frames are defined as received frames which passed the internal address match criteria but were missed due to a Receive FIFO overflow, the receiver being disabled (ENRCV = 0) or an excessive receive frame count (RCVFC > 15). The EINTR# line will be activated if the corresponding mask bit MPCOM = 0.<br><br>MPCO is READ/CLEAR only. It is set by the MACE and reset when read. Writing has no effect. It is also cleared by asserting ERESET#. |
| Bit 1 | RCVINT | Receive Interrupt. Indicates that the host read the last byte/word of a packet. The Receive Frame Status is available immediately on the next host read operation. The EINTR# line will be activated if the corresponding mask bit RCVINTM = 0.<br><br>RCVINT is READ/CLEAR only. It is set by the MACE and reset when read. Writing has no effect. It is also cleared by asserting ERESET#. |
| Bit 0 | XMTINT | Transmit Interrupt. Indicates that the MACE has completed the transmission of a packet and updated the Transmit Frame Status. The EINTR# line will be activated if the corresponding mask bit XMTINTM = 0.<br><br>XMTINT is READ/CLEAR only. It is set by the MACE and reset when read. Writing has no effect. It is also cleared by asserting ERESET#. |

3.4.3.10 Interrupt Mask Register (IMR) (REG ADDR 9)

This register contains mask bits for the interrupts. Writing a one into a bit will mask the corresponding interrupt. Bit assignments for the register are as follows:

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| RES | BABLM | CERRM | RES | RES | MPCOM | RCVINTM | XMTINTM |

| | | |
|---|---|---|
| Bit 7 | RES | Reserved. Written and read as zeroes. |
| Bit 6 | BABLM | Babble Error Mask. BABLM is the mask for BABL. When set by the host, an EINTR# will not be driven by the MACE regardless of the state of the BABL bit. It is cleared hardware ERESET#. |
| Bit 5 | CERRM | Collision Error Mask. CERRM is the mask for CERR. When set by the host, an EINTR# will not be driven by the MACE regardless of the state of the CERR bit. It is cleared by hardware ERESET#. |
| Bit 4-3 | RES | Reserved. Written and read as zeroes. |
| Bit 2 | MPCOM | Missed Packet Count Overflow Mask. MPCOM is the mask for MPCO (Missed Packet Count Overflow). When set, an EINTR# will not be driven by the MACE regardless of the state of the MPCO bit. It is set (the interrupt will be masked) after hardware ERESET#. |
| Bit 1 | RCVINTM | Receive Interrupt Mask. RCVINTM is the mask for RCVINT. When set by the host, an EINTR# will not be driven by the MACE regardless of the state of the RCVINT bit. It is cleared by hardware ERESET#. |
| Bit 0 | XMTINTM | Transmit Interrupt Mask. XMTINTM is the mask for XMTINT. When set by the host, an EINTR# will not be driven by the MACE regardless of the state of the XMTINT bit. It is cleared by hardware ERESET#. |

3.4.3.11 Poll Register (PR) (REG ADDR 10)

This register contains copies of internal status bits to simplify a host implementation which is non-interrupt driven. The register is read only, and its status is unaffected by read operations. All register bits are cleared by hardware ERESET#. Bit assignments are as follows:

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| XMTSV | TDTREQ | RDTREQ | RES | RES | RES | RES | RES |

| | | |
|---|---|---|
| Bit 7 | XMTSV | Transmit Status Valid. Transmit Status Valid indicates that the Transmit Frame Status is valid, and that a read operation can be performed. |
| Bit 6 | TDTREQ | Transmit Data Transfer Request. An internal indication of the current request status of the transmit FIFO. TDTREQ is set when the external TDTREQ# signal is asserted. |
| Bit 5 | RDTREQ | Receive Data Transfer Request. An internal indication of the current request status of the receive FIFO. RDTREQ is set when the external RDTREQ# signal is asserted. |
| Bit 4-0 | RES | Reserved. Read as zeroes. |

3.4.3.12 BIU Configuration Control (BIUCC) (REG ADDR 11)

All bits within the BIU Configuration Control register will be set to their default state upon a hardware or software reset. Bit assignments are as follows:

| RES | BSWP | XMTSP [1-0] | RES | RES | RES | SWRST |
|---|---|---|---|---|---|---|

| | | |
|---|---|---|
| Bit 7 | RES | Reserved. Written and read as zeroes. |
| Bit 6 | BSWP | Byte Swap. The BSWP function allows data to and from the FIFOs to be orientated according to Intel or Motorola byte ordering conventions. BSWP is cleared by ERESET#, defaulting to Intel byte ordering. |
| Bit 5-4 | XMTSP [1-0] | Transmit Start Point. XMTSP controls the point preamble transmission commences in relation to the number of write cycles performed on the transmit FIFO. When the entire frame is in the FIFO transmission will start regardless of the value in XMTSP. XMTSP is given a value of 10 (64 bytes) after hardware ERESET#. Regardless of XMTSP, the FIFO will not internally over write its data until at least 64 bytes, or the entire frame, has been transmitted onto the network. This ensures that for collisions within the slot time window, transmit data need not be re-written to the transmit FIFO, and re-tries will be handled autonomously by the MACE. |

| XMTSP [1-0] | BYTES |
|---|---|
| 00 | 4 |
| 01 | 16 |
| 10 | 64 |
| 11 | 112 |

Table A.5 : Transmit Start Point

Bit 3-1 RES Reserved. Written and read as zeroes.

Bit 0 SWRST Software Reset. When set, provides an equivalent of the hardware ERESET# function. All register bits will be set to their default values. The MACE will require re-initialization after SWRST has been activated. This bit will be cleared by the MACE after the reset sequence is completed.

3.4.3.13 FIFO Configuration Control (FIFOCC) (REG ADDR 12)

All bits within the FIFO Configuration Control register will be set to their default state upon a hardware reset. Bit assignments are as follows:

| XMTFW [1-0] | RCVFW [1-0] | XMTFWR | RCVFWR | XMTBRST | RCVBRST |
|---|---|---|---|---|---|

Bit 7-6 XMTFW [1-0] Transmit FIFO Watermark. XMTFW controls the point TDTREQ# is asserted in relation to the number of write cycles to the transmit FIFO. TDTREQ# will be asserted at any time that the number of write cycles specified by XMTFW can be executed. XMTFW is set to a value of 00 after hardware ERESET#.

| XMTFW [1-0] | WRITE CYCLES |
|---|---|
| 00 | 8 |
| 01 | 16 |
| 10 | 32 |
| 11 | XX |

Table A.6 : Transmit FIFO Watermarks

Bit 5-4 RCVFW [1-0] Receive FIFO Watermark. RCVFW controls the point RDTREQ# is asserted in relation to the number of full or empty bytes in the receive FIFO. RCVFW specifies the number of bytes which must be present (once the packet has been verified as a non-runt), before the RDTREQ# is asserted. Note however that in order for RDTREQ# to be activated for a new frame, at least 64 bytes must have been received. This effectively avoids reacting to receive frames which are runts or suffer a collision during the slot time (512 bit times). If the Runt Packet Accept feature (RPA in Receive Frame Control) is enabled, the RDTREQ# pin will be activated as soon as either the threshold is reached, or a complete valid receive frame is detected (regardless of length). RCVFW is set to a value of 10 (64 bytes) after hardware ERESET#.

| RCVFW [1-0] | BYTES |
|---|---|
| 00 | 16 |
| 01 | 32 |
| 10 | 64 |
| 11 | XX |

Table A.7 : Receive FIFO Watermarks

Bit 3 XMTFWR Transmit FIFO Watermark Reset. Allows reset of the Transmit FIFO Watermark control bits. The watermark can be written at any point but the new value in the XMTFW bits will be ignored until XMTFWR is set (or the transmit path is reset due to a retry failure). The transmit FIFO should be empty and all transmit activity complete before attempting this since the FIFO will be reset to allow the new pointer values to be loaded. XMTFWR will be reset by the MACE after the new XMTFW value has been loaded.

Bit 2 RCVFWR Receive FIFO Watermark Reset. Allows reset of the Receive FIFO Watermark control bits. The watermark can be written at any point but the new value in the RCVFW bits will be ignored until RCVFWR is set. The receive FIFO should be empty before attempting this since the FIFO will reset to allow the new pointer values to be loaded. RCVFWR will be reset by the MACE after the new RCVFW value has been loaded.

| | | |
|---|---|---|
| Bit 1 | XMTBRST | Transmit Burst. When set, the transmit burst mode is selected. The behavior of the transmit FIFO high watermark, and hence the de-assertion of TDTREQ#, will be modified. TDTREQ# will continue to be asserted when there is sufficient space in the FIFO to allow the specified number of write cycles to occur, as programmed by the XMTFW bits. TDTREQ# will de-asserted when there are only 2-bytes of space available in the transmit FIFO (so that a full word write can still occur) or immediately when only 4-bytes remain and the EOF# pin is asserted by the host. |
| Bit 0 | RCVBRST | Receive Burst. When set, the receive burst mode is selected. The behavior of the receive FIFO low watermark, and hence the de-assertion of RDTREQ#, will be modified. RDTREQ# will continue to be asserted when a minimum of 64-bytes have been received for a new frame (or a runt packet has been received and RPA is set). Once the 64-byte limit has been exceeded, RDTREQ# will be asserted providing there is sufficient data in the FIFO to exceed the threshold, as programmed by the RCVFW bits. RDTREQ# will de-assert when there are only 2-bytes of data available in the receive FIFO.(so that a full word read can still occur). |

### 3.4.3.14 MAC Configuration Control (MACCC) (REG ADDR 13)

This register programs the transmit and receive operation and behavior of the internal MAC engine. All bits within the MAC Configuration Control register are cleared upon a hardware reset. Bit assignments are as follows:

| PROM | DXMT2PD | EMBA | RES | RES | RES | ENXMT | ENRCV |
|---|---|---|---|---|---|---|---|

| | | |
|---|---|---|
| Bit 7 | PROM | Promiscuous. When PROM is set all incoming frames are received regardless of the destination address. PROM is cleared by ERESET#. |
| Bit 6: | DXMT2PD | Disable Transmit Two Part Deferral. When set, disables the transmit two part deferral option. DXMT2PD is cleared by hardware ERESET#. |
| Bit 5: | EMBA | Enable Modified Back-off Algorithm. When set, enables the modified backoff algorithm. EMBA is cleared by hardware ERESET#. |
| Bit 4-2 | RES | Reserved. Written and read as zeroes. |

| | | |
|---|---|---|
| Bit 1 | ENXMT | Enable Transmit. Setting ENXMT = 1 enables transmission. With ENXMT = 0, no transmission will occur. If ENXMT is written as 0 during frame transmission, a packet transmission which is incomplete will have a guaranteed CRC violation appended before the internal transmit FIFO is cleared. No subsequent attempts to load the FIFO should be made until ENXMT is set and TDTREQ# is asserted. ENXMT is cleared by hardware ERESET#. |
| Bit 0 | ENRCV | Enable Receive. Setting ENRCV = 1 enables reception of frames. With ENRCV = 0, no frames will be received from the network into the internal FIFO. When ENRCV is written as 0, any receive frame currently in progress will be completed before the receive FIFO is cleared and the MACE enters the monitoring state for missed packets. ENRCV is cleared by hardware ERESET#. |

3.4.3.15 PLS Configuration Control (PLSCC) (REF ADDR 14)

All bits within the PLS Configuration Control register are cleared upon a hardware ERESET#. Bit assignments are as follows:

| RES | RES | RES | RES | XMTSEL | PORTSEL [1-0] | ENSTS |
|---|---|---|---|---|---|---|

| | | |
|---|---|---|
| Bit 7-4 | RES | Reserved. Written and read as zeroes. |
| Bit 3 | XMTSEL | Transmit Mode Select. XMTSEL provides control over the AUI DO+ and DO- operation while the MACE is not transmitting. With XMTSEL = 0, DO+ and DO- will be equal during transmit idle state, providing zero differential to operate transformer coupled loads. The turn off and return to zero delays are controlled internally. With XMTSEL = 1, DO+ is positive with respect to DO- during the transmit idle state. |
| Bit 2-1 | PORTSEL [1-0] | Port Select. PORTSEL is used to select between the AUI interface and Digital Attachment Interface™ (DAI™). When PORTSEL = 00, the the AUI interface is operational. Setting PORTSEL = 10 activates the DAI™. PORTSEL is cleared by hardware ERESET#. PORTSEL will determine which of the interfaces is operational and tested when utilizing the loopback options (LOOP [1-0]) in the User Test Register. |

| PORTSEL [1 - 0] | ACTIVE INTERFACE | PWRDN# PIN |
|---|---|---|
| 0 0 | AUI | LOW |
| 0 1 | RESERVED | HIGH |
| 1 0 | DAI™ | HIGH |
| 1 1 | GPSI | TBD |

Table A.8 : PORTSEL Interface Definition

Bit 0 ENSTS Enable Status. ENSTS is used to enable the optional I/O functions from the PLS function. The following pins are affected by the ENSTS bit : RXCRS, RXDAT, TXEN, TXDAT+, TXDAT-, CLSN, STDCLK, SRDCLK and SRD. Note that if an external SIA is being utilized via the GPSI, PORTSEL [1-0] = 11 must be programmed before ENSTS is set, to avoid contention of clock, data and/or carrier indicator signals.

3.4.3.16 PHY Configuration Control (PHYCC) (REG ADDR 15)

All bits within the PHY Configuration Control register are reserved for future expansion. Cleared upon activation of the ERESET# pin or SWRST bit.

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| RES | RES | RES | RES | RES | RES | RES | RES |

3.4.3.17 MACE Chip Identification Register (CHIPID [15-00]); (REG ADDR 16 &17)

This 16-bit value corresponds to the specific version of the MACE device being used. The initial value will be programmed to 0941h.

| |
|---|
| CHIPID [07-00] |
| CHIPID [15-08] |

3.4.3.18 Internal Address Configuration (IAC) (REG ADDR 18)

This register allows access to and from the multi-byte Physical Address and Logical Address Filter locations, using only a single byte location.

The MACE will reset the IAC register PHYADDR and LOGADDR bits after the appropriate number of read or write cycles have been executed on the Physical Address Register or the Logical Address Filter. Once the LOGADDR bit is set, the MACE will reset the bit after 8 read or write operations have been performed. Once the PHYADDR bit is set, the MACE will reset the bit after 6 read or write operations have been performed. The MACE makes no distinction between read or write operations, advancing the internal RAM pointer with each access. If both PHYADDR and LOGADDR bits are set, the MACE will accept only the LOGADDR bit. If the PHYADDR bit is set and the Logical Address Filter location is accessed, a EDTV# will not be returned. Similarly, if the LOGADDR bit is set and the Physical Address Register location is accessed, EDTV# will not be returned.

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| RES | RES | RES | RES | RES | PHYADDR | LOGADDR | RES |

Bit 7-3 RES Reserved. Written and read as zeroes.

Bit 1 PHYADDR Physical Address Reset. When set, successive reads or writes to the Physical Address Register will occur in the order PADR [07-00], PADR [15-08],....,PADR [47-40]. Each read or write operation on the PADR location will auto-increment the internal pointer to access the next most significant byte.

Bit 0 LOGADDR Logical Address Reset. When set, successive reads or writes to the Logical Address Filter will occur in the order LADRF [07-00], LADRF [15-08],....,LADRF [63-56]. Each read or write operation on the LADRF location will auto-increment the internal pointer to access the next most significant byte.

Bit 0 RES Reserved. Written and read as zeroes.

3.4.3.19 Logical Address Filter (LADRF [63-00]) (REG ADDR 20)

| |
|---|
| LADRF [63-00] |

This 64-bit mask is used to accept incoming Logical Addresses. The Logical Address Filter is expected to be programmed at initialization (after hardware or software reset), and not altered subsequently.

If the least significant address bit of a received message is set (Destination Address bit 00 = 1), then the address is deemed logical, and passed through the FCS generator. After processing the 48-bit destination address, a 32-bit resultant FCS is produced and strobed into an internal register. The high order 6-bits of this resultant FCS are used to select one of the 64-bit positions in the Logical Address Filter (see diagram). If the selected filter bit is a "1", the address is accepted and the packet will be placed in memory.

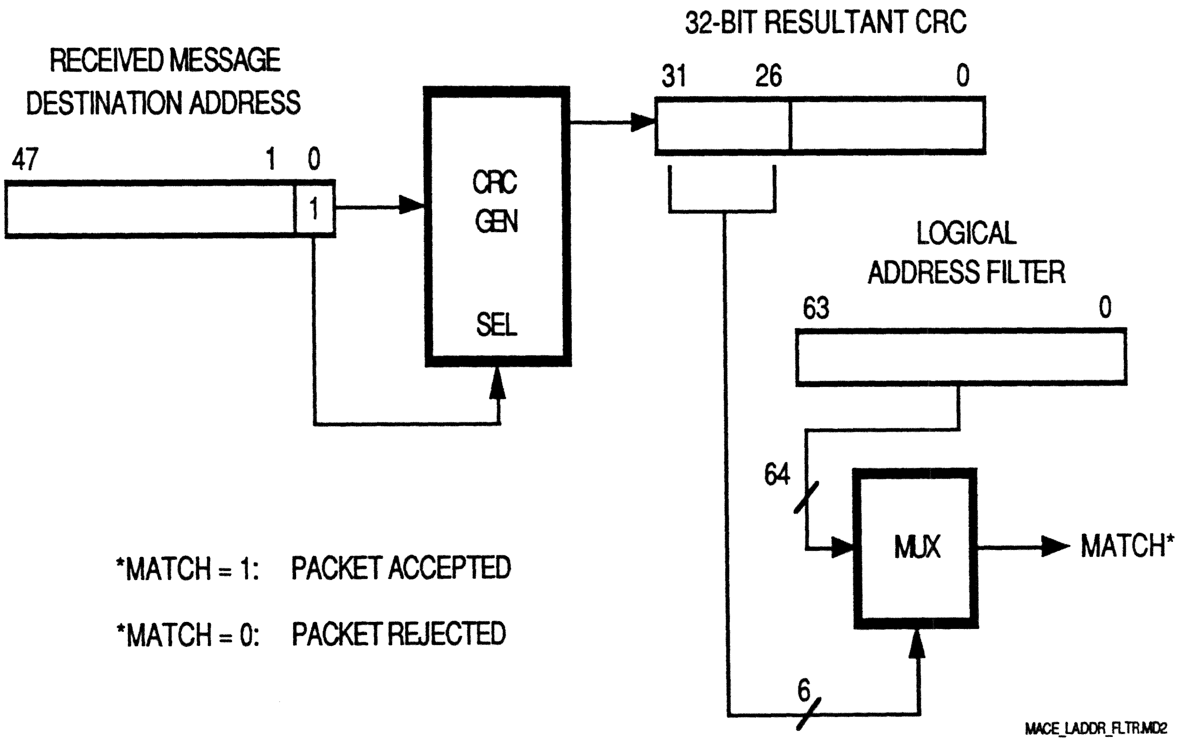



Figure A.9

The first bit of the incoming address must be a “1” for a logical address. If the first bit is a “0,” it is a physical address and is compared against the value stored in the Physical Address Register at initialization.

The Logical Address Filter is used in multicast addressing schemes. The acceptance of the incoming frame based on the filter value indicates that the message may be intended for the node. It is the user's responsibility to determine if the message is actually intended for the node by comparing the destination address of the stored message with a list of acceptable logical addresses.

The Broadcast address, which is all ones, does not go through the Logical Address Filter and is always enabled. If the Logical Address Filter is loaded with all zeroes (and PROM =0), all incoming logical addresses except broadcast will be rejected.

Multicast addressing can only be performed when using external loopback (LOOP [1-0] = 0) by programming RCVFCSE = 1 in the User Test Register. This effectively allocates the FCS logic to the receiver section, and allows the FCS to be computed on the incoming logical address.

3.4.3.20 Physical Address (PADR [47-00]) (REG ADDR 21)

This 48-bit value represents the unique node value assigned by the IEEE and used for internal address comparison.

| PADR [47-00] |
|---|

3.4.3.21 Missed Packet Count (MPC) (REG ADDR 24)

The Missed Packet Count (MPC) is an 8 bit count that maintains the number of address matches detected when the receiver is "deaf", due to one of the following conditions :
( i ) A receive FIFO overflow condition occurs, until the overflow is cleared by reading the Receive Frame Status.
( i i ) The receive function has been disabled by clearing the ENRCV bit in the MAC Configuration Control register.
( i i i ) The Receive Frame Count (RCVFC) in the FIFO Frame Count register exceeds its maximum value, indicating that greater than 15 frames are in the Receive FIFO.

| MPC [7-0] |
|---|

The MPCO (Missed Packet Count Overflow) bit in the Interrupt Register will be set and the EINTR# line asserted if the number of received frames that have been missed exceeds 255. MPCOM (Missed Packet Cont Overflow Mask) in the Interrupt Mask Register allows the EINTR# event to be masked, and will be set (the interrupt will be masked) after hardware ERESET#. The MPC will roll over at a value of 255, and continue from zero.

3.4.3.22 User Test Register (UTR) (REG ADDR 29)

The User Test Register is used to put the chip into test configurations. All bits within the Test Register are cleared upon a hardware or software reset. Bit assignments are as follows:

| RTRE | RTRD | RPA | FCOLL | RCVFCSE | LOOP [1-0] | RES |
|---|---|---|---|---|---|---|

Bit 7 RTRE Reserved Test Register Enable. Access to the Reserved Test Registers should not be attempted by the user. Note that access to the Reserved Test Register may cause damage to the MACE if configured in a system board application. Access to the Reserved Test Register is prevented, regardless of the state of RTRE, once RTRD has been set. RTRE is cleared on hardware ERESET#.

| | | |
|---|---|---|
| Bit 6 | RTRD | Reserved Test Register Disable. When set, access to the Reserved Test Registers is inhibited, and further writes to the RTRD bit are ignored. Access to the Reserved Test Register is prevented, regardless of the state of RTRE, once RTRD has been set. RTRD can only be cleared by hardware ERESET#. |
| Bit 5 | RPA | Runt Packet Accept. Allows receive packets which are less than the legal minimum as specified by IEEE 802.3/Ethernet, to be passed to the host interface via the receive FIFO. Cleared on ERESET#. |
| Bit 4 | FCOLL | Force Collision. Allows the collision logic to be tested. The MACE should be in an internal loopback test for the FCOLL test. When FCOLL = 1, a collision will be forced during the next transmission attempt. This will result in 16 total transmission attempts (if DRTRY = 0) with the RetryError reported in the Transmit Frame Status register. FCOLL is cleared by the activation of the ERESET# pin or SWRST bit. |
| Bit 3 | RCVFCSE | Receive FCS Enable Allows the hardware associated with the FCS generation to be allocated to the transmitter or receiver during loopback diagnostics. When clear, the FCS will be generated and appended to the transmit message (providing that DXMTFCS in the Transmit Frame Control is clear), and received after the loopback process through the receive FIFO. When set, the hardware associated with the FCS generation is allocated to the receiver. A transmit packet will be assumed to contain the FCS in the last 4 bytes of the frame passed through the transmit FIFO. The received frame will have the FCS calculated on the data field and the last 4 bytes will be compared with the computed value. An FCS error will be flagged in the Received Status (RFS1) if the received and calculated values do not match. RCVFCSE is only valid when in any one of the loopback modes as defined by LOOP [0-1]. RCVFCSE is cleared by activation of the ERESET# pin or SWRST bit. |
| Bit 2-1 | LOOP [1-0] | Loopback Control. The loopback functions allow the MACE to receive its own transmitted frames. Three levels of loopback are provided as shown in the following table. During loopback operation a multicast address can only be recognized if RCVFCSE = 1. LOOP [1-0] are cleared by activation of the ERESET# pin or SWRST bit. |

| LOOP [1 - 0] | FUNCTION |
|---|---|
| 0 0 | No Loopback |
| 0 1 | External Loopback |
| 1 0 | Internal Loopback, excludes MENDEC |
| 1 1 | Internal Loopback, includes MENDEC |

Table A.10 Loopback Functions

External loopback permits the MACE to transmit to the physical medium, using either the DAI™ or AUI port, dependent on the PORTSEL [1-0] bits in the PLS Configuration Control register. Using the internal loopback test will ensure that transmission does not disturb the physical medium and will prohibit frame reception from the network. One Internal loopback function includes the MENDEC in the loop.

Bit 0 RES Reserved. Written and read as zeroes.

3.4.3.23 Reserved Test Register 1 (RTR1) (REG ADDR 30)

The Reserved Test Register 1/2 are used to put the chip into test configurations. Some pin functions may change if this register is accessed. Note that access to the Reserved Test Register may cause re-definition of the external hardware attributes of the MACE and may cause irrecoverable damage if configured in a system board application.

| XFTSTA | XFTSTB | XFTSTC | XFTSTD | RSTSTST | RES | RES | SIATST |
|---|---|---|---|---|---|---|---|

| | | |
|---|---|---|
| Bit 7 | XFTSTA | Reserved for AMD internal use only. |
| Bit 6 | XFTSTB | Reserved for AMD internal use only. |
| Bit 5 | XFTSTC | Reserved for AMD internal use only. |
| Bit 4 | XFTSTD | Reserved for AMD internal use only. |
| Bit 3 | RSTSTST | Reserved for AMD internal use only. |
| Bit 2 | RES | Reserved for AMD internal use only. |
| Bit 1 | RES | Reserved for AMD internal use only. |
| Bit 0 | SIATST | Reserved for AMD internal use only. |

3.4.3.24 Reserved Test Register 2 (RTR2)(REG ADDR 31)

The Reserved Test Register 1/2 are used to put the chip into test configurations. Some pin functions may change if this register is accessed. Note that access to the Reserved Test Register may cause re-definition of the external hardware attributes of the MACE and may cause irrecoverable damage if configured in a system board application.

| VODITST | RTYTSTA | RTYTSTB | RTYTSTC | RTYTSTD | RFTSTA | RFTSTB | RFTSTC |
|---|---|---|---|---|---|---|---|

| | | |
|---|---|---|
| Bit 7 | VODITST | Reserved for AMD internal use only. |
| Bit 6 | RTYTSTA | Reserved for AMD internal use only. |
| Bit 5 | RTYTSTB | Reserved for AMD internal use only. |
| Bit 4 | RTYTSTC | Reserved for AMD internal use only. |
| Bit 3 | RTYTSTD | Reserved for AMD internal use only. |
| Bit 2 | RFTSTA | Reserved for AMD internal use only. |
| Bit 1 | RFTSTB | Reserved for AMD internal use only. |
| Bit 0 | RFTSTC | Reserved for AMD internal use only. |

3.4.3.25 Register Table Summary

| ADDRESS | MNEMONIC | CONTENTS | COMMENTS |
|---|---|---|---|
| 0 | RCVFIFO | Receive FIFO [15-00] | read only |
| 1 | XMTFIFO | Transmit FIFO [15-00] | write only |
| 2 | - | Transmit Frame Control | |
| 3 | | Transmit Frame Status | |
| 4 | | Transmit Retry Count | read only |
| 5 | | Receive Frame Control | |
| 6 | RFS0-3 | Receive Frame Status | read only |
| 7 | FFC | FIFO Frame Count | read only |
| 8 | IR | Interrupt Register | |
| 9 | IMR | Interrupt Mask Register | |
| 10 | PR | Poll Register | |
| 11 | BIUCC | BIU Configuration Control | |
| 12 | FIFOCC | FIFO Configuration Control | |
| 13 | MACCC | MAC Configuration Control | |
| 14 | PLSCC | PLS Configuration Control | |
| | PHYCC | PHY Configuration Control | |
| 16 | CHIPID | Chip Identification Register [07-00] | read only |
| 17 | CHIPID | Chip Identification Register [15-08] | read only |
| 18 | IAC | Internal Address Configuration | |
| 19 | | Reserved | |
| 20 | LADRF | Logical Address Filter [63-00] | |
| 21 | PADDR | Physical Address [47-00] | |
| 22 | | Reserved | |
| 23 | | Reserved | |
| 24 | MPC | Missed Packet Count | read only |
| 25 | | Reserved | |
| 26 | | Reserved | |
| 27 | | Reserved | |
| 28 | | Reserved | |
| 29 | UTR | User Test Register | |
| 30 | RTR1 | Reserved Test Register 1 | reserved |
| 31 | RTR2 | Reserved Test Register 2 | reserved |

Table A.11 : Internal Register Addresses

### 3.4.3.26 Register Bit Summary

#### 3.4.3.26.1 16-Bit Registers

| |
|---|
| RCVFIFO [15-0] |
| XMTFIFO [15-0] |

#### 3.4.3.26.2 8-Bit Registers

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| DRTRY | RES | RES | RES | DXMTFCS | RES | RES | APAD XMT |
| XMTSV | UFLO | LCOL | ONE | MORE | DEFER | LCAR | RTRY |
| TDTREQ | RDTREQ | RES | RES | XMTRC [3-0] | | | |
| RES | RES | RES | RES | RES | M/R# | RES | ASTRPRCV |
| RCVFS [31-00] | | | | | | | |
| RCVFC [3-0] | | | | XMTFC [3-0] | | | |
| RES | BABL | CERR | RES | RES | MPCO | RCVINT | XMTINT |
| RES | BABLM | CERRM | RES | RES | MPCOM | RCVINTM | XMTINTM |
| XMTSV | TDTREQ | RDTREQ | RES | RES | RES | RES | RES |
| RES | BSWP | XMTSP [1-0] | | RES | RES | RES | RES |
| XMTFW [1-0] | | RCVFW [1-0] | | XMTFWR | RCVFWR | XMTBRST | RCVBRST |
| PROM | DXMT2PD | EMBA | RES | RES | RES | ENXMT | ENRCV |
| RES | RES | RES | RES | XMTSEL | PORTSEL [1-0] | | RES |
| RES | RES | RES | RES | RES | RES | RES | RES |
| CHIPID [07-00] | | | | | | | |
| CHIPID [15-08] | | | | | | | |
| RES | RES | RES | RES | RES | PHYADDR | LOGADDR | RES |
| RESERVED | | | | | | | |
| LADRF [63-00] | | | | | | | |
| PADR [47-00] | | | | | | | |
| RESERVED | | | | | | | |
| RESERVED | | | | | | | |
| MPC [7-0] | | | | | | | |
| RESERVED | | | | | | | |
| RESERVED | | | | | | | |
| RESERVED | | | | | | | |
| RESERVED | | | | | | | |
| RTRE | RTRD | RPA | FCOLL | RCVFCSE | LOOP [1-0] | | RES |
| XFTSTA | XFTSTB | XFTSTC | XFTSTD | RSTSTST | RES | RES | RES |
| VODITST | RTYTSTA | RTYTSTB | RTYTSTC | RTYTSTD | RFTSTA | RFTSTB | RFTSTC |

3.4.3.26.3 Receive Frame Status

| | | | | |
|---|---|---|---|---|
| RCVCNT [7:0] | | | | |
| OFLO | CLSN | FRAM | FCS | RCVCNT [10:8] |
| RNTPC [7-0] | | | | |
| RCVCC [7-0] | | | | |

## 3.5 SCSI FUNCTIONAL DESCRIPTION

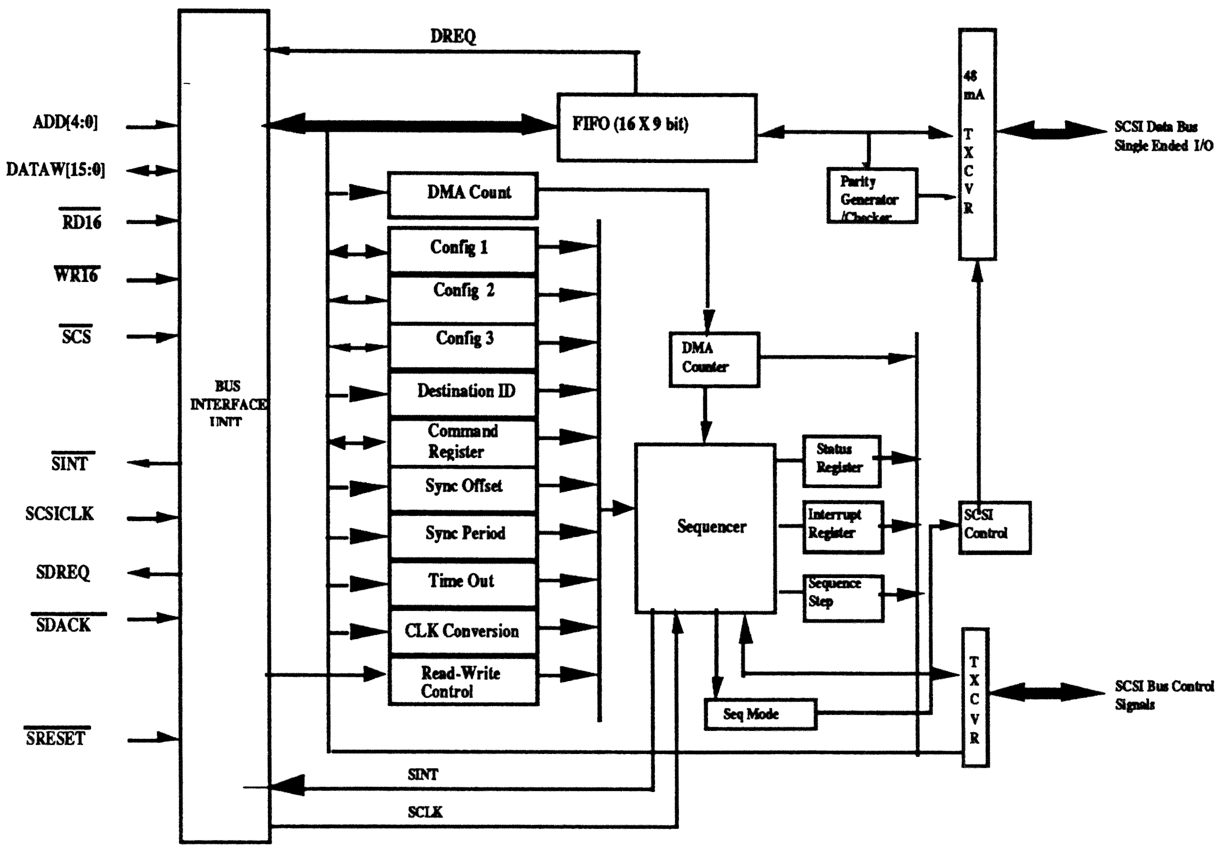



Figure B.1

The SCSI subsystem is designed to minimize host intervention by implementing common SCSI sequences directly in hardware. On-chip state machines reduce protocol overhead by performing these sequences in response to a single command from the host. Selection, Reselection, Information transfer, and Disconnection are directly supported.

The host is further assisted by the internal 16 entry FIFO. The FIFO provides temporary storage for all Command, Data, Status, and Message bytes as they are transferred between the 16-bit Data bus and the SCSI bus.

Both DMA and non-DMA instructions benefit from the FIFO. For DMA instructions the FIFO acts as a buffer to allow greater latency in the DMA channel. This permits the DMA channel to be suspended for higher priority events such as DRAM refresh or reception of a data packet. For non-DMA instructions the FIFO permits several Commands to be queued up.

### 3.5.1 SCSI Bus Sequences

SCSI Selection

When one of the Selection or Reselection commands begins execution, it will clear the Enable Selection or Reselection after arbitration has been granted.The ANSI standard specifies a 250ms period where the SCSI controller can be re-enabled onto the SCSI bus. If this 250ms time-out is exceeded, either an Initiator or a Target trying to connect to the SCSI bus will abort the operation.

If a SCSI bus initiated event occurs prior to the time-out, the SCSI controller will clear the FIFO, clear the Command Register, and ignore any writes from the host until the Interrupt Register is read. The interrupt service routine for a Selection or Reselection command will have to examine the Interrupt Register to determine the selector and the selected. A Function Complete Interrupt indicates that the SCSI controller selected another device. A Selection or Reselection Interrupt indicates that another device selected the SCSI controller.

When the SCSI controller is Selected in the Target role, the FIFO will contain the Bus ID, Identify Message and the Command Descriptor Block.

The Bus ID is a mandatory one byte entry that represents the state of the SCSI bus during the Selection Phase. The Initiator ID must be set in arbitrating systems and is optional in non-arbitrating systems. The Target ID is set for both arbitrating and non-arbitrating systems.

The Identify Message is a mandatory entry that will be one byte for SCSI -1 systems, but may be one or three bytes for SCSI -2 systems. If the SCSI controller is selected with ATN# deasserted this field will contain null (00) byte(s).

If the SCSI -2 bit of the Configuration-2 Register is set, then the controller will examine both the first byte of the Identify Message and the ATN# signal. ATN# being asserted after the first Identify Message byte will cause the controller to request two more bytes. The SCSI controller will then begin requesting the Command Descriptor Block unless the first byte of the Identify Message is invalid, a parity error is detected, ATN# is deasserted between the second and third bytes, or ATN# remains asserted and the SCSI-2 bit is cleared. Any of these events will cause the controller to stop and generate an interrupt.

The Command Descriptor Block may be six, ten or 12 bytes long. The CDB always begins at the third or fifth byte in the FIFO. In SCSI-2 it is possible to fill the entire FIFO with a tagged queue after a twelve byte command is issued.

SCSI Reselection

After receiving the Enable Selection or Reselection command, the SCSI controller can be Reselected in an Initiator role by a Target. If the sequence completes without exceptions the FIFO will contain the Bus ID and the Identify Message.

Both fields are mandatory one byte fields. The Bus ID is identical to the Selection case.

SCSI Reset

The SCSI RST cannot be disabled by the controller. When reset from the SCSI bus the controller will release the SCSI bus, and reset its internal sequencer. If the SRD bit of Configuration-1 Register is cleared the controller will generate an interrupt to indicate a SCSI Reset Detected.

### 3.5.2 Host Command Sequences

Selection Phase

Once the SCSI controller receives an Enable Selection or Reselection command it can be Selected or Reselected by another Initiator or Target.

In the Initiator role the host will first load the FIFO with the Command Descriptor Block and either one or three optional Message bytes. The host will then issue one of the Select commands and then be available for other tasks. The SCSI controller will arbitrate for the SCSI bus. Once access to the bus has been granted, the controller will independently transfer the Message byte(s), followed by the Command Descriptor Block. After the transfers have completed the SCSI controller will generate an Interrupt.

In the target role the host will issue an Enable Selection and then be available for other tasks. Once an Initiator selects a Target, the SCSI controller will independently process the Selection and Command Phases prior to generating an interrupt. When the host is interrupted, the entire Command Descriptor Block and any of the optional Message bytes sent by the Initiator will be in the FIFO.

Information Phase

Following the successful completion of the Selection Phase, the SCSI controller may transfer bytes in any of the Information Phases regardless of whether the controller is functioning in an Initiator or Target role. The SCSI controller supports Disconnect and Reselect in both roles, thereby easing the difficulty of implementing multi-threaded systems.

Data Phase

The SCSI controller is capable of synchronous SCSI bus transfers upto 5 MBytes / sec, and asynchronous SCSI bus transfers upto 5 MBytes / sec. The transfer mode is transparent to the user with the exception that synchronous transfers require the Synchronous Offset Register and the Synchronous Transfer Period Register **TO BE INITIALIZED**. After a hardware or software reset the transfer mode defaults to asynchronous.

During the Data Phase, bytes are normally transferred using DMA. The SCSI controller is designed to support operation with an external DMA controller (such as the Am9517). The host need only initialize the DMA Count, issue a DMA transfer instruction, and then wait for an interrupt. To improve system bus performance the FIFO supports an optional 8 byte burst mode transfer and 8 byte threshold.

Disconnect Phase

The host may instruct the Target to complete the SCSI transaction and release the SCSI bus. Once one of the Disconnect commands is received, the Target places the Status byte and a Message byte in the FIFO. The SCSI controller will first enter the Status Phase and transfer the Status byte. Next Message In Phase will be asserted and the second byte will be sent. After the Initiator releases ACK#, the controller will release the SCSI bus.

The Initiator and Target SCSI transaction ending operations are similar. The Initiator will keep the Target from immediately disconnecting by holding ACK# asserted. This permits the host time to examine the Status and Message bytes transmitted from the Target. If both the Status byte and the Message byte indicate a successful transaction the host will send the Message Accepted command to the Initiator. The Initiator will then deassert ACK#, causing the Target to release the SCSI bus. The Initiator will then interrupt the host. If the host detects a problem with either the Status or Message byte the Set ATN command should be issued to the Initiator prior to the Message Accepted command. This will cause the SCSI controller to assert ATN# before releasing ACK#. This will result in the Target requesting Message Out Phase and not releasing the bus.

### 3.5.3 Parity Detection and Generation

The Config-1 Register contains the enable bit for optional parity detection and generation on the SCSI bus, along with a bit to force parity errors. Parity, whether generated internally or passed directly from the pins, is always stored with the byte in the FIFO. The Table below shows how these bits effect parity.

Data Direction: SCSI to FIFO

| | |
|---|---|
| Config-1, bit 4:<br>(Parity Checking bit) | If set, SDP is loaded into FIFO and Parity checking and Error reporting is enabled.<br>If reset, Parity generated output bit is loaded into FIFO and Parity checking and error reporting is disabled. |

Data Direction: FIFO to SCSI

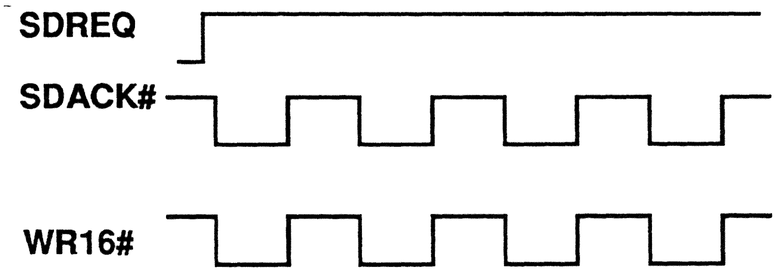
| | |
|---|---|
| Config-1, bit 5: | If set, SDP is a replica of SDIO[7].<br>If reset, FIFO parity bit is loaded onto SDP. |

Parity errors detected by the Target will cause the Parity Error Status bit to be set and the Command Register to be cleared. Parity errors detected by the Initiator will cause the Parity Error bit to be set and ATN# to be asserted prior to releasing ACK#. Parity errors detected a few bytes after a phase change to Synchronous Data In are handled slightly differently in the Initiator role. This is explained further in the section on Initiator Commands.

The Parity Test Mode bit allows the controller to force parity errors by making SDP a replica of SDIO[7].

### 3.5.4 FIFO Threshold

A DMA request is made whenever the number of bytes in the FIFO equals the threshold level. If the Threshold -8 bit in the Config -3 Register is set the threshold will be eight bytes. The default threshold level is two bytes. During a DMA read SDREQ is asserted whenever the FIFO contains at least the threshold number of bytes. SDREQ remains active as long as the number of bytes in the FIFO does not drop below the threshold level.

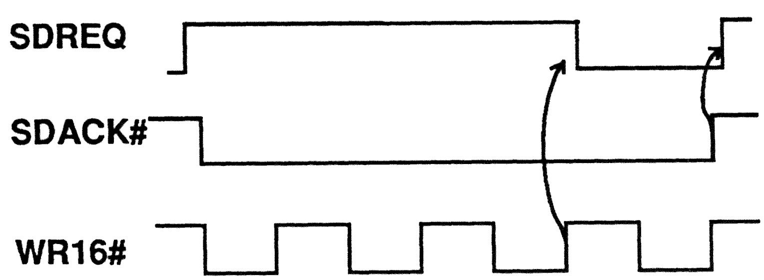



Normal DMA mode

Figure B.2

### 3.5.5 Burst Mode DMA

Burst Mode permits more effecient use of the system bus by transfering a greater number of words per bus arbitration cycle. Burst Mode transfers are selected by enabling the Threshold-8 and Alternate DMA Mode bits in the Config-3 Register. With Threshold -8 enabled the SCSI controller will not assert the DMA request until eight bytes (four words) can be transferred. The Alternate DMA Mode is designed to support the timing of an external DMA controller (such as the Am9517). Figure B.3 shows SDREQ is deasserted after three word transfers causing the external DMA controller to release the system bus after four transfer cycles.




Burst Mode DMA

Figure B.3

### 3.5.6 SCSI Bus Throughput

The SCSI controller is designed to directly drive the SCSI bus without external transceivers. The following discussion on Data rates assumes this configuration:

Synchronous data transmission rate is set by the value in the Synchronous Period Register and is equal to the SCSICLK input frequency divided by its encoded value. This device has been designed to sustain a 5MByte per second synchronous transfer data rate.

The speed of asynchronous data transfer is determined by the SCSI cable length and the SCSICLK frequency. A sustained 5 MBytes per second transfer is acheiveable on a one foot cable under normal voltage and temperature conditions. This rate will be reduced to 4 MByte per second with two typical SCSI devices on a 20 feet long cable. The worst case asynchronous transfer rate, over voltage, temperature, and process deviations is 3 MB/s on a maximum length (single-ended) cable and 4 MB/s on a one foot cable.

The asynchronous transmission data rate is somewhat influenced by SCSICLK when sending data. The SCSI controller drives the bus for a minimum of one SCSICLK period (plus any additional time required to meet the ANSI required 55 ns setup time) before asserting REQ# or ACK#. The SCSICLK frequency does not affect the asynchronous transfer rate when receiving data.

### 3.5.7 Data Alignment

The SCSI controller contains no hardware to align data to uniform boundaries (i.e. no read-modify-write operations). Therefore, when data words are not aligned on word boundaries, host intervention is required to ensure proper handling of the misaligned data.

A word that is written to an odd address will partially occupy two words in the memory. The high byte is stored in the lower half of the system word address, while the low byte is stored in the upper half of the next system address.

To provide aligned data to the SCSI controller, the host must move the first byte after which the DMA controller can move the rest. How the host handles the first byte depends on whether the external processor is performing a read or write and whether the SCSI bus is operating synchronously or asynchronously.

When data is being transmitted to the SCSI bus from the FIFO, the host may preload the FIFO by placing the first byte in the FIFO before issuing the DMA command (any of the Initiator or Target transfer commands).

During asynchronous data transfer from the SCSI bus, the host must read the first byte from the FIFO and store it into the odd memory address. Normal DMA transfers can now occur since the individual bytes can be paired into words with even addresses. When the Save Residual Byte (bit 2 in the Config-3 Register) is enabled SDREQ will not be asserted for the last byte at the end of a DMA transfer. So the host has to read the last byte.

When Target data is flowing synchronously from the SCSI bus into the FIFO (Target Synchronous Data Out Phase), the host processor must first load the FIFO with the lower byte of the destination word. The host must read this byte from memory, write it to the FIFO, then issue the DMA Receive Data command. When DMA writes this 16-bit word (one byte of data padded with one byte from the memory destination) from the FIFO to memory, the low byte will be overwritten with a copy of itself. The high byte of this first word will be the first byte received from the SCSI bus. All subsequent bytes will now be word-aligned.

For the Initiator Synchronous Data In case (data flowing into FIFO from the SCSI bus) the Reserve FIFO Byte option (bit 7 in the Config-2 Register) must be used, premitting the host to preload a location at the bottom of the FIFO (Register OF).

The Reserve FIFO Byte bit must be enabled prior to changing phase to Synchronous Data In. When interrupt arrives, the processor must copy the low byte of the word from its own memory to the bottom of the FIFO register. The DMA Transfer Info command can then be issued. The first 16-bit DMA word written to memory from the FIFO will then be over writing the low byte with a copy of itself; while the first byte received over the SCSI bus will be the high byte of this word. All subsequent bytes will now be word-aligned.

## INTERNAL PROGRAMMABLE REGISTERS

| Address (hex) | READ | WRITE |
|---|---|---|
| 0 | DMA Counter LSB | DMA Count LSB |
| 1 | DMA Counter MSB | DMA Count MSB |
| 2 | FIFO | FIFO |
| 3 | Command | Command |
| 4 | Status | Destination Bus ID |
| 5 | Interrupt | Select/Reselect Timeout |
| 6 | Sequence step | Synchronous Period |
| 7 | FIFO Flags/ Sequence Step | Synchronous Offset |
| 8 | Configuration 1 Register | Configuration 1 Register |
| 9 | Reserved | Clock Conversion Factor |
| A | Reserved | Test Mode |
| B | Configuration 2 Register | Configuration 2 Register |
| C | Vendor ID/Rev / Configuration 3 | Configuration 3 Register |
| D | Reserved | Reserved |
| E | Reserved | Reserved |
| F | Reserved | Reserve FIFO Byte (Config-2) |

Table B.4

### 3.5.8 Register Descriptions

The SCSI controller has distinct internal read and write address spaces. Hence, in addition to the Address pins ADD[4:0], RD16# and WR16# are also used to address a register. SCS# must be asserted to access the register set. However, the FIFO may be accessed with either SCS# or SDACK# together with RD16# or WR16#. When SDACK# is asserted Address pins ADD[4:0] are ignored.

### 3.5.8.1 DMA Counter Write Address 0,1

| |
|---|
| DMA Count LSB |

| |
|---|
| DMA Count MSB |

These two registers implement a 16-bit DMA Count for DMA operations. This count value specifies the number of bytes to be transferred over the SCSI bus. The value of the DMA Count is automatically loaded into the Byte Count by any DMA command. The original contents of these registers are not affected during the transfer. Unlike the Byte Counter which is decremented after each transfer. This enables successive blocks of equal size to be transferred without reprogramming the count. The registers may be reprogrammed any time. Regardless of the state of the previous DMA operation. The maximum count is 65536, and is programmed by writing a zero to both registers. These registers are uneffected by reset and are undefined after power-up.

### 3.5.8.2 DMA Counter Read Address 0,1

| |
|---|
| DMA Counter LSB |

| |
|---|
| DMA Counter MSB |

A read of these two registers returns the value currently in the DMA Counter. The DMA Counter is used by the DMA commands to determine the length of the transfers. All the DMA commands copy the contents of the DMA Count Registers into the DMA Counter. The counter can be loaded at any time by a DMA NOP (80 hex). The non-DMA NOP (00) will not update the counter.

The DMA Counter auto-decrements by one for byte transfers and by two for word transfers. The decrement is performed under Initiator Mode by the leading edges of SDACK# during Synchronous Data In and Data Out phases, and by ACK# during the Asynchronous Data In phase. Under the Target Mode, decrement is performed on the leading edge of SDACK# during Data In phase and by REQ# during Data Out phase.

Care should be taken to insure that false SDACK's are not produced. SDACK# can cause the counter to decrement even if RD16# or WR16# are not asserted. The SDREQ Hi-Z bit in the Config -2 Register should be use to place SDACK# into a high impedence state if false SDACK# could corrupt the counters during a suspended DMA operation.

Two non-DMA commands use the counter - Bus Initiated Selection and Target Receive Command Sequence. When these two commands are decoded by the SCSI controller, the Counter is loaded with the number of bytes in the Command Descriptor Block and subsequently decremented once for every byte received. Other non-DMA commands do not use this counter.

### 3.5.8.3 FIFO Register Read/Write Address 02

| FIFO |
|---|

A 16x9-bit bi-directional FIFO is placed in between the SCSI Bus and the Data Bus. Data from the SCSI bus may optionally be transferred in 8 or 9-bit bytes to the FIFO, depending on the parity control bit setting. Similarly, the Data Bus may also transfer 8 or 9-bit bytes to the FIFO under the control of SCS#, RD16# or WR16#, and the address bits. The FIFO and bus interfaces are designed to support an external DMA controller that may transfer 16-bit words.

After a reset or Bus Initiated Selection or ReSelection the bottom FIFO element and the FIFO Flags are initialized to zeros. The contents of the remainder of the FIFO are not changed by reset. When the FIFO flags are zeros, subsequent successive FIFO reads will access the bottom register.

### 3.5.8.4 **Command Register** Read/Write Address 03

The Command Register is a two byte read/write FIFO that transfers instructions from the host to the SCSI controller. This FIFO allows a second command to be written before the previous command completes. Reset Chip, Reset SCSI Bus and Target Stop DMA execute as soon as they are written into the controller. The remainder of the commands must wait for the previous command to complete before they can begin execution. Reading the Command Register will yield the most recently executed or executing command. The Command Register will be cleared by any of the following conditions:

1. Hardware, Software, SCSI bus RESETs.
2. Bus-initiated Selection or ReSelection, Select or ReSelect Time-out.
3. Assertion of ATN# in Target Mode, Reconnect Command if ATN# is set.
4. Target Terminate Command, Parity Error detected in Target Mode.
5. Select Command, SCSI bus Disconnect Command.
6. Any phase change in Initiator Mode.
7. Illegal Commands.

**Command Register**

| EN DMA | Command Code | | | | | | |
|---|---|---|---|---|---|---|---|
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

Since two commands can be present in the Command Register, two interrupts could result. In the event the first interrupt is not serviced before the second completes, the second interrupt is stacked behind the first. When the Interrupt Register is read to service the first interrupt the contents of the Status Register, Sequence Step Register, and Interrupt Register are updated to describe the second interrupt. When using stacked commands, the Phase Latch bit (bit-6, Config-2 Register) must be enabled.

Bit 7, EN DMA: Enable DMA

If bit 7 is set the instruction is a DMA instruction. DMA instructions automatically cause the contents of the Transfer Count Registers to be loaded into the DMA Counter (a 16-bit DMA Counter). Non-DMA instruction will not load the DMA Counter.

Bits 6 - 0: Command Code

Table B.16 shows the instruction set for the SCSI controller. The Mode Group is determined by bits 4, 5 and 6. The Miscellaneous Group Commands may be issued during any mode (except Target Abort DMA). The Disconnected State Group, Target State Group or Initiator State Group Commands will be accepted only if the SCSI controller is in the same command mode when it is inserted into the Command FIFO. Otherwise, an Illegal Command Interrupt will be generated.

### 3.5.8.5 Status Register Read Address 04

The Status register provides access to two group of event flags. Bits 7 - 3 of the Status Register are latched until the Interrupt Register is read. Once read, these bits will be updated to reflect any stacked interrupts. Bits 2..0 (the SCSI Phase bits) are not normally latched. However, they may be latched by setting bit-6 in the Config -2 Register. Thereby, permitting commands to be stacked.

**Status Register**

| INT | GE | PE | TC | VGC | MSG | C/D# | I/O# |
|---|---|---|---|---|---|---|---|
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

Bit 7, INT: Interrupt

The Interrupt bit is set when the SCSI SINT# pin is asserted. This bit is buffered from the actual SINT# output signal, so that in an external wired-OR configuration the bit reflects the actual status of the SINT# signal. A hardware or software reset, or the reading of the Interrupt Register will deassert an active SINT# signal and clears this bit.

Bit 6, GE: Gross Error

The following conditions will constitute a Gross Error and set the GE bit:

I) The FIFO has overflowed.
2) The Command Register FIFO has overflowed.
3) The DMA transfer is in the opposite direction of SCSI transfer
4) A unexpected phase change occured in the Initiator role during the synchronous Data Phase

A Gross Error does not generate an interrupt. So the presence of a Gross Error can only be identified while servicing another event that resulted in an interrupt. This bit will be cleared by reading the Interrupt Register during an active interrupt or by a SCSI core chip reset.

Bit 5, PE: Parity Error

This bit is set if a parity error occurs on either the Data Bus or the SCSI Bus and parity checking is enabled in the Config -1 Register. An interrupt is not generated when a parity error is detected. Instead the error is recorded in the Status Register. If a parity error is discovered during the Initiator Information In Phase ATN will be asserted. This bit will be cleared by reading the Interrupt Register during an active interrupt or by a SCSI core chip reset.

Bit 4, TC: Terminal Count

When set this bit indicates that the DMA Counter has decremented to zero. The bit is cleared whenever the DMA Count is loaded by a DMA operation, or by a chip reset. Reading the Interrupt register does not effect this bit.

Bit 3, VGC: Valid Group Code

Once Selected the SCSI controller decodes the Group Code field in the first byte of the Command Descriptor Block. This bit will be set if the Group Code is valid (according to ANSI X3.131-1986). This bit will be cleared if an undefined or invalid Group Code is detected. In SCSI-2 mode (SCSI-2 bit is set in the Config-2 register), the controller will detect Group 2 commands as ten byte commands and the bit will be set. In SCSI-1 mode the bit will be cleared, and Group 2 commands will be treated as reserved commands. Group 3 and 4 commands are always treated as reserved commands. The SCSI controller will request six-bytes in response to a reserved group command. The SCSI controller treats Group 6 commands as six-byte vendor unique commands and Group 7 as ten-byte vendor unique commands. The Valid Group Code bit is cleared by reading the Interrupt Register during an interrupt or a reset.

Bits 2-0, MSG, C/D#, I/O#: Phase Bits

These bits indicate the current phase of the SCSI bus. Bit 6 of the Config-2 Register determines if these bits are latched or not. These bits are stable irrespective of the state of bit 6 of the Config-2 Register due to the ANSI definition of the Phase Signals.

| | BITS | | SCSI PHASE |
|---|---|---|---|
| MSG | C/D# | I/O# | |
| 0 | 0 | 0 | Data out |
| 0 | 0 | 1 | Data in |
| 0 | 1 | 0 | Command |
| 0 | 1 | 1 | Status |
| 1 | 0 | 0 | ANSI Reserved |
| 1 | 0 | 1 | ANSI Reserved |
| 1 | 1 | 0 | Message Out |
| 1 | 1 | 1 | Message In |

Latching these bits will permit stacking of commands. When enabled the SCSI Phase is latched upon command completion. The latch is reopened whenever the Interrupt Register is read.

### 3.5.8.6 Destination ID    Write Address 04

| Destination ID |
|---|

Bits2..0 of this register specify the binary encoded destination bus ID for a Selection or ReSelection command. Device ID 7 is expressed as 111 which appears as 80 hex on the SCSI bus. Bits 7..3 are reserved. The contents of the Destination ID is not affected by a reset and is undefined on power-up.

### 3.5.8.7 **Interrupt Register** Read Address 05

| SRD | ICMD | DIS | BS | FC | ReSEL | SELATN | SEL |
|---|---|---|---|---|---|---|---|
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

This register is needed along with the Status Register and Sequence Step Register to determine the cause of an interrupt. Reading this register while the SCSI controller is asserting SINT will clear all three registers. The Interrupt Register will be cleared after a reset.

Bit 7, SRD: SCSI Reset Detected

This bit will indicate a reset is present on the SCSI bus. The bit is set if the SCSI Reset Reporting bit in the Config-1 Register is not enabled and the reset is detected on the SCSI bus.

Bit 6, ICMD: illegal Command

This bit is set if an unused or reserved code is placed in the Command Register or if the command is from a mode group different from the current phase of the SCSI controller.

Bit 5, DIS: Disconnect

When set this bit indicates the SCSI controller has released the SCSI bus. Under the Initiator mode, the SCSI bus is released when the Target disconnects or a Selection or ReSelection Time-out occurs. In the Target mode, this occurs when a Terminate Sequence, or Command Complete Sequence causes the SCSI controller to disconnect from the bus.

Bit 4, BS: Bus Service

This bit indicates that another device is requesting service. This may occur under the Initiator mode when the Target is requesting an Information Transfer Phase and Under the Target mode when the Initiator asserts Attention.

Bit 3, FC: Function Complete

This bit is set after the completion of a Target mode command. In Initiator mode, the bit is set after a Target Selection, after a Command Complete, or after a Transfer Info command when the Target is requesting Message In Phase.

Bit 2, ReSEL: ReSelected

This bit is set to indicate the SCSI controller has been Reselected as an Initiator.

Bit 1, SELATN: Selected with ATN

This bit is set during the Selection Phase once the SCSI controller has been Selected in the Target role and ATN# is asserted.

Bit 0, SEL: Selected

This bit is set during the Selection Phase once the SCSI controller is Selected in a Target role and that ATN# was false during Selection.

### 3.5.8.8 SELECT/ReSELECT Time-Out Write Address 05

| Select/Reselect Timeout |
|---|

During Selection or ReSelection this 8-bit register is used to determine the maximum time to wait for a response. The Time-Out Register is normally loaded to support a 250 milliseconds time-out to comply with the ANSI standard. The Time Out Value (TOV) is calculated from the following:

$$TOV = \frac{(SCSICLK\ Frequency)*(Time\text{-}out\ Period)}{8192 * (Clock\ Conversion\ Factor)}$$

Under a typical condition when SCSICLK is running at 25 MHz, the TOV that gives a 250 ms time-out period is 153 decimal or 99 hexadecimal. The Clock Conversion Factor is defined in the description of Write Address 9. The Timeout Register is not effected by reset, and the state of the register is undefined at power-up.

### 3.5.8.9 Sequence Step Read Address 06

| Reserved | Reserved | Reserved | Reserved | SOM# | Seq Step 2 | Seq Step 1 | Seq Step 0 |
|---|---|---|---|---|---|---|---|
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

Bits 2 - 0 of this register are used to indicate how far the internal sequencer was able to proceed in executing stacked commands. Information from this register can be combined with the Interrrupt Register and Status Register to determine the cause of an interrupt.

Bit[7:4]: Reserved.

Read and Written as zeros.

Bit 3, SOM#: Synchronous Offset Max

This bit is asserted when the synchronous offset counter reaches its maximum value.

Bit[2:0], Seq Step: Sequence Step

The Sequence Step counter is initialized by certain commands and is used to trace progress through the internal algorithm. The possible states are described in section 3.5.9 on Command execution.

### 3.5.8.10 Synchronous Period Write address 06

| reserved | reserved | reserved | ST[4] | ST[3] | ST[2] | ST[1] | ST[0] |
|---|---|---|---|---|---|---|---|
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

Bits 7 - 5, reserved.

These bits are reserved and shall be read and written as zeros.

Bits 4 - 0, ST[4:0]: Synchronous Transfer Period

These bits specify the number of clock periods between successive rising edges of REQ# or ACK# pulses. Data will be transmitted or received synchronously at the rate of one byte every N clocks (SCSICLK) where N is specified in binary value stored in the lower five bits of this register.

| ST[4:0] | Clocks / Byte |
|---|---|
| 0 0 1 0 0 | 5 |
| 0 0 1 0 1 | 5 |
| 0 0 1 1 0 | 6 |
| • • • | • • |
| • • • | • • |
| 1 1 1 1 1 | 31 |
| 0 0 0 0 0 | 32 |
| 0 0 0 0 1 | 33 |
| 0 0 0 1 0 | 34 |
| 0 0 0 1 1 | 35 |

The default value of ST[4:0] is 5 Clocks/Byte after a hardware SRESET#, or a software Reset SCSI Core.

### 3.5.8.11 FIFO Flags / Sequence Step Read address 07

| SS[2] | SS[1] | SS[0] | FF[4] | FF[3] | FF[2] | FF[1] | FF[0] |
|---|---|---|---|---|---|---|---|
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

Bits 7 - 5, SS[2:0]: Sequence Step

These 3 bits are duplicates of the Sequence Step Register bits in normal mode. Bit SS[0] is set under the Test Mode to indicate that the Synchronous Offset counter is not zero. (A non-zero value in the Synchronous Offset counter means data may continued to be transferred, while a zero value means the synchronous offset count has expired and the SCSI controller will not transfer any more data until it receives an acknowledge.)

Bits 4 - 0, FF[4:0]: FIFO Flags

These five bits indicate in binary the number of bytes currently in the FIFO. This flag will not be stable while the SCSI subsection is transferring data and should not be polled until such transfers are completed.

### 3.5.8.12 Synchronous Offset Write address 07

| reserved | reserved | reserved | reserved | SO[3] | SO[2] | SO[1] | SO[0] |
|---|---|---|---|---|---|---|---|
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

Bits 7 - 4, reserved.

These bits are reserved and shall be read and written as zeros.

Bits 3 - 0, SO[3:0]: Synchronous Offset.

These bits should be specified as zero for asynchronous data transfers. A non-zero value specifies the Synchronous Offset during synchronous data transfers.

The Synchronous Offset is the number of Data Phase bytes that may be sent synchronously without either a REQ# or ACK#, depending on whether the device is in Initiator or Target mode. During transmission, the device will stop sending data bytes when this offset is reached and thereafter sends one byte for every ACK# received from other SCSI devices on the bus. When receiving data from the SCSI bus, an ACK# will be sent every time a byte is removed from its FIFO on the system bus interface. The maximum offset of 15 allows the device to store data in its FIFO, while the external DMA controller gains control of the system memory bus.

The Synchronous Offset is cleared by either a hardware SRESET# or a software Reset SCSI Core, but is unaffected by a SCSI reset.

### 3.5.8.13 Configuration-1 Register Read / Write address 08

| Slow | SRD | Parity_T | En_P_Ck | Chip Test | BID[2] | BID[1] | BID[0] |
|---|---|---|---|---|---|---|---|
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

This 8-bit read / write register specifies the various conditions for the SCSI subsection to operate. The bit pattern of this register is not altered during operation.

Bit 7, Slow: Slow Cable Mode.

Setting this bit will compensate for excessive capacitive loading on the SCSI data signals by inserting an extra SCSICLK period between the instant data is being asserted on the bus to the point when either REQ# or ACK# is asserted. This bit is cleared by a hardware SRESET# or a software Reset SCSI Core, but is unaffected by a SCSI reset.

Bit 6, SRD: SCSI Reset Reporting Interrupt Disable.

Setting this bit will disable the reporting of a SCSI reset. A SCSI reset will have its normal effect of disconnecting the device from the SCSI bus but will allow it to remain idle in the disconnected state without generating a host interrupt. If this bit is not set, the device will interrupt the host before disconnecting itself from the SCSI bus. A hardware SRESET# or a software Reset SCSI Core will clear this bit, but a SCSI reset will not.

Bit 5, Parity_T: Parity Test.

This bit allows parity errors to be created to enable the testing of hardware or software associated with the SCSI subsection. When this bit is set, the parity signals will duplicate the most significant bit of the byte to which they are attached when unloading the FIFO to the SCSI bus. This bit must not be zero during normal operation. It is cleared by either a hardware SRESET# or a software Reset SCSI Core, but is unaffected by a SCSI reset.

Bit 4, En_P_Ck: Enable Parity Checking.

Setting this bit will cause the SCSI device to check parity on incoming SCSI bytes during any Information Transfer Phase except on receiving PAD bytes. When a parity error is detected, a bit will be set in the Status Register. No interrupt will be asserted. In Initiator role, bad parity will cause ATN# to be asserted on the SCSI bus. If this bit is not set, parity checking will be bypassed and both the Status Register bit and the ATN# signal will not be asserted. This bit is cleared by either a hardware SRESET# of a software Reset SCSI Core, but is unaffected by a SCSI reset.

Bit 3, SCSI Chip Test: SCSI Subsection Test Mode Enable.

This bit is used for testing only and should not be set during normal operation. When this bit is set, the Test Register at address 0A hex is enabled and the device will be placed under a special test mode. A hardware SRESET# or a software Reset SCSI Core will reset this bit. This must be performed everytime after Chip Test Mode is used before resuming normal operation.

Bit 2 - 0, BID[2:0], My Bus ID.

This is a 3-bit binary field for storing the Bus ID of this device. This ID is used by the device to respond to any bus initiated Selection or ReSelection; or simply when it is arbitrating for the bus. This field is unaffected by any hardware or software resets, and must be initialized after power-up of this device.

### 3.5.8.14 Clock Conversion		Write address 09

| Clock Conversion Register |
|---|

The Clock Conversion register contains a coded value for the input SCSICLK frequency which must be set correctly for timeouts greater than 400ns. There are four legal values as specified in the table below. The register is initialized to 2 by a hardware or software RESET and is unchanged by SCSI reset. The unused 5 bits of this register are reserved.

| SCSICLK Frequency (MHz) | CLK Conversion Factor |
|---|---|
| 10 | 2 |
| 10.01 to 15 | 3 |
| 15.01 to 20 | 4 |
| 20.01 to 25 | 5 |

### 3.5.8.15 Test Mode Register		Write Address 0A

The Test Mode Register controls SCSI subsection operation when in the Test Mode. The Test Mode is entered by setting the SCSI Chip Test Mode Enable bit in the Configuration-1 Register and can only be cleared by a hardware or software RESET.

**Test Mode Register**

| reserved | reserved | reserved | reserved | reserved | Hi-Z | Init | Target |
|---|---|---|---|---|---|---|---|
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

Write Address 0A

Bit 2, HiZ: All Outputs to High Impedance

Setting bit 2 forces all bidirectional and output pins to the high impedance state.

Bit 1, Init: Initiator Mode

Setting bit 1 puts the SCSI CORE into Initiator mode and any Initiator command will be accepted.

Bit 0, Target: Target Mode

Setting bit 0 puts the SCSI CORE into Target mode and any Target command will be accepted.

### 3.5.8.16 Configuration-2 Register R/W Address OB

The register is cleared after a hardware or software RESET.

**Configuration-2 Register**

| RFB | EPL | Reserved | DREQ HiZ | SCSI - 2 | BPA | Reserved | Reserved |
|---|---|---|---|---|---|---|---|
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

Read Write Address OB

Bit 7, RFB: Reserve FIFO Byte

Setting this bit reserves a byte in the FIFO so that DMA transfers can begin on odd byte boundaries during Initiator Synchronous Data In. Since the byte is reserved when the phase changes to synchronous Data In, this bit must be set before the phase change. The reserved byte is merged with the first byte received across the SCSI bus and becomes the low byte of the first 16-bit word that is written to memory.

When the start address is odd, the interrupt service routine for a phase change to synchronous Data In must copy the byte at start address-1 from memory to SCSI CORE register OF (hex) and then issue the Transfer Info command. When the SCSI CORE writes its first word to memory (via DMA) it will overwrite the byte at start address-1 (which is not part of the current SCSI data block) with the copy placed in register OF (hex).

The Reserve FIFO Byte bit is cleared by either a hardware or software RESET and is unchanged by a SCSI Bus Reset. The bit is also cleared by a write to register OF(hex) when the Interrupt output is true and the SCSI phase is synchronous Data In. The bit has no effect for phases other than Initiator Synchronous Data In.

Bit 6, EPL: Enable Phase Latch

Setting this bit causes the phase to be latched at each command completion which makes the software for stacked commands simpler. The latch is reopened when the Interrupt register is read. If the Enable Phase Latch bit is cleared, the status register phase bits are instantaneous indicators of the SCSI phase line state. The Enable Phase Latch bit is cleared by hardware or software RESET and is unchanged by SCSI Reset.

Bit 5 Reserved

This bit is reserved and should be read and written as zero.

Bit 4, DREQ HiZ: DREQ High Impedance

Setting this bit forces the SDREQ output (DMA Request) to high impedance and is used to wire-OR the DMA Request line from several devices. The SCSI CORE will ignore any activity on the SDACK# (DMA acknowledge) input while in this state.

Clearing this bit allows the SCSI core to drive the SDREQ output and respond to SDACK# to decrement the DMA Counter and load or unload the FIFO. SDACK# should not pulse true without RD16# or WR16# or the DMA Counter will decrement without transferring any data. See *DMA Counter Register.*

Bit 3, SCSI-2: SCSI-2 feature enable

Setting this bit allows the SCSI CORE to support the following SCSI-2 features:

Tagged-Queuing

The SCSI core will request one or three Message bytes before switching to the Command Phase depending on the status of ATN# after the first received Message byte. Should ATN# be true after reception of the first byte, two more will be requested by the SCSI core. On the contrary, if ATN# goes false after the first byte, or if the SCSI-2 bit is cleared, SCSI core will switch to Command Phase immediately. When the SCSI-2 bit is cleared, the Selection sequence will be aborted if the Target does not switch to Command Phase after one message byte has been received. Also see section 3.5.1 SCSI Selection.

Group 2 Commands

SCSI-2 Group 2 commands are l0 byte commands. When they are received with the SCSI-2 bit set, the SCSI core will set the Valid Group Code bit. On the contrary, when and SCSI-2 Group commands are received with the SCSI-2 bit cleared, the SCSI core will treat these commands as Reserved Commands and will request for only 6 bytes in the Command Phase and will not set the Valid Group Code bit.

Bit 2, BPA: Target Bad Parity Abort

Setting this bit causes the SCSI CORE to abort a Receive Command or Receive Data sequence when a parity error is detected.

Bit 1 Reserved

This bit is reserved and should be read and written as zero.

Bit 0 Reserved

This bit is reserved and should be read and written as zero.

### 3.5.8.17 Vendor ID/Rev / Configuration-3 Register R/W address OC

The first read to this register, before any other SCSI core register accesses, after power up returns the SCSI core revision information about the CURIO. The ID/rev number is coded as an 8-bit byte. Subsequent accesses to this location will return information in the Configuration-3 Register. The format of which is described below:

The register is cleared after a hardware or software RESET for software compatibility with previous generation products.

**Configuration-3 Register**

| reserved | reserved | reserved | reserved | reserved | SRB | DMA | THRSH |
|---|---|---|---|---|---|---|---|
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

Read Write Address OC

Bit 2, SRB: Save Residual Byte

The residual byte is the remaining byte at the end of a 16-bit DMA data stream when the block does not end with a full word.

Setting this bit prevents the DMA Request (SDREQ) from being asserted for the residual byte at the end of a transfer. The microprocessor must transfer the residual byte to or from the FIFO. If this bit is clear and there is a single byte left over, SDREQ will be asserted and a 16-bit DMA transfer will move the last byte on the lower half of the bus. The upper byte is set to ones if the operation is a DMA read. The Save Residual Byte bit is cleared by hardware or software RESET and is unchanged by SCSI Reset. It has no effect during any SCSI phase except Data In or Data Out.

Bit 1, DMA: Alternate DMA Mode

Setting this bit allows the DMA interface to use the Demand Mode on an Am9517 DMA controller if the Threshold-8 bit is also set (refer to the section 3.5.5 on *Burst Mode DMA*). This bit should only be set if Threshold-8 is also set. All of the DMA burst transfers will be four words long except during the last word, when any remaining number of one to four words are transferred.

When using the Alternate DMA mode, SDACK# remains asserted and a word is transferred each time WR16# or RD16# toggles. If the Alternate DMA bit is clear, SDACK# is asserted and WR16# toggles for a DMA write or data is output when SDACK# is true for a DMA read. RD16# is not needed for a single word transfer.

Bit 0, THRSH: Threshold-8

Setting this bit causes the SCSI CORE to delay SCSI DMA request (SDREQ) until it can transfer four words. This threshold applies only to SCSI data phases.

When Threshold-8 is set, SDREQ will go true when the top eight bytes of the FIFO are empty during a DMA write or the FIFO has 8 bytes available during a DMA read. The DMA Counter must also be greater than 7 in Initiator synchronous data in mode. SDREQ will also go true at the DMA READ end of transfer, ie., (a) the DMA Counter is zero in target mode. (b) ATN# is set in target mode. (c) the DMA Counter is less than 8 in Initiator synchronous data in mode. (d) the DMA Counter is zero in Initiator mode. (e) the phase changes in Initiator mode.

#### 3.5.8.18 FIFO Bottom Register Write address OF

| FIFO Bottom Register |
|---|

Write address 0F

The FIFO Bottom register is used with the Reserve FIFO Byte control bit in Configuration Register-2 to align 16-bit DMA transfers to word boundaries. When the phase changes to Initiator Synchronous Data In, the SCSI CORE reserves a byte in the bottom of the FIFO if the Reserve FIFO Byte control bit is set. After the interrupt for synchronous Data In, the microprocessor should write the byte to the FIFO Bottom register that will become the low byte of the first word transfered out from the FIFO to the external DMA controller. The first byte received across the SCSI bus is merged with the FIFO Bottom register contents to form this first word.

### 3.5.9 Command Execution

When the SCSI core interrupts for CPU service, the cause of interrupt can be investigated through examining the Sequence Step Register (Reg 06H) in conjunction with the Interrupt Register (Reg 05H). The following tables provide detailed information regarding each interrupt in which the SCSI core needs CPU intervention. Each table corresponds to a command. For example, if a user issues a "SELECT with ATN3", the SCSI core will continue executing the command until it completes it or encounters a problem. At this time the SCSI core will set its interrupt active, and update both Reg 06H and 05H for the cause of the interrupt. The user may examine the cause of the interrupt by reading these two register and decide on the next course of action. The tables are divided into two groups: INITIATOR MODE INTERRUPTS and TARGET MODE INTERRUPTS.

#### 3.5.9.1 Initiator mode interrupts

Command: SELECT without ATN

| Reg 06H | Reg 05H | |
|---|---|---|
| 0H | 20H | Arbitration completed. Target timed out. SCSI core disconnected. |
| 2H | 18H | Arbitration & selection completed. Target did not change to SCSI command phase. |
| 3H | 18H | Arbitration & selection completed. Target changed to Command phase, but Target changed phase before all the command bytes are transferred. |
| 4H | 18H | Arbitration & selection & command phases completed normally. |

Command: SELECT with ATN

| Reg 06H | Reg 05H |
|---|---|

| | | |
|---|---|---|
| 0H | 20H | Arbitration completed. Target timed out. SCSI core is in disconnected state. |
| 0H | 18H | Arbitration & selection completed. Target did not change to SCSI MSG-out phase & ATN is still active. |
| 2H | 18H | Arbitration & selection completed. Target changed to MSG-out phase, one byte of MSG transferred, ATN# is released. **Target did not change to Command Phase.** |
| 3H | 18H | Arbitration & selection & MSG-out phase completed normally, but Target changed phase before all the CDB bytes are transferred. Some bytes are left in the FIFO, which can be determined by reading the FIFO Flags register. |
| 4H | 18H | Arbitration, selection, MSG-out, Command phase completed normally. |

Command: SELECT with ATN3

| Reg 06H | Reg 05H | |
|---|---|---|
| 0H | 20H | Arbitration completed. Target timed out. SCSI core is in disconnected state. |
| 0H | 18H | Arbitration & selection completed. Target did not change to SCSI MSG-out phase & ATN# is still active. |
| 2H | 18H | Arbitration & selection completed. Target changed to MSG-out phase. 1, 2 or 3 byte of MSG transferred. Target either changed phase before the third byte, or did not assert Command phase. ATN# may or may not be active. It is active if the third byte has not been sent. |
| 3H | 18H | Arbitration & selection & MSG-out phase completed normally, but Target changed phase before all CDB bytes are transferred. Some bytes are left in the FIFO, which can be determined by reading the FIFO Flag register. |
| 4H | 18H | Arbitration, selection, MSG-out, Command phases all completed normally. |

Command: SELECT with ATN & Stop

| Reg 06H | Reg 05H | |
|---|---|---|
| 0H | 20H | Arbitration completed. Target timed out. SCSI core is in disconnected state. |
| 0H | 18H | Arbitration & selection completed. Target did not change to SCSI MSG-out phase & ATN# is still active. |
| 2H | 18H | Arbitration & selection completed. Target changed to MSG-out phase, one byte of MSG transferred, ATN# is still active. Additional MSG bytes could be sent by filling the FIFO and issuing Transfer Information Command. ATN# will remain active till transfer counter decrements to zero. |

#### 3.5.9.2 Target mode interrupts

Command: SCSI Core selected as a Target without ATN

| Reg 06H | Reg 05H | |
|---|---|---|
| 0H | 01H | SCSI Core has been selected. Initiator ID has been loaded into FIFO. |
| 1H | 01H | SCSI Core has been selected. Initiator ID has been loaded into FIFO. Command phase encountered a parity error. Some Command bytes are in the FIFO. Check FIFO Flags register to determine how many bytes are received. |
| 1H | 11H | Same as above. However, Initiator has interrupted Command phase by asserting ATN#. |
| 2H | 01H | SCSI core has been selected. Initiator ID & entire CDB received. Check bit 3 of register 4 to verify the command group valid ID. |
| 2H | 11H | SCSI core has been selected. Initiator ID & entire CDB received. However, ATN# is asserted in Command phase. Check bit 3 of register 4 for Command group valid ID. |

Command: SCSI core selected as a Target with ATN, SCSI-2 bit not set.

| Reg 06H | Reg 05H | |
|---|---|---|
| 0H | 02H | SCSI Core has been selected. Initiator ID has been loaded into FIFO as well as one byte of MSG. Invalid ID message or parity error encountered. |
| 1H | 12H | SCSI Core has been selected. Initiator ID has been loaded into FIFO as well as one byte of MSG. However, ATN# is still asserted by the Initiator. |
| 1H | 02H | Proceeded to Command phase. Parity error is encountered in the Command bytes. Check bit 3 of register 4 (Valid Group Code). |
| 2H | 12H | Proceeded to Command phase. Parity error encountered and ATN# has been asserted by the Initiator. |
| 2H | 02H | SCSI Core has been selected. Initiator ID, one MSG byte & entire CDB has been received. |
| 2H | 12H | Same as the above. However, Initiator has asserted ATN# in the Command phase. |

Command: SCSI Core selected as a Target with ATN, SCSI-2 bit set.

| Reg 06H | Reg 05H | |
|---|---|---|
| 0H | 02H | SCSI Core has been selected. Initiator ID has been loaded into FIFO as well as one byte of MSG. Invalid ID message or parity error encountered. |
| 4H | 02H | SCSI Core has been selected. Initiator ID has been loaded into FIFO as well as one byte of MSG. Parity error encountered during the second and third MSG bytes. |
| 4H | 12H | SCSI Core has been selected. Initiator ID has been loaded into FIFO. Three MSG bytes are received. However, ATN# is still asserted by Initiator. |

| | | |
|---|---|---|
| 5H | 02H | Proceeded to Command phase. Parity error encountered. Check bit 3 of register 4H for Valid Group Code. Check FIFO Flags register to determine how may bytes has been received. |
| 5H | 12H | Same as the above. ATN# has also been asserted in the CMD phase. |
| 6H | 02H | SCSI Core has been selected. Three bytes of MSG and the entire CDB has been received. |

Command: Target Receive Command

| Reg 06H | Reg 05H | |
|---|---|---|
| 1H | 04H | Parity error encountered during command transfer. Check FIFO Flags to determine the number of bytes left in the FIFO. |
| 1H | 14H | Same as the above. However, ATN# has also been asserted. |
| 2H | 04H | SCSI Core has received the entire command block successfully. |
| 2H | 14H | Same as the above with ATN# asserted by the Initiator. |

Command: Target Disconnect sequence Command

| Reg 06H | Reg 05H | |
|---|---|---|
| 0H | 14H | One byte of message has been sent. Initiator has asserted ATN#. |
| 1H | 14H | Two bytes of message has been sent. Initiator has asserted ATN#. |
| 2H | 24H | SCSI Core has completed disconnect sequence successfully and released the SCSI bus. |

Command: Target Terminate sequence Command

| Reg 06H | Reg 05H | |
|---|---|---|
| 0H | 14H | One byte of status has been sent. Initiator has asserted ATN#. |
| 1H | 14H | Two bytes of status & message has been sent. Initiator has asserted ATN#. |
| 2H | 24H | SCSI Core has completed terminate sequence successfully and released the SCSI bus. |

Command: Target Command Complete

| Reg 06H | Reg 05H | |
|---|---|---|
| 0H | 14H | One byte of status has been sent. Initiator has asserted ATN#. |
| 1H | 14H | Two bytes of status & message has been sent. Initiator has asserted ATN#. |
| 2H | 04H | SCSI Core has completed Command complete sequence successfully and released the SCSI bus. |

### 3.5.10 Command Set

The SCSI core moves data between its internal FIFO and the SCSI bus. DMA commands use an external DMA controller to move data between the FIFO and memory. Non-DMA commands require the microprocessor to move data between the FIFO and memory. DMA commands will load the DMA Counter with the initialized value in the DMA Count Register. DMA commands have bit 7 set and non-DMA commands have bit 7 cleared.

## SCSI CORE COMMAND SET

| Command Register (Hex) DMA | Non-DMA | Command Mnemonic | Interrupt |
|---|---|---|---|
| | | **INITIATOR STATE GROUP** | |
| 90H | 10H | Transfer information | yes |
| 91H | 11H | Initiator command complete sequence | yes |
| - | 12H | Message accepted | yes |
| 98H | 18H | Transfer pad | yes |
| - | 1AH | Set ATN | no |
| - | 1BH | Reset ATN | no |
| | | **TARGET STATE GROUP** | |
| A0H | 20H | Send message | yes |
| A1H | 21H | Send status | yes |
| A2H | 22H | Send data | yes |
| A3H | 23H | Disconnect sequence | yes |
| A4H | 24H | Terminate sequence | yes |
| A5H | 25H | Target command complete sequence | yes |
| A7H | 27H | Disconnect | no |
| A8H | 28H | Receive message sequence | yes |
| A9H | 29H | Receive command | yes |
| AAH | 2AH | Receive data | yes |
| ABH | 2BH | Receive command sequence | yes |
| 84H | 04H | Target abort DMA | no** |
| | | **DISCONNECTED STATE GROUP** | |
| C0H | 40H | Reselected sequence | yes |
| C1H | 41H | Select without ATN sequence | yes |
| C2H | 42H | Select with ATN sequence | yes |
| C3H | 43H | Select with ATN and stop sequence | yes |
| C4H | 44H | Enable selection / reselection | no |
| C5H | 45H | Disable selection / reselection | yes |
| C6H | 46H | Select with ATN3 | yes |

**MISCELLANEOUS GROUP**

| | | | |
|---|---|---|---|
| 80H | 00H | NOP | no |
| 81H | 01H | Flush FIFO | no |
| 82H | 02H | Reset SCSI Core | no |
| 83H | 03H | Reset SCSI bus | no |

**** This command does not cause an interrupt. However, it may allow a stalled command to complete and generate an interrupt.**

Table B.16

### 3.5.10.1 Initiator Commands

When the SCSI CORE is in the Initiator State, it will only accept Initiator Commands. An Illegal Command Interrupt will be generated if a Disconnected or Target Command is received or if an Initiator Command is received while in another state. An illegal command will be ignored and the Command Register cleared.

Should BSY be deasserted during the Initiator State, a Disconnected Interrupt will be generated within 1.5 to 3.5 SCSICLK cycles.

A parity error is signalled if parity checking is enabled and the SCSI CORE detects a parity error while in Initiator mode. The signal consists of asserting ATN# before ACK# is deasserted for the byte which has the error. However, after a phase change to Synchronous Data In, there is a delay in parity error reporting. Neither the parity bit in the Status Register will be set nor will ATN# be asserted until after the SCSI CORE receives the next Transfer Information Command.

A description of the events during a phase change to Synchronous Data In demonstrates this delay in error reporting. First the DMA interface is disabled, the FIFO Flags are latched to indicate how many bytes were in the FIFO (to be discarded), and the FIFO is cleared. The FIFO is then loaded with the first Data In byte, an interrupt is generated, and the FIFO continues to be loaded with incoming Data In bytes until the specified offset is reached. The microprocessor then issues the Transfer Information Command to re-enable the DMA interface and continue receiving Data In bytes. If a parity error occurred during the Data In phase, the parity bit will be set with ATN# asserted when the next Transfer Information Command is received. If parity error occured on a previous input phase (Status or Message In) when parity checking is enabled, then the parity error flag will be set in the Status Register and the ATN# pin is asserted on the SCSI bus.

Transfer Information — Commands: 90H, 10H.

The Transfer Information command is used for Data Transfer and can also be used to transfer Information Phase bytes. Data transfer is continuous until terminated by one of the following:

- On successful transfer, a Bus Service Interrupt is generated. DMA transfers are complete when the Target asserts REQ#, the FIFO is empty and the DMA Count is zero. Non-DMA transfers are complete when the Target asserts REQ# and the FIFO is empty. For non-DMA transfers during which the SCSI CORE is receiving bytes from the SCSI bus, the transfer is complete after each byte is received. An interrupt will be generated for every byte.

  During a Message Out phase, the SCSI CORE will remove ATN# prior to asserting ACK# for the last byte of the message. This can be determined by the FIFO Flags for non-DMA or the DMA Counter for DMA.

- The transfer is terminated if the Target changes phase. After the Target asserts REQuest, the Command Register will be cleared and a Bus Service Interrupt will be generated.

- The transfer is terminated if the Target releases BSY at which time a Disconnected Interrupt will be generated.

- The transfer is terminated when the SCSI CORE receives the last byte of a Message In Phase as determined by the DMA Counter for DMA transfers. Every byte is assumed to be the last byte for non-DMA transfers. ACK# remains asserted and a Function Complete Interrupt will be generated.

Data is handled on a byte-by-byte basis for all Status Phase and Message In transfers. For DMA transfer commands, consecutive bytes will only be received after the previous byte has been written into memory, and that the FIFO is empty. For non-DMA transfer commands, an interrupt will be generated for each received byte.

Initiator Command Complete Sequence — Commands: 91H, 11H.

This command tells the SCSI CORE to accept a Status byte and a Message byte. If the Target does not assert Message In Phase, or if it disconnects, then the device will terminate early. But after receiving the Message byte, the SCSI CORE will keep ACK# asserted so that it can assert ATN# if the message is unacceptable.

Message Accepted — Commands: 12H.

The Message Accepted command deasserts ACK# on the SCSI bus. Any command that receives Message Phase bytes will leave ACK# asserted. The message can be accepted by issuing this command. The message can be rejected by first asserting ATN# and then issuing Message Accepted.

Transfer Pad — Commands: 98H, 18H.

The Transfer Pad command is used to satisfy DMA Counts when a Target requests more data than there is to be sent by the Initiator or when an Initiator must receive and discard a number of bytes from a Target.

Null bytes are placed in the FIFO and sent to the SCSI bus when transmitting to the bus. DMA must be enabled even though no DMA requests are made since the SCSI CORE uses the DMA Counter to end the transfer. When receiving from the SCSI bus, the received data is placed in the FIFO and discarded.

The same terminating conditions as the Transfer Info command will also terminate Transfer Pad except that the ACK# signal will not be asserted on the last byte of a Message In Phase. The FIFO may not be emptied if the command terminates before the DMA Counter reaches zero (because of either a disconnect or a change in SCSI bus phase).

Set ATN — Commands: 1AH.

The Set ATN command asserts ATN# on the SCSI bus and does not generate an interrupt. ATN# will remain asserted until the last byte of a Message Out Phase or if the Target disconnects. The last byte is determined by the DMA Counter and an empty FIFO as in the Transfer Information command.

Reset ATN — Commands: 1BH.

The Reset ATN command causes ATN# to be released and does not generate an interrupt. It is provided to support earlier generation devices which do not properly comply with the Common Command Set.

**Do not use this command when connected to a device supporting the Common Command Set.**

### 3.5.10.2 Target Commands

While the SCSI CORE is in Target State, it can only receive the following list of SCSI commands. An Illegal Command Interrupt will occur if Disconnected or Initiator commands are received or if Target command is received while the SCSI CORE is in another state. An illegal command is ignored and the leaves the Command Register clear.

A Function Complete interrupt will be generated after normal completion of valid Target commands. If ATN# is asserted while the SCSI CORE is in Target State, the Bus Service bit will be set in the Status Register. A Bus Service Interrupt will be generated if the SCSI CORE was idle when ATN# was asserted. In this case the Function Complete bit will be zero and the Command Register will be cleared.

Send Message — Commands: A0H, 20H.

The Send Message command asserts Message In Phase and sends data until the FIFO is empty or if using DMA until the FIFO is empty and the DMA Counter is zero.

Send Status — Commands: A1H, 21H.

The Send Status command is identical to the Send Message command except that Status Phase is asserted.

Send Data — Commands: A2H, 22H.

The Send Data command is identical to the Send Message command except that Data In Phase is asserted.

Disconnect Sequence — Commands: A3H, 23H.

The Disconnect Sequence command asserts Message In Phase, sends two bytes, and disconnects from the SCSI bus. Normally, the two bytes will be Save Data Pointers and Disconnect. The microprocessor (or DMA) must place these messages in the FIFO before the command is issued. If the initiator asserts ATN#, the Bus Service and Function Complete bits will be set and an interrupt will be generated but the SCSI CORE will remain connected.

Terminate Sequence — Commands: A4H, 24H.

The Terminate Sequence command asserts Status Phase, sends one byte, then asserts Message In Phase and sends another byte. The microprocessor (or DMA) must place these two bytes in the FIFO before the command is issued. If the initiator asserts ATN#, the Bus Service and Function Complete bits will be set and an interrupt will be generated but the SCSI CORE will remain connected.

Target Command Complete Sequence — Commands: A5H, 25H.

The Target Command Complete Sequence command is used to terminate sequences of linked commands. It asserts Status Phase, sends one byte, then asserts Message In Phase and sends another byte which will usually be Command Complete. The microprocessor (or DMA) must place these two bytes in the FIFO before the command is issued. If the initiator asserts ATN#, the Bus Service and Function Complete bits will be set and an interrupt will be generated but the SCSI CORE will remain connected.

Disconnect — Commands: A7H, 27H.

The Disconnect command releases all SCSI bus signals except RST which will not be released if it is asserted and the pulse duration of 25-40 µs depending on SCSICLK frequency and Clock Conversion Factor has not yet elapsed. The SCSI CORE returns to the disconnected state without generating an interrupt.

Receive Message Sequence — Commands: A8H, 28H.

The Receive Message Sequence command asserts Message Phase and receives bytes from the Initiator. For a non-DMA Receive Message Sequence command, an interrupt is generated after one byte and for a DMA Receive Message Sequence Command, an interrupt is generated after the DMA Counter decrements to zero.

Receive Command — Commands: A9H, 29H.

The Receive Command command is identical to the Receive Message Sequence command except that Command Phase is asserted.

Receive Data — Commands: AAH, 2AH.

The Receive Data command is identical to the Receive Message Sequence command except that Data Out Phase is asserted.

Receive Command Sequence — Commands: ABH, 2BH.

The Receive Command Sequence command asserts Command Phase and receives a variable number of bytes depending on the the Group Code field of the first byte. Group 2 commands are 10 byte commands for SCSI-2. If the SCSI-2 bit in the Configuration-2 register is clear, Group 2 commands are 6 byte Reserved commands. Group 3 and 4 commands are 6 byte reserved commands, Group 6 are 6byte Vendor Unique Commands, and Group 7 are 10 byte Vendor Unique Commands.

Target Abort DMA — Commands: 84H, 04H.

The Target Abort DMA command allows the microprocessor to stop a DMA data transfer command. An Illegal Command interrupt will be generated if the SCSI CORE is not in Target State when the command is executed. Target Abort DMA may only be used under the following conditions:

l) The current command is Target Send Data, the DMA controller is stopped and the FIFO is empty.
2) The current command is Target Receive Data, the transfer mode is asynchronous, the DMA controller is stopped and the FIFO is full or the DMA Count is zero.
3) The current command is Target Receive Data, the transfer mode is synchronous, the DMA controller is stopped and the FIFO Flags are 15 or the DMA Count is zero.

When the SCSI CORE receives this command, it will reset the DMA interface (de-assert SDREQ) and terminate the current command. An interrupt occurs immediately at the end of sending or receiving asynchronous data when the offset counter is zero. On receiving synchronous data, interrupts occur after all outstanding SCSI ACKs have been received. The FIFO will contain data that should be removed by the microprocessor if a Target Receive Data was stopped.

### 3.5.10.3 Disconnected State Commands

While the SCSI CORE is in the Disconnected State, the Disconnected State commands are the only valid events. An Illegal Command interrupt will occur if Initiator or Target commands are received or if a Disconnected Command is received when in the non Disconnected Command state. An illegal command is ignored and leaves the command register clear.

ReSelect Sequence — Commands: C0H, 40H.

The ReSelect Sequence command will cause the SCSI CORE to arbitrate for the bus and enter ReSelection Phase when it wins. The microprocessor (or DMA) must place the Identify message in the FIFO before the command is issued as required by the SCSI protocol. The Time-Out and Destination ID Registers must also have been programmed before the command is issued. A Function Complete interrupt occurs after normal termination. If a ReSelect Time-Out occurs, the sequence will terminate early.

Select Without ATN Sequence — Commands: C1H, 41H.

The Select Without ATN Sequence command will cause the SCSI CORE to arbitrate for the bus, enter Selection Phase when it wins and followed by sending the Command Descriptor Block (CDB). The microprocessor (or DMA) must place the 6, 10 or 12 byte CDB in the FIFO before the command is issued. The Time-Out and Destination ID Registers must also have been programmed before the command is issued. A Function- Complete/Bus-Service interrupt occurs after normal termination. If a ReSelect Time-Out occurs, or the Target does not assert Command Phase, or the Target removes Command Phase early, the sequence will terminate early.

Select With ATN Sequence — Commands: C2H, 42H.

The Select With ATN Sequence command will cause the SCSI CORE to Arbitrate for the bus. A device with ATN# true is then selected, to be followed by sending one Message Phase byte with subsequent Command Phase bytes. The microprocessor (or DMA) must place the message and command bytes in the FIFO before the command is issued. The Time-Out and Destination ID Registers must also have been programmed prior to issuing the command. A Function-Complete and BusService interrupt occurs after normal termination. If a Select Time-Out occurs, or the Target does not assert Message Phase followed by Command Phase, or the Target removes Command Phase early, the sequence will terminate early.

Select With ATN and Stop — Commands: C3H, 43H.

The Select With ATN and Stop command is used in place of Select With ATN Sequence when multiple Message Phase bytes need to be sent. The command selects a Target with ATN# asserted and sends one Message Phase byte; initiates a bus service interrupt and then stops. After the interrupt, message bytes can be placed into the FIFO. ATN# will remain true while a Transfer Info command is empting the FIFO. If a DMA Transfer Info command is used, the ATN# signal will remain true until the DMA Counter reaches zero.

Enable Selection / ReSelection — Commands: C4H, 44H.

The Enable Selection/Reselection command allows the SCSI CORE to respond to bus initiated Selection or ReSelection. The command will be cancelled by any command that causes the SCSI CORE to Select or ReSelect. It must be re-issued within 250 ms after the SCSI CORE disconnects in order to remain compliant with ANSI recommended timings. If DMA is enabled, incoming data will be transferred to memory. Otherwise, incoming data remains in the FIFO.

Disable Selection / ReSelection — Commands: C5H, 45H.

The Disable Selection/ReSelection command cancels an earlier Enable Selection/ReSelection command. A Function Complete Interrupt occurs as long as bus initiated Selection or ReSelection had not yet begun. Otherwise, this command (and every other command) will be ignored. Also look at the sections on SCSI Bus Sequences and Host Bus Sequences.

Select With ATN3 Sequence — Commands: C6H, 46H.

The Select With ATN3 Sequence command causes the SCSI CORE to arbitrate for the bus, select a device with ATN# asserted, send Message Phase bytes (3), deassert ATN#, and send the Command Phase bytes (9, 13 or 15 bytes). The microprocessor (or DMA) must place the message and command bytes in the FIFO before the command is issued. The Time-Out and Destination ID Registers must also have been programmed. A Function-Complete and Bus Service interrupt occurs if the command terminates normally. If a Selection Time-Out occurs, or if the Target does not assert Message Phase followed by Command Phase, or if the Target removes Command Phase early, the sequence will terminate early.

### 3.5.10.4 Miscellaneous Commands

NOP — Commands: 80H, 00H.

The NOP command does nothing for a non-DMA command but will load the DMA Counter with the value in DMA Count Register for a DMA NOP. The SCSI CORE only needs this command after a hardware or software RESET. NOP does not generate an interrupt.

Flush FIFO — Commands: 81H, 01H.

The Flush FIFO command clears the FIFO Flags and the bottom byte of the FIFO.

Reset SCSI Core — Commands: 82H, 02H.

The Reset Chip (SCSI Core) command is a software RESET and has the same effect as a hardware reset. It resets all functions in the SCSI Core and returns it to a disconnected state.

Reset SCSI Bus — Commands: 83H, 03H.

The Reset SCSI Bus command will assert the SCSI Reset Output (RST) signal for Tr us,

$$Tr = 130*(SCSICLK)*(CMF)$$

where SCSICLK is the period of the Clock input to the SCSI CORE and CMF = Clock Multiplier Factor (see *Write Register 9*). An Interrupt will be generated with this command unless disabled by the Config-1 Register (bit 6).

## 3.6 ESCC FUNCTIONAL DESCRIPTION

### 3.6.1 ESCC Block Diagram

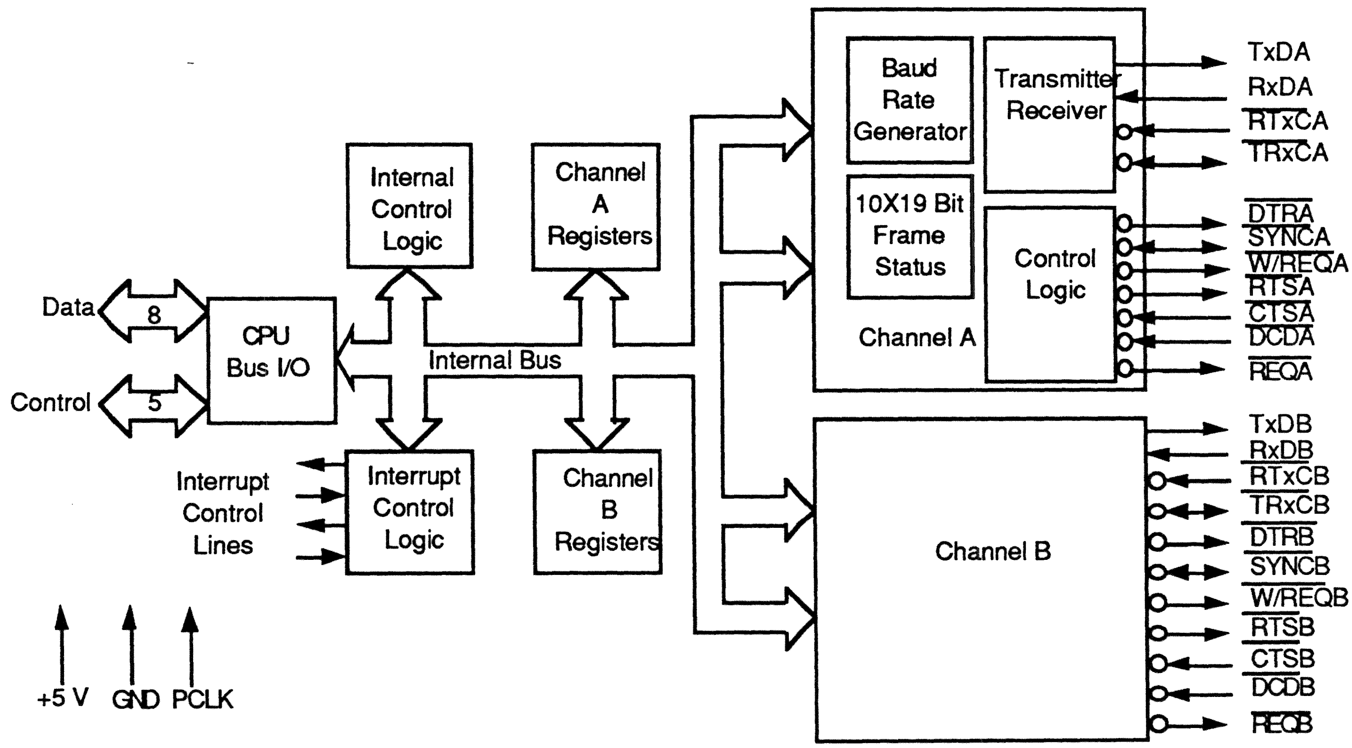



Figure C.1

### 3.6.2 ESCC Data Path

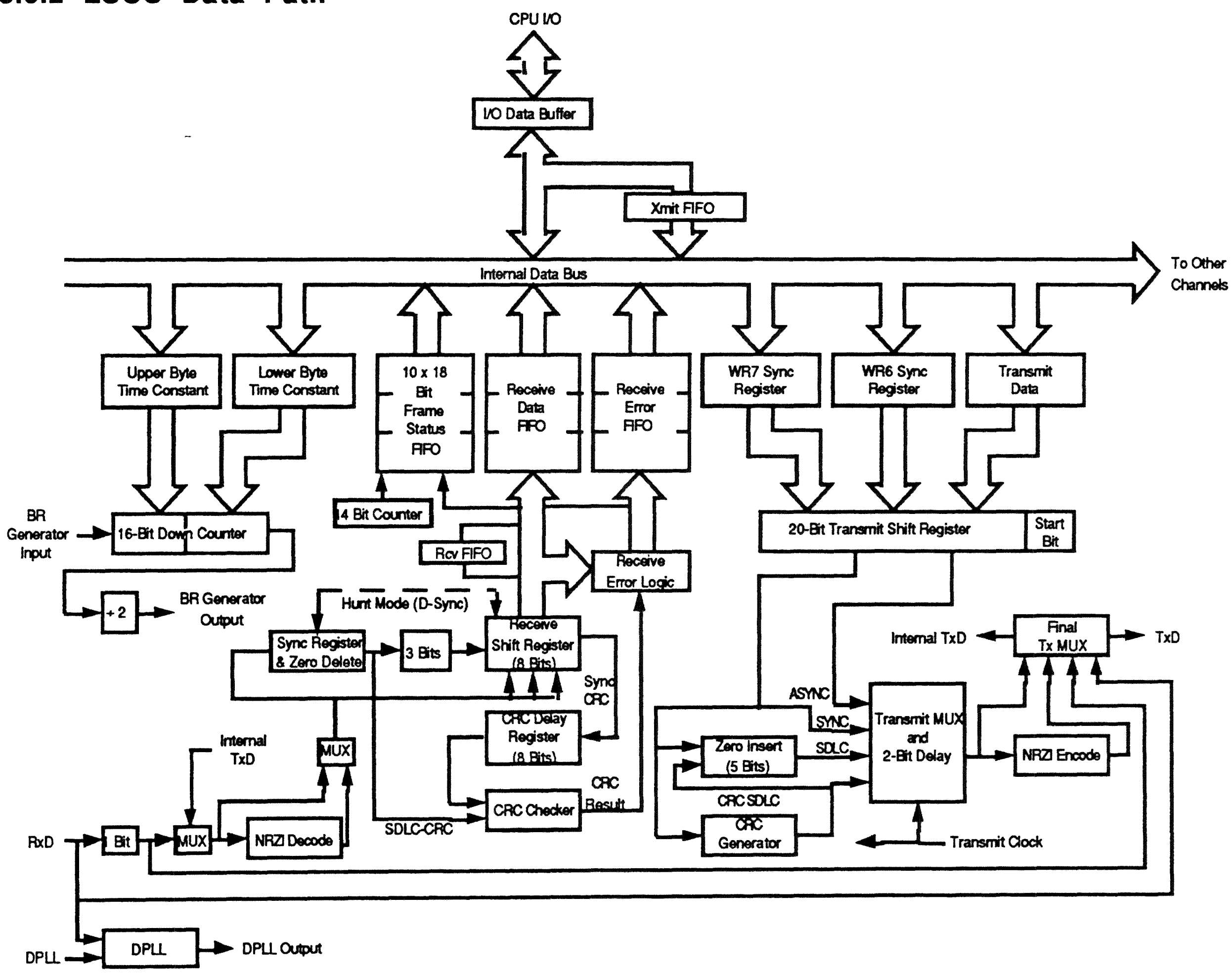



Figure C.2

### 3.6.3 Detailed ESCC Description

The functional capabilities of the ESCC can be described from two different points of view; as a data communications device, it transmits and receives data in a wide variety of data communications protocols; as a microprocessor peripheral, it interacts with the CPU and provides vectored interrupts and handshaking signals.

#### 3.6.3.1 Data Communications Capabililties

The ESCC provides two independent full-duplex channels programmable for use in any common asynchronous or SYNC data-communication protocol. The following description briefly detail these protocols.

### 3.6.3.2 Asynchronous Modes

Transmission and reception can be accomplished independently on each channel with five to eight bits per character, plus optional even or odd parity. The transmiters can supply one, one-and-a-half or two stop bits per character and can provide a break output at any time. The receiver break-detection logic interrupts the CPU both at the start and at the end of a received break. Reception is protected from spikes by a transient spike-rejection mechanism that checks the signal one-half a bit time after a Low level is detected on the receive data input. If the Low does not persist (as in the case of a transient), the character assembly process does not start.

Framing errors and overrun errors are detected and buffered together with the partial character on which they occur. Vectored interrupts allow fast servicing of error conditions using dedicated routines. Futhermore, a built-in checking process avoids the interpretation of framing error as a new start bit; a framing error resumes in the addition of one-half a bit time to the point at which the search tor the next start bit begins.

The ESCC does not require symmetric transmit and receive clock signals- feature allowing use of a wide variety of clock sources. The transmitter and receiver can handle data at a rate of 1, 1/16, 1/32, or 1/64 of the clock rate supplied to the receive and transmit Clock inputs. In asynchronous modes, the SYNC pin may be programmed as an input used for functions, such as monitoring ring indicator.

### 3.6.3.3 Synchronous Modes

The ESCC supports both byte-oriented and bit-oriented synchronous communication. SYNC byte-oriented protocols can be handled in several modes, allowing character synchronization with a 6-bit or 8-bit SYNC character (Monosync), any 12-bit or 16-bit SYNC pattern (Bisync), or with an external SYNC signal. Leading SYNC characters can be removed without interrupting the CPU.

Five- or 7-bit SYNC characters are detected with 8- or 16 bit patterns in the ESCC by overlapping the larger pattern across multiple incoming SYNC characters as shown in Figure C.4.

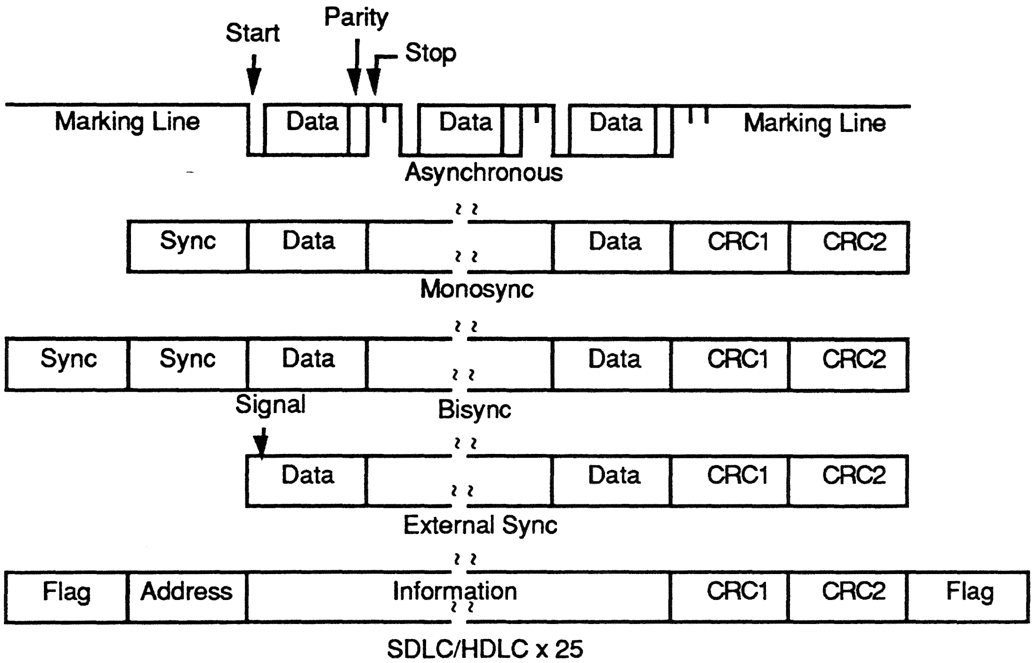



**Figure C.3. SCC Protocols**

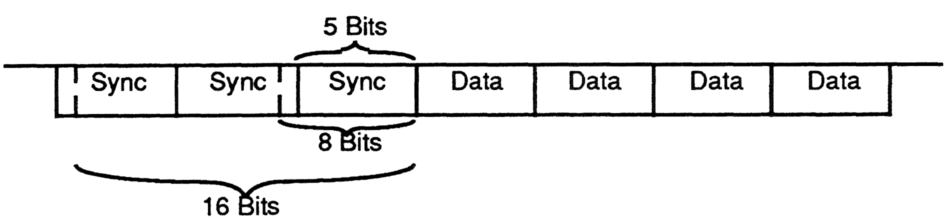



**Figure C.4. Detecting 5- or 7-Bit Sychronous Characters**

CRC checking for Synchronous byte-oriented modes is delayed by one character time so that the CPU may disable CRC checking on specific characters. This permits the implementation of protocols, such as IBM BISYNC.

Both CRC-16 (X16+ X15 + X2 + 1 ) and CCITT (X16 + X12 + X5 + 1 ) error checking polynomials are supported. Either polynomial may be selected in BISYNC and MONOSYNC modes. Users may preset the CRC generator and checker to all "1"s or all "0"s. The ESCC also provides a feature that automatically transmits CRC data when no other data are available for transmission. This allows for high-speed transmissions under DMA control with no need for CPU intervention at the end of a message. When there are no data or CRC to send in SYNC modes, the transmitter inserts 6-, 8-, or 16-bit SYNC characters, regardless of the programmed character length.

The ESCC supports SYNC bit-oriented protocols, such as SDLC and HDLC, by performing automatic flag sending, zero bit insertion, and CRC generation. A special command can be used to abort a frame in transmission. At the end of a message, the ESCC automatically transmits the CRC and trailing flag when the transmitter underruns. The transmitter may also be programmed to send an idle line consisting of continuous flag characters or a steady marking condition.

If a transmit underrun occurs in the middle of a message, an external/status interrupt warns the CPU of this status change so that an abort may be issued. The ESCC may also be programmed to send an abort itself in case of an underrun, relieving the CPU of this task. One to eight bits per character can be sent allowing reception of a message with no prior information about the character structure in the information field of a frame.

The receiver automatically acquires synchronization on the leading flag of a frame in SDLC or HDLC and provides a synchronization signal on the SYNC pin (an interrupt can also be programmed). The receiver can be programmed to search for frames addressed by a single byte (or four bits within a byte) of a user-selected address or to a global broadcast address. In this mode, frames not matching either the user-selected or broadcast address are ignored. The number of address bytes can be extended under software control. For receiving data, an interrupt on the first received character, or an interrupt on every character, or on special condition only (end -of-frame) can be selected. The receiver automatically deletes all "0"s inserted by the transmitter during character assembly. CRC Is also calculated and is automatically checked to validate frame transmission. At the end of transmission, the status of a received frame is available in the status registers. In SDLC mode, the ESCC must be programmed to use the SDLC CRC polynomial, but the generator and checker may be preset to all "1"s or all "0"s. The CRC is inverted before transmission and the receiver checks against the bit pattern "0011101000001111".

NRZ, NRZI or FM coding may be used in any 1X mode. The parity options available in asynchronous modes are available in synchronous modes.

The ESCC can be conveniently used under DMA control to provide high-speed reception or transmission. In reception, for example, the ESCC can interrupt the CPU when the first character of a message is received. The CPU then enables the DMA to transfer the message to memory. The ESCC then issues an end-of-frame interrupt and the CPU can check the status of the received message. Thus, the CPU is freed for other services while the message is being received. The CPU may also enable the DMA first and have the ESCC interrupt only on end-of-frame. This procedure allows all data to be transferred via the DMA.

### 3.6.3.4 SDLC Loop Mode

The ESCC supports SDLC Loop mode in addition to normal SDLC. In an SDLC Loop, there is a primary controller station that manages the message traffic flow and any number of secondary stations. In SDLC Loop mode, the ESCC performs the functions of a secondary station while an ESCC operating in regular SDLC mode can act as a controller (Figure C.5).

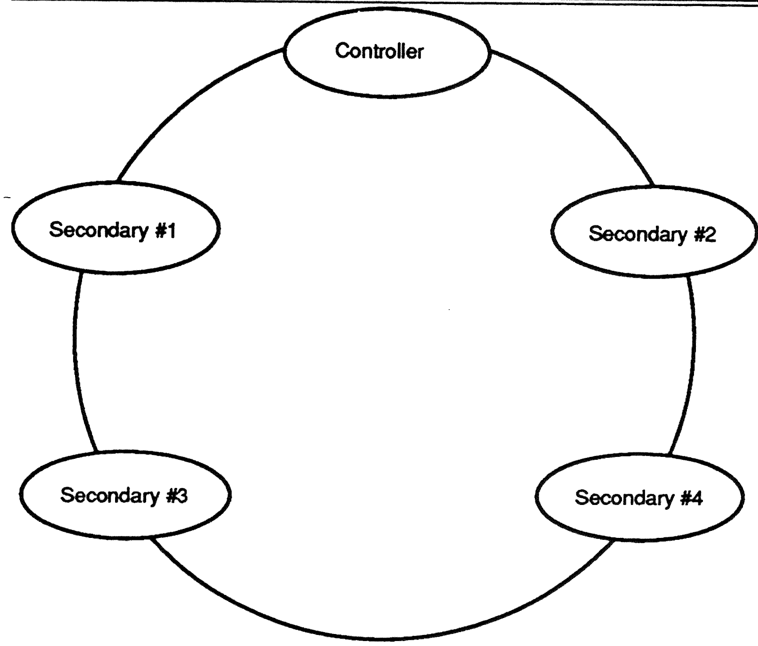



Figure C.5

A secondary station in an SDLC Loop is always listening to the messages being sent around the loop and, in fact, must pass these messages to the rest of the loop by retransmitting them with a 1-bit time delay. The secondary station can place its own message on the loop only at specific times. The controller signals that secondary stations may transmit messages by sending a special character, called an EOP (End of Poll), around the loop. The EOP character is the bit pattern "1111110". Because of zero insertion during messages, this bit pattern is unique and easily recognized.

When a secondary station has a message to transmit and recongnizes an EOP on the line, it changes the last binary one of the EOP to a zero before transmission. This has the effect of timing the EOP Into a flag sequence. The secondary station now places its message on the loop and terminates the message with an EOP. Any secondary stations further down the loop with messages to transmit can then append their messages to the message of the first secondary station by the same process. Any secondary stations without messages to send merely echo the incoming messages and are prohibited from placing messages on the loop (except upon recognizing an EOP).

SDLC Loop mode is a programmable option in the ESCC. NRZ, NRZI, and FM coding may all be used in SDLC Loop mode.

### 3.6.3.5 Baud Rate Generator

Each channel in the ESCC contains a programmable baud rate generator. Each generator consists of two 8-bit time constant registers that form a 16-bit time constant, a 16-bit down counter, and a flip-flop on the output producing a sqarewave. On start-up, the flip-flop on the output is set in a High state; the value in the time constant register is baded into the counter; and the counter starts counting down. The output of the baud rate generator toggles upon reaching zero; the value in the time constant register is baded into the counter, and the process is repeated. The time constant may be changed at any time, but the new value does not take effect until the next load of the counter.

The output of the baud rate generator may be used as either the transmit clock, the receive clock, or both. It can also drive the digital phase-locked loop (see next section).

If the receive clock or transmit clock is not programmed to come from the TRxC pin, the output of the baud rate generator may be echoed out via the TRxC pin.

The following formula relates the time constant to the baud rate where PCLK or RTxC is the baud rate generator input frequency in Hz. The clock mode is X1, X16, X32, or X64 as selected in Write Register 4, bits D6 and D7. Synchronous operation modes should select X1 and Asynchronous should select X16, X32, or X64.

$$\text{Time Constant} = \frac{\text{PCLK or RTxC Frequency}}{2\ (\text{Baud Rate})(\text{Clock Mode})} - 2$$

The following formula relates the time constant to the baud rate. (The baud rate is in bits/second and the BR clock period is in seconds given by Clock Mode/Clock Frequency.)

$$\text{baud rate} = \frac{1}{2 \times (\text{Time Constant} + 2) \times (\text{BR Clock Period})}$$

Time Constant Values
for standard Baud Rates at BR Clock
= 3.9936 MHz

| Baud Rate | Time Constant Decimal / (Hex notation) | Error |
|---|---|---|
| 19200 | 102 (0066) | 0 |
| 9600 | 206 (00CE) | 0 |
| 7200 | 275 (0113) | 0.12% |
| 4800 | 414 (019E) | 0 |
| 3600 | 553 (0229) | 0.06% |
| 2400 | 830 (033E) | 0 |
| 2000 | 996 (03E4) | 0.04% |
| 1800 | 1107 (0453) | 0.03% |
| 1200 | 1662 (067E) | 0 |

| | | |
|---|---|---|
| 600 | 3326 (0CFE) | 0 |
| 300 | 6654 (19FE) | 0 |
| 150 | 13310 (33FE) | 0 |
| 134.5 | 14844 (39FC) | .0007% |
| 110 | 18151 (46E7) | .00015% |
| 75 | 26622 (67FE) | 0 |
| 50 | 39934 (98FE) | 0 |

### 3.6.3.6 Digital Phase-Locked Loop

The ESCC contains a digital phase-locked loop (DPLL) to recover clock information from a data stream with NRZI or FM encoding. The DPLL is driven by a clock that is nominally 32 (NRZI) or 16 (FM) times the data rate. The DPLL uses this clock, along with the data stream, to construct a clock for the data. This clock may then be used as the ESCC receive clock, the transmit clock, or both.

For NRZI encoding, the DPLL counts the 32X clock to create nominal bit times. As the 32X clock is counted, the DPLL is searching the incoming data stream for edges (either 1/0 or 0/1). As long as no transitions are detected, the DPLL output will be free running and its input clock source will be divided by 32, producing an output clock without any phase jitter. Upon detecting a transition the DPLL will adjust its clock output (during the next counting cycle) by adding or subtracting a count of 1, thus producing a terminal count closer to the center of the bit cell. The adding or subtracting of a count of 1 will produce a phase jitter of +/- 63 on the output of the DPLL. Because the ESCC's DPLL uses both edges of the incoming signal to compare with its clock source, the mark-space ratio (50%) of the incoming signal should not deviate by more than ±1.5% if proper locking is to occur.

For FM encoding, the DPLL still counts from 0 to 31, but with a cycle corresponding to two bit times. When the DPLL is locked, the clock edges in the data stream should occur between counts 15 and 16 and between counts 31 and 0. The DPLL looks for edges only during a time centered on the 15/16 counting transition.

The 32X clock for the DPLL can be programmed to come from either the RTxC input or the output of the baud rate generator. The DPLL output may be programmed to be echoed out of the ESCC via the TRxC pin (if this pin is not being used as an input).

### 3.6.3.7 Crystal Oscillator

When using a crystal oscillator to supply the receive or transmit clocks to a channel of the ESCC, the user should:

1. Select a crystal oscillator which satisfies the following specifications:
   - 30 ppm @ 25C
   - 50 ppm over temperature of -20 to 70C
   - 5 ppm/year aging

- 5 mW drive level

2. Place crystal across RTxC and SYNC pins
3. Place 30 pF capacitors to ground from both RTxC and SYNC pins
4. Set bit D7 of WR11 to "1".

### 3.6.3.8 Data Encoding

The ESCC may be programmed to encode and decode the serial data in four different ways (Figure C.6). In NRZ encoding, a "1" is represented by a High level, and a "O"is represented by a Low level. In NRZI encoding, a "1" is represented by no change in level, and a FM1 is represented by a change in level. In FM1 (more properly, bi-phase mark), a transition occurs at the beginning of every bit cell A "1" is represented by an additional transition at the center of the bit cell, and a "O" is represented by no additional transition at the center of the bit cell. In FMo (bi-phase space), a transition occurs at the beginning of every bit cell. A "0" is represented by an additional transition at the center of the bit cell, and a "1" is represented by no additional transition at the center of the bit cell. In addition to these four methods, the ESCC can be used to decode Manchester (bi-phase level) data by using the DPLL in the FM mode and programming the receiver for NRZ data. Manchester encoding always produces a transition at the center of the bit cell. If the transition is 0/1, the bit is a "O" If the transition is 1/0, the bit is a "1".

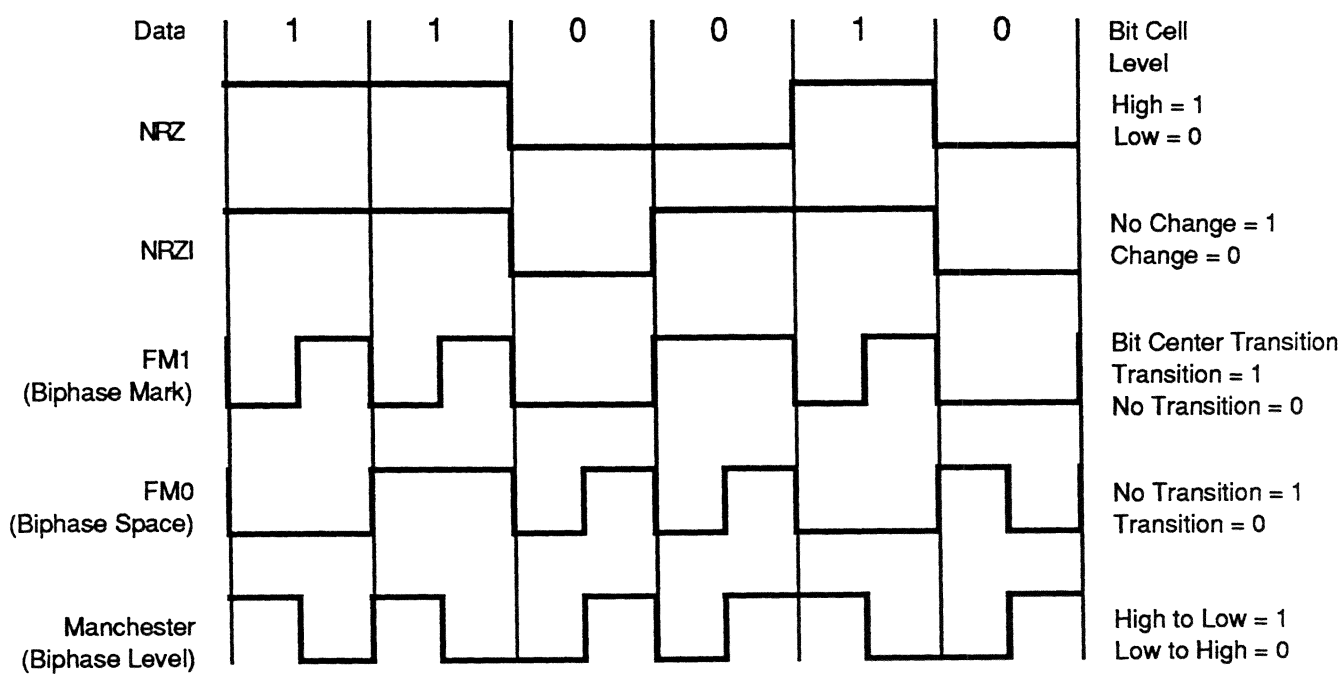



Figure C.6

### 3.6.3.9 Auto Echo and Local Loopback

The ESCC is capable of automatically echoing everything it receives. This feature is useful mainly in asynchronous modes but works in SYNC and SDLC modes as well. In Auto Echo mode, TxD is internally connected to RxD. Auto Echo mode can be used with NRZI or FM encoding with no additional delay, because the data stream is not decoded before retransmission. In this mode, the transmitter is actually bypassed, and the programmer is responsible for disabling transmitter interrupts and WAIT/ REQUEST on transmit.

The ESCC is also capable of Local Loopback. In this mode, TxD is internally connected to RxD just as in Auto Echo mode. However, in Local Loopback mode, the internal transmit data is tied to the internal receive data, and RxD is ignored (except to be echoed out via TxD). The DCD input is also ignored as transmit and receive enables. However, transitions on these inputs can still cause interrupts. Local Loopback works in asynchronous, SYNC and SDLC modes with NRZ, NRZI or FM coding of the data stream.

### 3.6.3.10 I/O Interface Capabilities

The ESCC offers the choice of Polling, Interrupt (vectored or nonvectored), and Block Transfer modes to transfer data, status, and control information to and from the CPU. The Block Transfer mode can be implemented under CPU or DMA control.

### 3.6.3.11 Polling

All interrupts are disabled. Three status registers in the ESCC are automatically updated whenever any function is performed. For example, end-of-frame in SDLC mode sets a bit in one of these status registers. The idea behind polling is for the CPU to periodically read a status register until the register contents indicate the need for data to be transferred. Only one register needs to be read; depending on its contents, the CPU either writes data, reads data, or continues. Two bits in the register indicate the need for data transfer. An alternative is a poll of the Interrupt Pending register to determine the source of an interrupt. The status for both channels resides in one register.

### 3.6.3.12 Interrupts

When an ESCC responds to an Interrupt Acknowledge signal (INTACK#) from the CPU, an interrupt vector may be placed on the data bus. This vector is written in WR2 and may be read in RR2A or RR2B (Figures C.8 and C.9).

To speed interrupt response time, the ESCC can modify three bits in this vector to indicate status. If the vector is read in Channel A, status is never included; if it is read in Channel B, status is always included.

Each of the six sources of interrupts in the ESCC (Transmit, Receive and External/Status interrupts in both channels) has three bits associated with the interrupt source: Interrupt Pending (IP), Interrupt Under Service (IUS), and Interrupt Enable (IE). Operation of the IE bit is straight forward. It the IE bit is set for a given interrupt source, then that source can request interrupts. The exception is when the MIE (Master Interrupt Enable) bit in WR9 is reset and no interrupts may be requested. The IE bits are write-only.

In the ESCC, the IP bit signals a need for interrupt servicing. When an IP bit is set to "1", the INT output is pulled Low, requesting an interrupt. In the ESCC, if the IE bit is set for an interrupt, then the IP for that source can never be set. The IP bits are readable in RR3A.

There are three types of interrupts: Transmit, Receive and External/Status. Each interrupt type is enabled under program control with Channel A having higher priority than Channel B, and with Receive, Transmit and External/Status interrupts prioritized in that order within each channel. When the Transmit interrupt is enabled, the CPU is interrupted when the transmit buffer becomes empty. (This implies that the transmitter must have had a data character written into it so that it can become empty.) When enabled, the Receive can interrupt the CPU in one of three ways:

- Interrupt on First Receive Character or Special Receive condition
- Interrupt on all Receive Characters or Special Receive condition
- Interrupt on Special Receive condition only.

Interrupt on First Character or Special Condition and Interrupt on Special Condition Only are typically used with the Block Transfer mode. A Special Receive Condition is one of the following: receiver overrun, framing error in asynchronous mode, end-of-frame in SDLC mode, and optionally, a parity error. The Special Receive Condition interrupt is different from an ordinary Receive Character Available interrupt only in the status placed in the vector during the Interrupt Acknowledge cycle. In Interrupt on First Receive Character, an interrupt can occur from Special Receive Conditions any time after the first Receive Character Interrupt.

The main function of the External/Status interrupt is to monitor the signal transitions of the DCD, and SYNCA pins; however, an External/Status interrupt is also caused by a Transmit Underrun condition a zero count in the baud rate generator, the detection of a Break (asynchronous mode), Abort (SDLC mode) or EOP (SDLC Loop mode) sequence in the data stream. The interrupt caused by the Abort or EOP has a special feature allowing the ESCC to interrupt when the Abort or EOP sequence is detected or terminated. This feature facilitates the proper termination of the current message, correct initialization of the next message, and the accurate timing of the Abort condition in external logic in SDLC mode. In SDLC Loop mode, this feature allows secondary status to recognize the wishes of the primary station to regain control of the loop during a poll sequence.

### 3.6.3.13 CPU/DMA Block Transfer

The ESCC provides a Block Transfer mode to accommodate CPU block transfer functions and DMA controllers. The Block Transfer mode uses the WAIT/REQUEST output in conjuction with the Wait/Request bits in WR1. The WAIT/REQUEST output can be defined under software control as a WAIT line in the CPU Block Transfer mode or as a REQUEST line in the DMA Block Transfer mode.

To a DMA controller, the ESCC REQUEST output indicates that the ESCC is ready to transfer data to or from memory. To the CPU, the WAIT line indicates that the ESCC is not ready to transfer data, thereby requesting that the CPU extend the I/O cycle. The DTR/REQUEST can be used as the transmit request line, thus allowing full-duplex operation under DMA control.

### 3.6.4 Programming Information

Each channel has fifteen Write registers that are individually programmed from the system bus to configure the functional personality of each channel. Each channel also has eight Read registers from which the system can read Status, Baud rate, or Interrupt information.

On the ESCC, only four data registers (Read and Write for Channels A and B) are directly selected by a High on the D/C# input and the appropriate levels on the RD8#, WR8#, and A/B# pins. All other registers are addressed indirectly by the content of Write Register 0 in conjunction with a Low on the D/C# input and the appropriate levels on the RD8#, WR8#, and A/B# pins. If bit D3 in WR0 is 1 and bits 5 and 6 are 0, then bits 0, 1 and 2 address the higher registers 8 through 15. If bits 4, 5, and 6 contain a diffent code, bits 0, 1, and 2 address the lower registers 0 through 7 as shown in Table C.7.

Writing to or reading from any register except RR0, WR0, and the data registers thus involves two operations:

First, write the appropriate code into WR0, then follow this by a Write or Read operation on the register thus specified. Bits 0 through 4 in WR0 are automatically cleared after this operation, so that WR0 then points to WR0 or RR0 again.

Channel A/Channel B selection is made using the A/B# input in conjunction with the ESCC Circuit Enable input CE1#.

The system program first issues a series of commands to initialize the basic mode of operation. This is followed by other commands to qualify conditions within the selected mode. For example, the asynchronous mode, character length, clock rate, number of stop bits, even or odd parity might be set first. Then the interrupt mode would be set and, finally, receiver or transmitter enable.

| E/N # | D/C# | "Point High" Code in WR0: | D2, | D1, | D0 In WR0: | Write Register | Read Register |
|---|---|---|---|---|---|---|---|
| Low | High | Either Way | x | X | X | Data | Data |
| Low | Low | Not True | 0 | 0 | 0 | 0 | 0 |
| Low | Low | Not True | 0 | 0 | 1 | 1 | 1 |
| Low | Low | Not True | 0 | 1 | 0 | 2 | 2 |
| Low | Low | Not True | 0 | 1 | 1 | 3 | 3 |
| Low | Low | Not True | 1 | 0 | 0 | 4 | (0) |
| Low | Low | Not True | 1 | 0 | 1 | 5 | (1) |
| Low | Low | Not True | 1 | 1 | 0 | 6 | (2) |
| Low | Low | Not True | 1 | 1 | 1 | 7 | (3) |
| Low | Low | True | 0 | 0 | 0 | Data | Data |
| Low | Low | True | 0 | 0 | 1 | 9 | - |
| Low | Low | True | 0 | 1 | 0 | 10 | 10 |
| Low | Low | True | 0 | 1 | 1 | 11 | (15) |
| Low | Low | True | 1 | 0 | 0 | 12 | 12 |
| Low | Low | True | 1 | 0 | 1 | 13 | 13 |
| Low | Low | True | 1 | 1 | 0 | 14 | (10) |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| Low | Low | True | 1 | 1 | 1 | 15 | 15 |
| High | Low | Not True | 0 | 1 | 0 | Enhancement register. | |
| High | Low | Not True | 0 | 1 | 1 | VERSION register. | |

**Table C.7. Register Addressing**

### 3.6.4.1 Read Registers

The ESCC contains eight read registers (actually nine, counting the receive buffer (RR8) in each channel). Four of these may be read to obtain status information (RRO, RR1, RR10, and RR15). Two registers (RR12 and RR13) may be read to learn the baud rate generator time constant. RR2 contains either the unmodified interrupt vector (Channel A) or the vector modified by status information (Channel B). RR3 contains the Interrupt Pending (IP) bits (Channel A). In addition, ff bit D2 of WR15 is set RR6 and RR7 are available for providing frame status from the 10 x 19 bit Frame Status FIFO. Figure C.8 shows the formats for each read register.

The status bits of RRO and RR1 are carefully grouped to simplify status monitoring, for example, when the interrupt vector indicates a Special Receive Condition interrupt, all the appropriate error bits can be read from a single register(RR1).

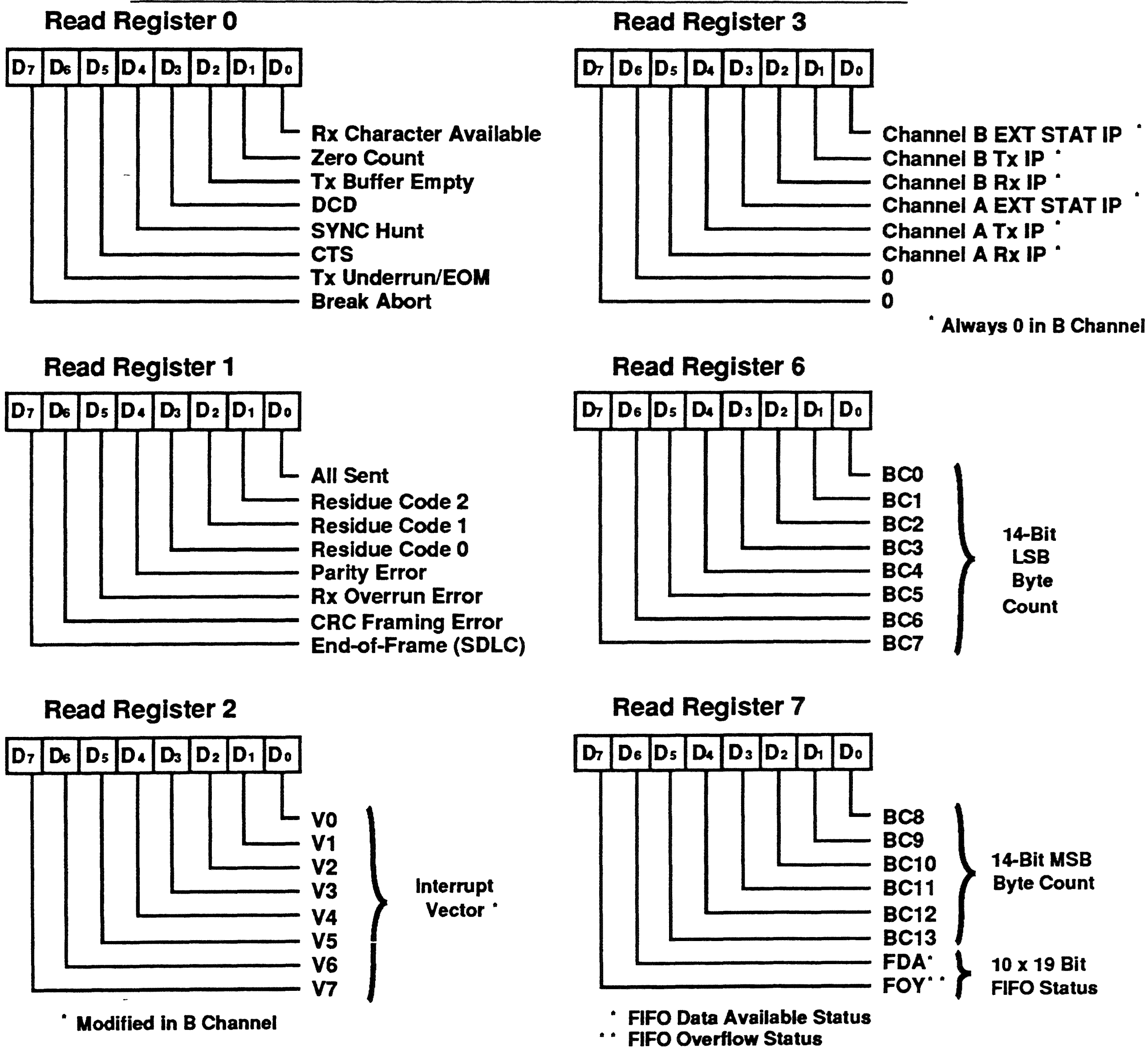



Figure C.8. Read Register Bit Functions

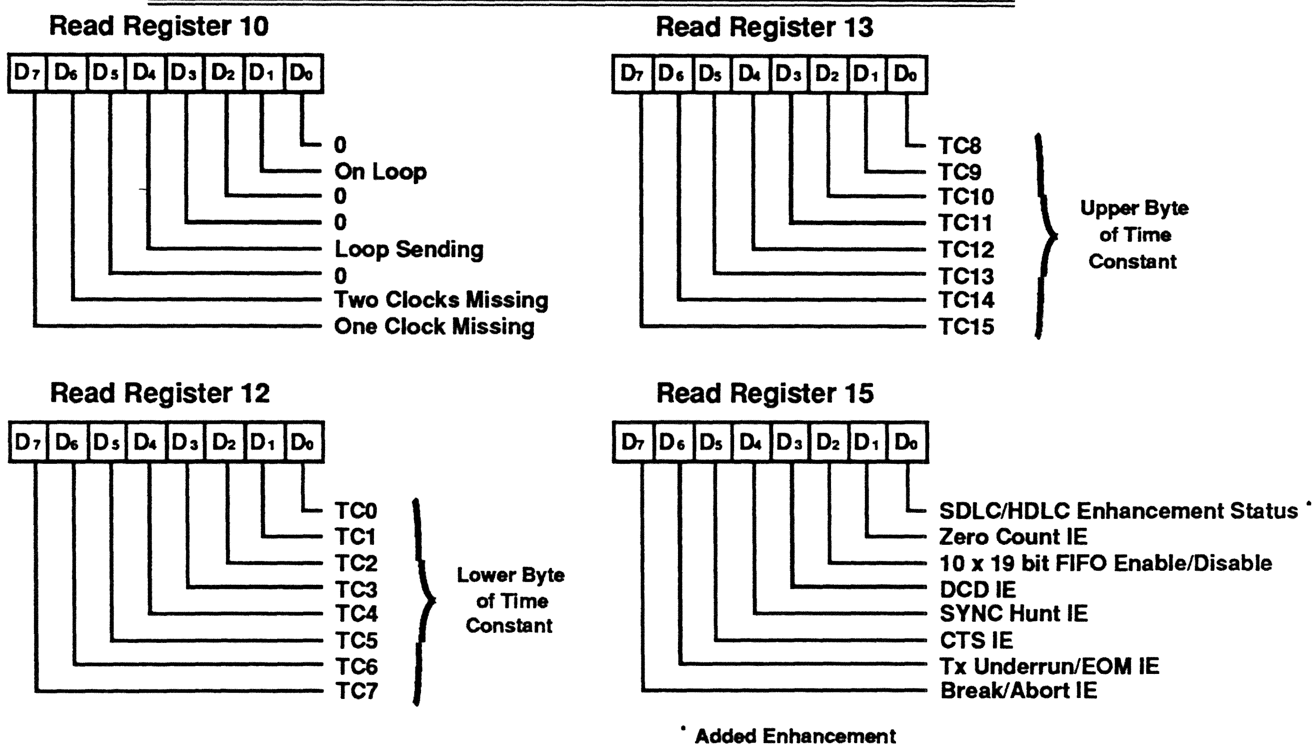



Figure C.8. Read Register Bit Functions (continued)

### 3.6.4.2 Write Registers

The ESCC contains 15 write registers (16 counting WR8, the transmit buffer) in each channel. These write registers are programmed separately to configure the functional "personality" of the channels. Two registers (WR2 and WR9) are shared by the two channels that can be accessed through either of them. WR2 contains the interrupt vector for both channels, while WR9 contains the interrupt control bits. In addition, if bit DO of WR15 is set, executing register seven prime (WR7') is available for programming additional SDLC/HDLC enhancements. When bit DO of WR15 is set, executing a write to WR7 actually writes to WR7' to further enhance the functional "personality" of each channel. Figure C.9 shows the format of each write register.

## Write Register 0

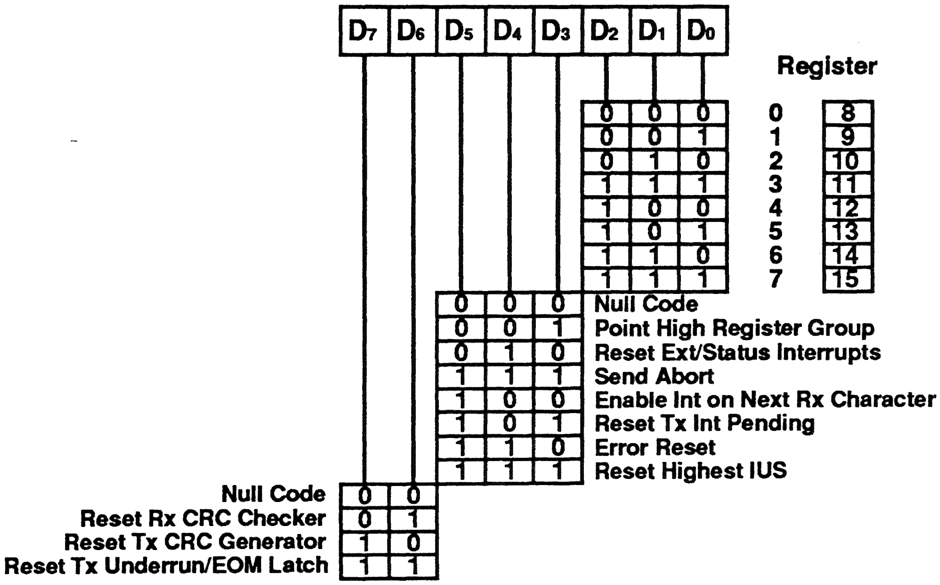
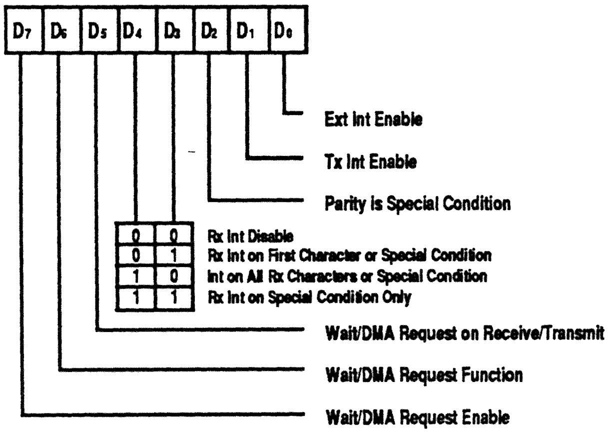
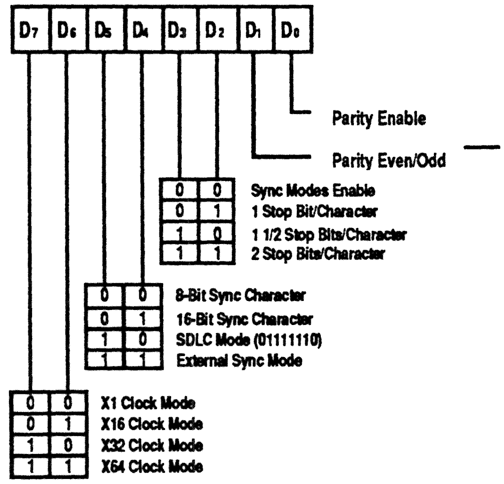



Figure C.9. Write Register Bit Functions

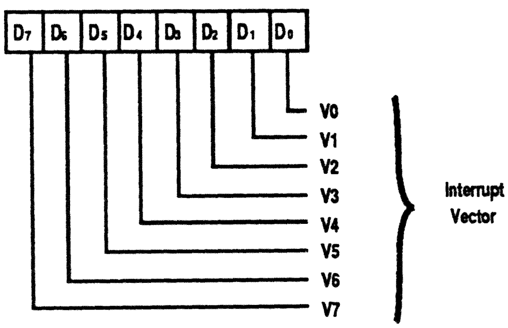
Write Register 1



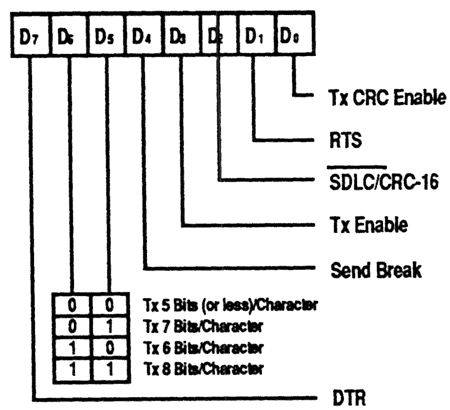

Write Register 4




Read Register 2




Read Register 5




Read Register 3

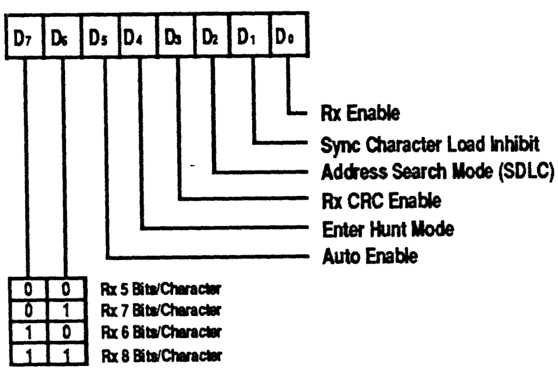



Figure C.9. Write Register Bit Functions (continued)

**Write Register 6**

| $D_7$ | $D_6$ | $D_5$ | $D_4$ | $D_3$ | $D_2$ | $D_1$ | $D_0$ | |
|---|---|---|---|---|---|---|---|---|
| $SYNC_7$ | $SYNC_6$ | $SYNC_5$ | $SYNC_4$ | $SYNC_3$ | $SYNC_2$ | $SYNC_1$ | $SYNC_0$ | Monosync 8 Bits |
| $SYNC_1$ | $SYNC_0$ | $SYNC_5$ | $SYNC_4$ | $SYNC_3$ | $SYNC_2$ | $SYNC_1$ | $SYNC_0$ | Monosync 8 Bits |
| $SYNC_7$ | $SYNC_6$ | $SYNC_5$ | $SYNC_4$ | $SYNC_3$ | $SYNC_2$ | $SYNC_1$ | $SYNC_0$ | Bisync 16 Bits |
| $SYNC_3$ | $SYNC_2$ | $SYNC_1$ | $SYNC_0$ | 1 | 1 | 1 | 1 | Bisync 12 Bits |
| $ADR_7$ | $ADR_6$ | $ADR_5$ | $ADR_4$ | $ADR_3$ | $ADR_2$ | $ADR_1$ | $ADR_0$ | SDLC |
| $ADR_7$ | $ADR_6$ | $ADR_5$ | $ADR_4$ | 1 | 1 | 1 | 1 | SDLC (Address 0) |

Figure C.9. Write Register Bit Functions (continued)

### Write Register 7

| $D_7$ | $D_6$ | $D_5$ | $D_4$ | $D_3$ | $D_2$ | $D_1$ | $D_0$ | |
|---|---|---|---|---|---|---|---|---|
| $SYNC_7$ | $SYNC_6$ | $SYNC_5$ | $SYNC_4$ | $SYNC_3$ | $SYNC_2$ | $SYNC_1$ | $SYNC_0$ | Monosync 8 Bits |
| $SYNC_5$ | $SYNC_4$ | $SYNC_3$ | $SYNC_2$ | $SYNC_1$ | $SYNC_0$ | 1 | 1 | Monosync 8 Bits |
| $SYNC_5$ | $SYNC_{14}$ | $SYNC_{13}$ | $SYNC_{12}$ | $SYNC_{11}$ | $SYNC_{10}$ | $SYNC_9$ | $SYNC_8$ | Bisync 16 Bits |
| $SYNC_{11}$ | $SYNC_{10}$ | $SYNC_9$ | $SYNC_8$ | $SYNC_7$ | $SYNC_6$ | $SYNC_5$ | $SYNC_4$ | Bisync 12 Bits |
| 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | SDLC |

### Write Register 7'

| Bit | Function |
|---|---|
| $D_0$ | Auto Tx Flag |
| $D_1$ | Auto EOM Latch Reset |
| $D_2$ | Auto RTS |
| $D_3$ | TxD Pulled High in SDLC NRZI Mode |
| $D_4$ | Fast DTR/REQ Mode |
| $D_5$ | CRC Check Bytes Completely Received |
| $D_6$ | Extended Read Enable |
| $D_7$ | Must Be Set to 0 |

Figure C.9. Write Register Bit Functions (continued)

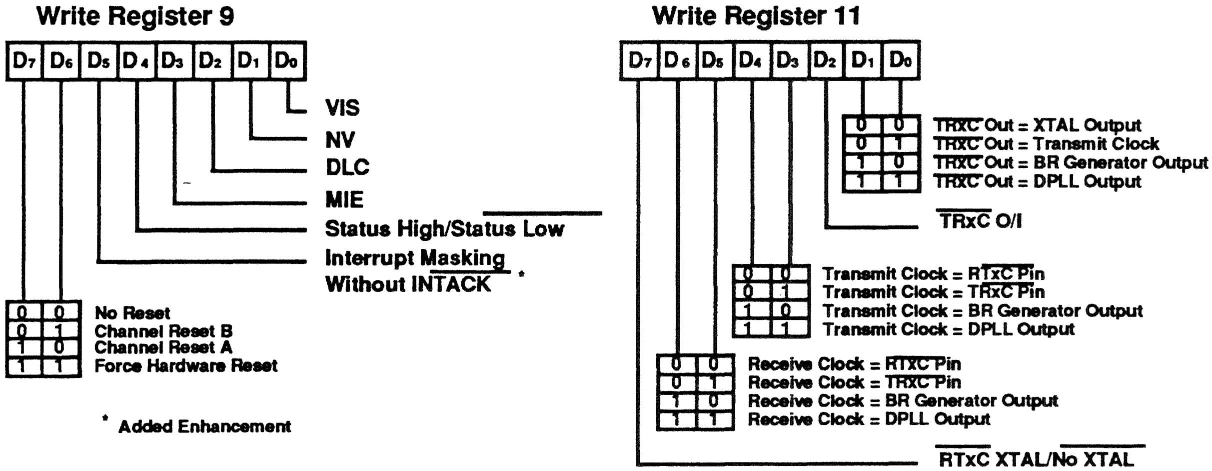



Figure C.9. Write Register Bit Functions (continued)

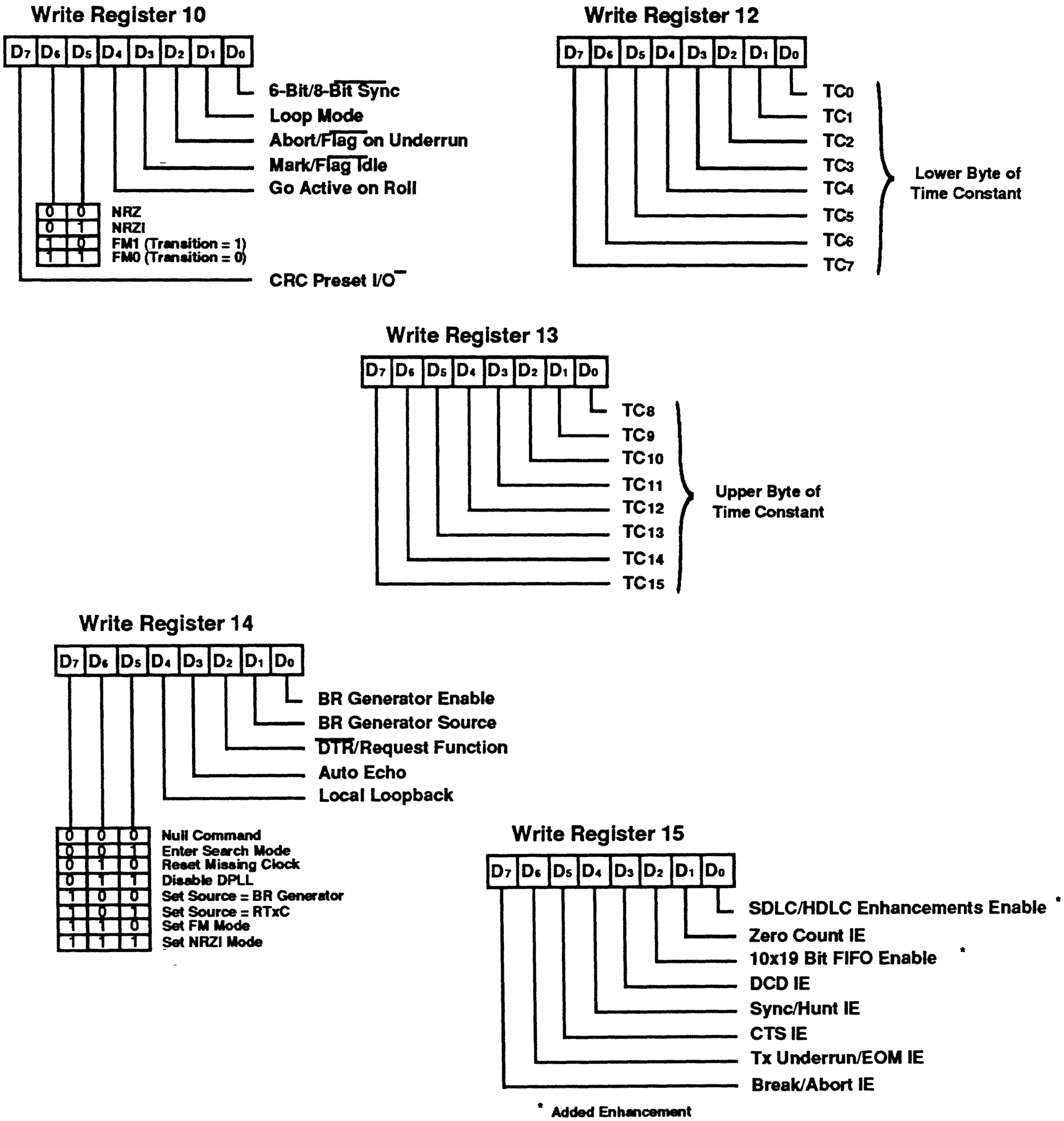



Figure C.9

### 3.6.5 ESCC Timing

The ESCC generates internal control signals from WR8# and RD8# that are related to PCLK. Since PCLK has no phase relationship with WR8# and RD8# the circuitry generating these internal control signals must provide time for metastable conditions to disappear. This gives rise to a recovery time related to PCLK. The recovery time applies only between bus transactions involving the ESCC. The recovery time required for proper operation is specified from the falling edge of WR8# or RD8# in the first transaction involving the ESCC, to the falling edge of WR8# or RD8# in the second transaction involving the ESCC. This time must be at least 3 and 1/2 PCLK regardless of which register or channel is being accessed.

#### 3.6.5.1 Read Cycle Timing

Figure C.10 illustrates Read cycle timing. Addresses on A/B# and D/C# and the status on INTACK# must remain stable throughout the cycle. If CE1# falls after RD8# falls or if it rises before RD8# rises, the effective RD8# is shortened.

#### 3.6.5.2 Write Cycle Timing

Figure C.11 illustrates Write Cycle timing. Addresses on A/B# and D/C# and the status on INTACK# must remain stable throughout the cycle. If CE1# falls after WR8# falls or if it rises before WR8# rises, the effective WR8# is shortened. Data must be valid before the rising edge of WR8#.

#### 3.6.5.3 Interrupt Acknowledge Cycle Timing

Figure C.12 illustrates Interrupt Acknowledge cycle timing. The ESCC may be programmed to respond to RD8# Low by placing its interrupt vector on DATAB[7:0] and it then sets the appropriate Interrupt-Under-Service latch internally.

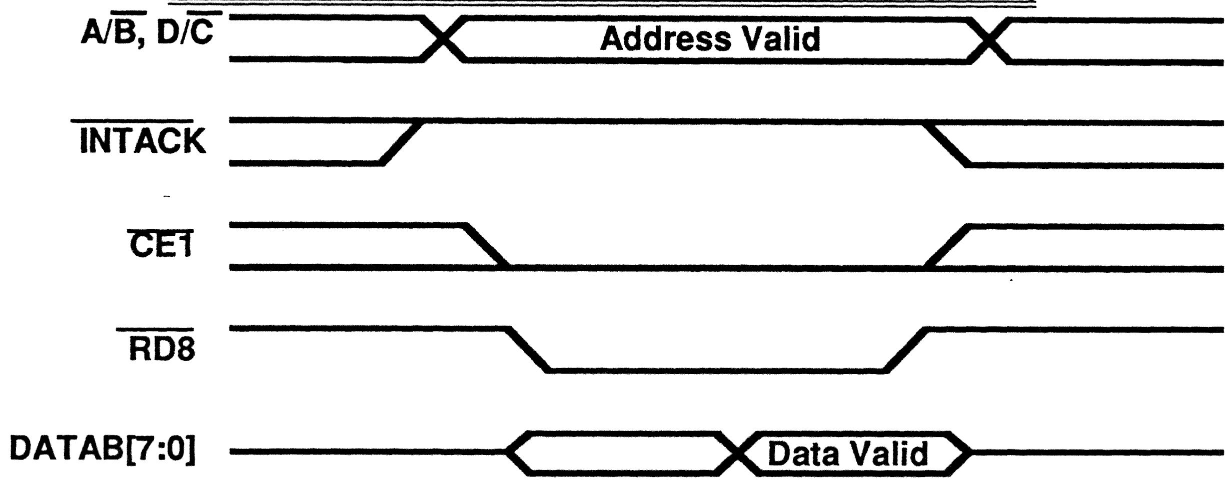



Figure C.10 READ CYCLE TIMING

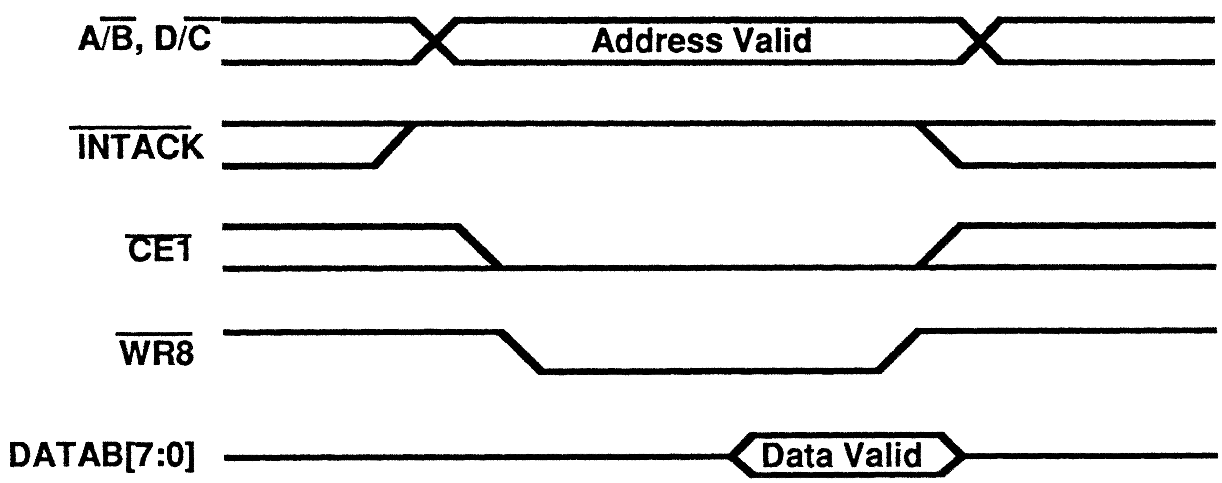



Figure C.11 WRITE CYCLE TIMING

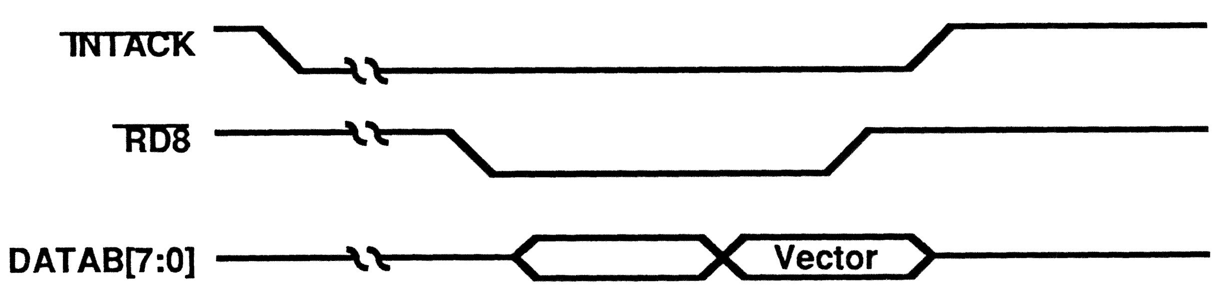



Figure C.12 INTERRUPT ACKNOWLEDGE CYCLE TIMING

### 3.6.6 Status FIFO Enhancements

When used with a DMA controller, the ESCC Frame Status FIFO enhancement maximizes the ESCC's ability to receive high-speed back-to-back SDLC messages while minimizing frame overruns due to CPU latencies in responding to interrupts.

Additional logic was added to the NMOS Z8530 consisting of a 10-deep by 19-bit status FIFO, a 14-bit receive byte counter, and control logic as shown in Figure C.13. The 10 x 19 bit status FIFO is separate from the existing three-byte receive data and Error FIFOs.
When the enhancement is enabled, the status in Read Register 1 (RR1) and byte count for the SDLC frame will be stored in the 10 x 19 bit status FIFO. This allows the DMA controller to transfer the next frame into memory while the CPU verifies that the message was properly received.

Summarizing the operation, data is received, assembled, and loaded into the three-byte receive FIFO before being transferred to memory by the DMA controller. When a flag is received at the end of an SDLC frame, the frame byte count from the 14 bit counter and five status bits are loaded into the status FIFO for verification by the CPU. The CRC checker is automatically reset in preparation for the next frame which can begin immediately. Since the byte count and status are saved for each frame, the message integrity can be verified at a later time. Status information for up to 10 frames can be stored before a status FIFO overrun could occur.

If receive interrupts are enabled while the 10 x 19 FIFO is enabled, an SDLC end-of-frame special condition will not lock the three-byte Receive data FIFO. An SDLC end-of-frame still locks the three-byte Receive data FIFO in "Interrupt on first Receive Character" or" Special Condition" and "Interrupt on Special Condition Only" modes when the 10 x 19 FIFO is disabled. This feature allows the 10 x 19 SDLC FIFO to accept multiple SDLC frames without CPU intervention at the end of each frame.

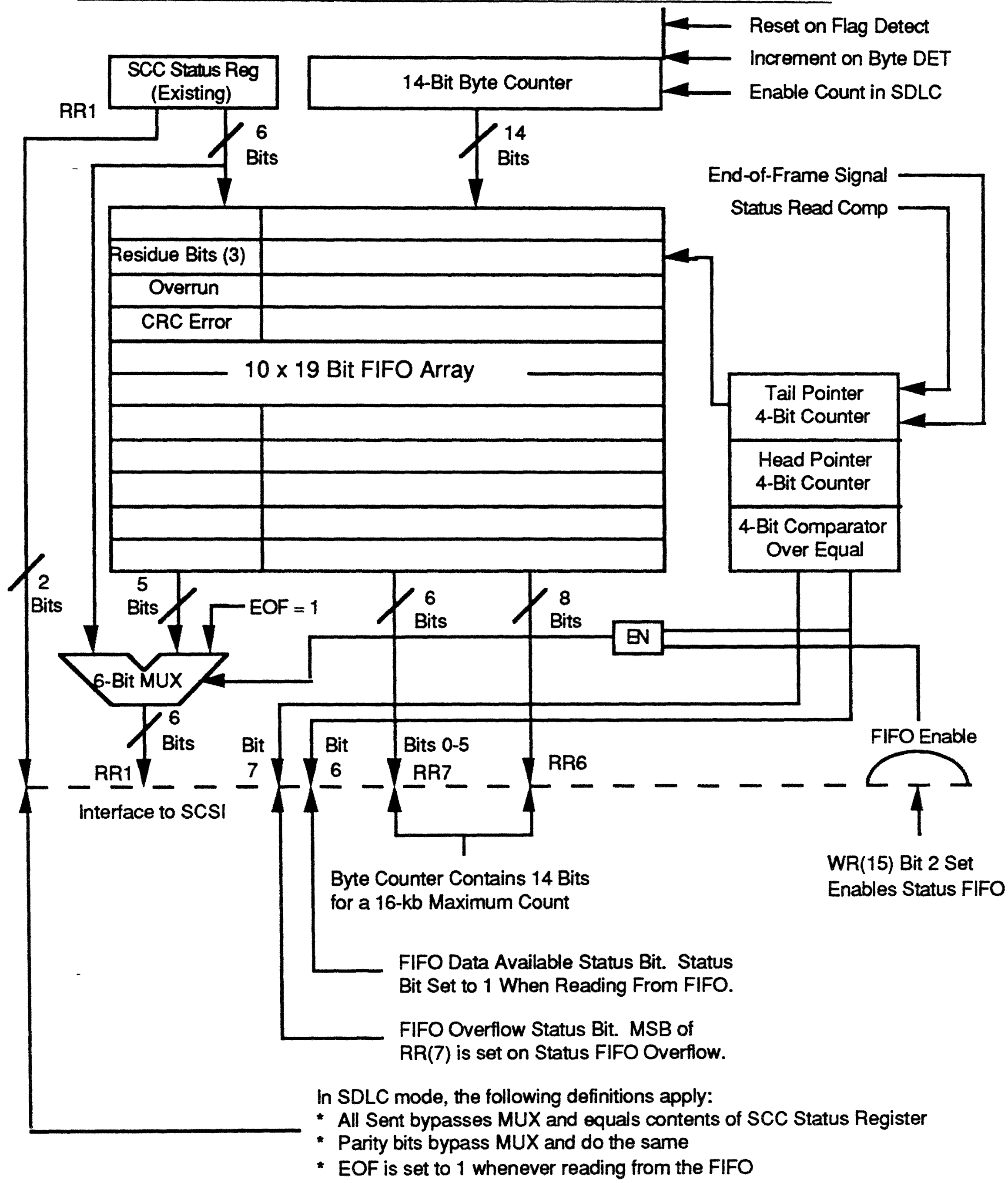



Figure C.13

### 3.6.6.1 FIFO Detail

For a better understanding of details of the FIFO operation, refer to the block diagram contained in Figure C.13.

### 3.6.6.2 Enable/Disable

This FIFO is implemented so that it is enabled when WR15 bit 2 is set and the ESCC is in the SDLC/HDLC mode, otherwise the status register contents bypass the FIFO and go directly to the bus interface (the FIFO pointer logic is reset either when disabled or via a channel or power-on reset). When the FIFO mode is disabled, the ESCC is completely downward-compatible with the NMOS Z8530. The FIFO mode is disabled on power-up (WR15 bit 2 is set to "0" on reset). The effects of backword compatibility on the register set are that RR4 is an image of RRO, RR5 is an image of RR1, RR6 is an image of RR2, and RR7 is an image of RR3. For the details of the added registers, refer to Figure C.8. The status of the FIFO Enable signal can be obtained by reading RR15 bit 2. If the FIFO is enabled, the bit will be set to "1"; otherwise, it will be reset.

### 3.6.6.3 Read Operation

When WR15 bit 2 is set and the FIFO is not empty, the next read to status register RR1 or the additional registers RR7 and RR6 will actually be from the FIFO. Reading status register RR1 causes one location of the FIFO to be emptied, so status should be read after reading the byte count, otherwise the count will be incorrect. Before the FIFO underflows, it is disabled. In this case, the multiplexer is switched to allow status to be read directly from the status register, and reads from RR7 and RR6 will contain bits that are undefined. Bit 6 of RR7 (FIFO Data Available) can be used to determine if status data is coming from the FIFO or directly from the status register, since it is set to "1" whenever the FIFO is not empty.

Because not all status bits are stored in the FIFO, the All Sent, Parity, and EOF bits will bypass the FIFO. The status bits sent through the FIFO will be Residue Bits (3), Overrun, and CRC Error.

The sequence for proper operation of the byte count and FIFO logic is to read the registers in the following order, RR7, RR6, and RR1 (reading RR6 is optional). Additional logic prevents the FIFO from being emptied by multiple reads from RR1. The read from RR7 latches the FIFO empty/full status bit (bit 6) and steers the status multiplexer to read from the ESCC megacell instead of the status FIFO (since the status FIFO is empty). The read from RR1 allows an entry to be read from the FIFO (if the FIFO was empty, logic is added to prevent a FIFO underflow conditon).

### 3.6.6.4 Write Operation

When the end of an SDLC frame (EOF) has been received and the FIFO is enabled, the contents of the status and byte count registers are loaded into the FIFO. The EOF signal is used to increment the FIFO. If the FIFO overflows, the MS8 of RR7 (FIFO Overflow) is set to indicate the overflow. This bit and the FIFO control logic is reset by disabling and re-enabling the FIFO control bit (WR15 bit 2). For details of FIFO control timing during an SDLC frame, refer to the figures in the AC timing section.

### 3.6.6.5 Byte Counter Detail

The 14-bit byte counter allows for packets up to 16K bytes to be received. For a better understanding of its operation, refer to Figures C.14 and C.15.

### Enable

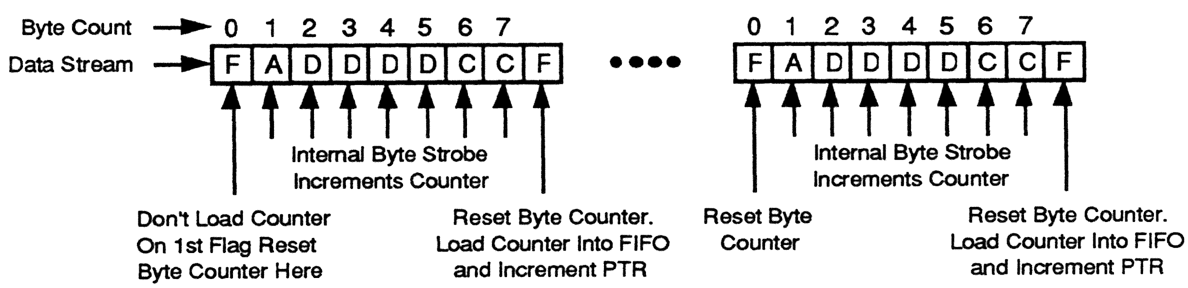
The byte counter is enabled when the ESCC is in the SDLC/HDLC mode and WR15 bit 2 Is set to "1".

### Reset

The byte counter is reset whenever an SDLC flag character is received. The reset is timed so that the contents of the byte counter are successfully written into the FIFO.

### Increment

The byte counter is incremented by writes to the data FIFO. The counter represents the number of bytes received by the ESCC, rather than the number of bytes transferred from the ESCC. (These counts may differ by up to the number of bytes in the receive data FIFO contained in the ESCC.)




Key

F : Flag
A : Address Field
D : Data
C : Control Field

Figure C.14

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| RR7 | FOY | FDA | BC 13 | BC 12 | BC 11 | BC 10 | BC 9 | BC 8 |

FDA: FIFO Data Available Status
1 = Status Reads Will Come From FIFO
0 = Status Reads Will Come From SCC

FOY: FIFO Overflow Status
1 = FIFO Overflowed During Operation
0 = Normal

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| RR7 | BC 7 | BC 6 | BC 5 | BC 4 | BC 3 | BC 2 | BC 1 | BC 0 |

Read From FIFO
LSB Byte Count

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| RR7 | • | • | • | • | • | FEN | • | ENH |

ENH: SDLC/HDLC Enhancement Status
1 = Enhancements Enabled
0 = Enhancements Disabled

FEN: Status FIFO Enable Control Bit
1 = Status and Byte Count Will Be Held
in the Status FIFO Until Read
0 = Status Will Not be Held (SCC Emulation Mode)

• = No Change from NMOS SCC DFN

Figure C.15. ESCC Additional Registers

### 3.6.7 LocalTalk Enhancements

A LocalTalk enhancement mode has been implemented into the CURIO in order to improve overall software/hardware efficiency when the ESCC is executing the LocalTalk communication protocol. When selected, this function will help relieve the control processor from extensive monitoring and timekeeping activities in relation to the ESCC.

On power up, hardware or channel resets, the ESCC will default to the normal mode. The LocalTalk enhancement mode can be selected by writing to two special control registers, ENRA and ENRB, for channels A and B respectively. These registers are both readable and writable. An additional pin, E/N# is added to provide the extra address space required to access these registers:

| E/N# | WR8# | RD8# | A/B# | D/C# | Description |
|---|---|---|---|---|---|
| 0 | x | x | x | x | Normal ESCC mode |
| 1 | 1 | 0 | 1 | 0 | Read ENRA register, WR0 address ptr = 2. |
| 1 | 1 | 0 | 0 | 0 | Read ENRB register, WR0 address ptr = 2. |
| 1 | 0 | 1 | 1 | 0 | Write ENRA register, WR0 address ptr = 2. |
| 1 | 0 | 1 | 0 | 0 | Write ENRB register, WR0 address ptr = 2. |
| 1 | 1 | 0 | x | 0 | Read VERSION register, WR0 address ptr = 3. |

#### 3.6.7.1 ENRA, ENRB Register Format

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| Res | Res | Res | Res | TXT4 | RXT4 | EFEN | LTEN |

Bit [7:4] Reserved : These bits are reserved. Writing to these bits will have no effect on the device functionality. These bits will be read as 0s.

Bit 3 TXT4 : Transmit threshold four. When the EFEN bit is on, setting the TXT4 bit will instruct the ESCC to assert DTR/REQ# and REQ# when there are four or more empty locations inside the transmit FIFO.

If TXT4 bit is reset, the ESCC will assert DTR/REQ# and REQ# when there are one or more empty locations in the transmit FIFO.

This bit is reset on hardware or channel reset.

Bit 2 RXT4 : Receive threshold four. When the EFEN bit is on, setting the RXT4 bit will instruct the ESCC to assert W#/REQ# when there are four or more receive bytes in the receive FIFO.

If the RXT4 bit is reset, the ESCC will assert W#/REQ# when there are one or more received bytes in the receive FIFO.
This bit is reset on hardware or channel reset.

Bit 1 EFEN : Extended FIFO enable. When set to 1, this bit enables the 8 deep transmit and receive FIFOs.
This bit is reset on hardware or channel reset.

Bit 0 LTEN : LocalTalk enable. If this bit is set to 1, and the SDLC mode is enabled, the ESCC will automatically start transmission with the SYNC pulse and two FLAGs, and will close transmission with the CRC, FLAG and the ABORT sequence.
This bit is reset on hardware or channel reset.

### 3.6.7.2 LocalTalk protocol format

When the LocalTalk Enable bit (LTEN) is set in the corresponding ENRA or ENRB registers, and the ESCC is running under the SDLC mode, the LocalTalk module will control the state transitions within the ESCC transmitter, thereby generating the protocol signalling format in hardware. When not running under the SDLC mode, the ESCC functionality is not affected by the state of the LocalTalk enable bits.

LocalTalk signalling as implemented in the CURIO is based on the SDLC frame format with the following changes:

i. Generate a SYNC pulse before the SDLC opening flag: It is a way of implementing collision avoidance. The sync pulse will set a bit in all listening SCCs in the network, which will in turn prevent them from transmitting. The sync pulse is 3 bits long. (Minimum requirement is 2 bits.)

ii. Generate two SDLC opening flags: It is a part of the LocalTalk packet format.

iii. Generate an abort sequence at the end of LocalTalk packets: It is required to "wake up" transmitters that were deferring transmission based on detecting the sync pulse prior to an attempt to transmit. The abort sequence will be 13 bits long. (Minimum requirement is 12 bits.)

iv. Receive disable during transmit: When the device is in the LocalTalk mode, enabling the transmitter will automatically disable the receiver.

### 3.6.7.3 ESCC enhanced features

The ESCC contains enhancements over the industry standard SCC, of which the following were used in the LocalTalk protocol. These features are offered in the ESCC normal mode as well as in the LocalTalk enhancement mode.

i. Operation with Auto Flag bit set: The LocalTalk enhancements will work with or without the auto flag bit set. In fact this bit may be regarded as a don't care when the device is running under the LocalTalk mode.

ii. Operation with Mark Idle/Flag Idle: The LocalTalk enhancements will work with the MARK IDLE bit set for either FLAG or MARK idling.

iii. Auto CRC preset: Writing to Write Register 10, bit 7 controls how the receive CRC checker, and the transmit CRC generator are preset.

iv. Auto reset of Transmit underrun latch: Setting bit 1 of the Write Register 7 will automatically reset the transmit underrun latch after the first byte is transmitted.

Notes:

1. Saving Frame Status: The ESCC currently has a FRAME STATUS FIFO (10x19) that could be used to store SDLC frame status. When enabled, this FIFO holds status information for upto 10 SDLC frames.

2. Low power standby mode: ESCC control register contents are not affected by PCLK frequency changes. In particular, stopping PCLK will not destroy contents of ESCC control registers.

### 3.6.8 Extended Transmit and Receive Data FIFOs

When the Extended FIFO Enable bit (EFEN) is set in the corresponding ENRA or ENRB registers, all transmit and receive data will be switched to pass through an extended 8 level deep FIFO.

The extended receive FIFO is connected in between the received data and the three deep receive buffer in the ESCC receiver. The combined depth is eight. Receive data bubbles through this FIFO into the receive buffer. When disabled, the receive data path will simply bypass the extended FIFO, and connect directly to the three deep receive buffer as in the normal ESCC mode.

The extended transmit FIFO sits between the CPU port and the one byte internal transmit buffer. The combined depth is eight. Data writes to the ESCC will bubble through this FIFO into the transmit buffer. When disabled, the transmit data path will bypass the extended FIFO to revert to the normal ESCC mode arrangement.

If the receive FIFO threshold is set to four, the device will assert W#/REQ# when the packet being received ends. This will alert the CPU that there is data to be read from the receive FIFO, even if there are less than four bytes in it. (Furthermore, the W#/REQ# will stay asserted until all data bytes are read from the receive FIFO.)

### 3.7 SBP INTERFACE FUNCTIONAL DESCRIPTION

The SBP Interface provides a direct interface to the Serial Bus Port of the Am79C30A ISDN Digital Subscriber Controller. Its function is to provide a connection path between the Am79C30A serial channel with the CURIO processor bus, the two internal ESCC channels or the two external SCC channels. Serial data from the Am79C30A follows a frame format which can be subdivided into three independent 64Kbps channels. These three 8-bit channels, designated as Bd, Be and Bf, are intended to carry the following combinations of digitized voice and data:

Bd, Be : contain any combinations of voice and data.
Bf : Contains only digitized voice.

The SBP programmable paths are designed to pass voice transparently to and from the 8-bit bus interface, while passing data to and from the internal or external ESCC SDLC channels.

### 3.7.1 SBP Interface Block Diagram

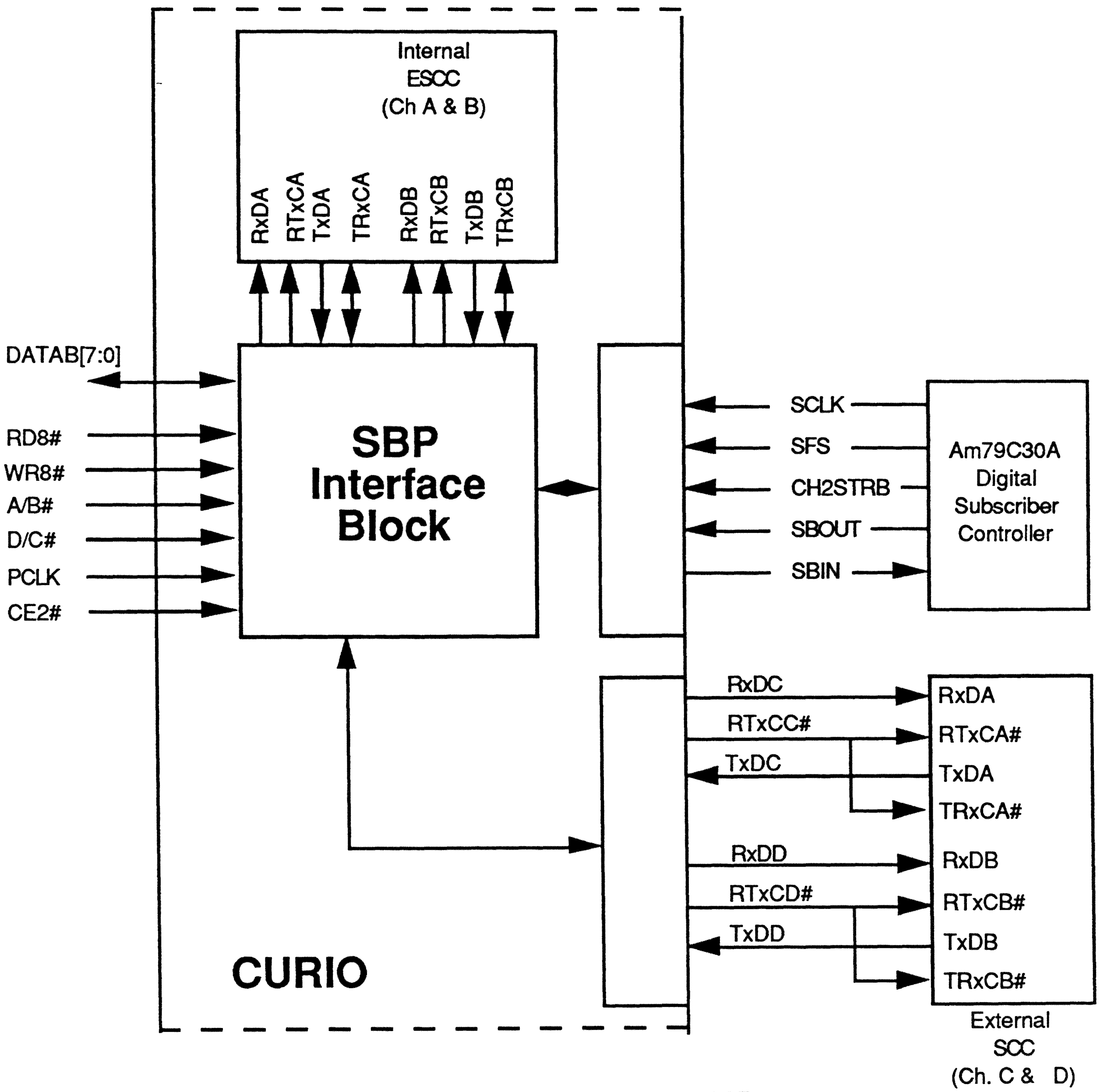



**Figure D.1 SERIAL BUS PORT INTERFACE**

### 3.7.2 SBP Interface Data Path

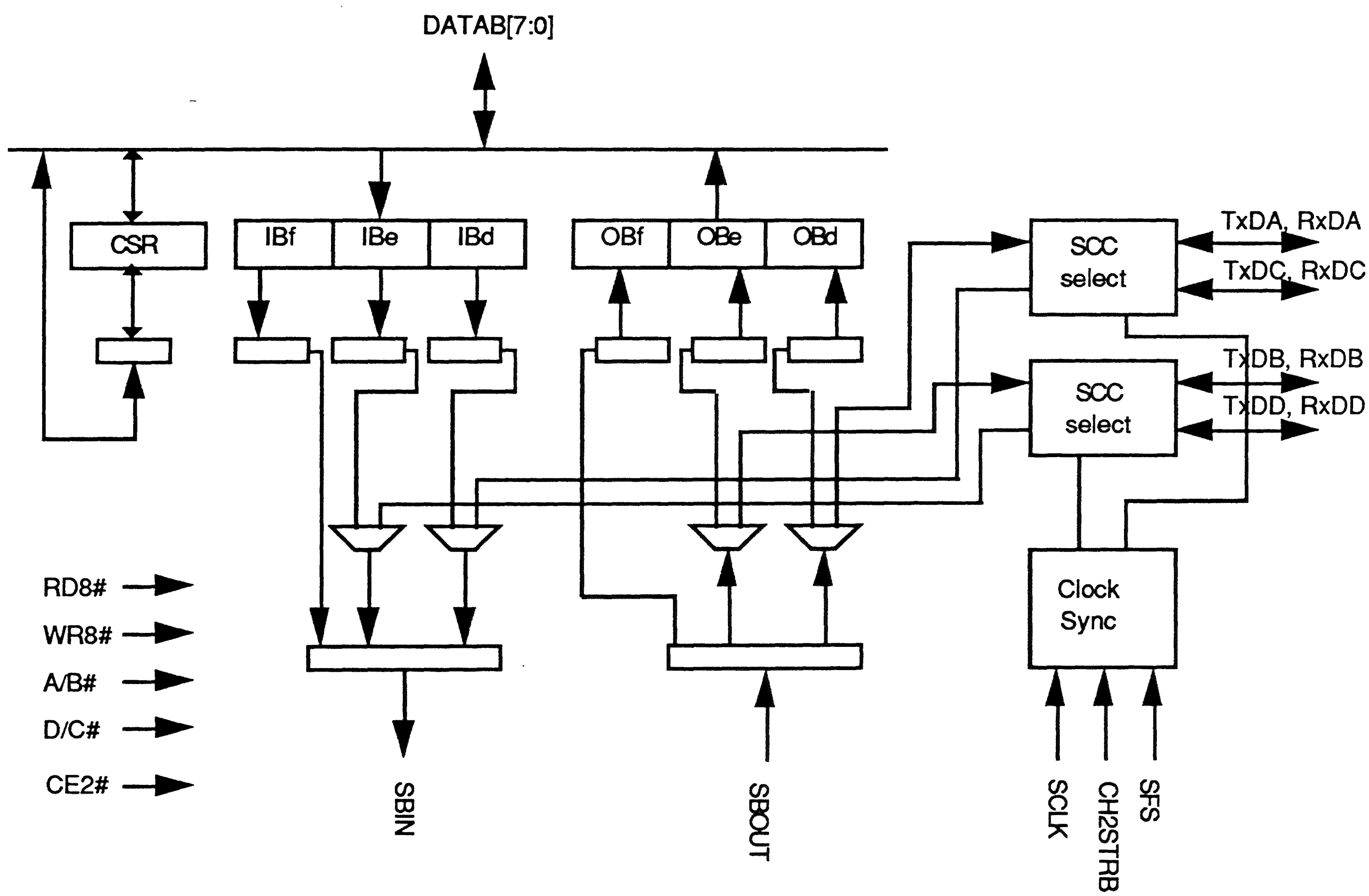



**Figure D.2 SBPI INTERNAL DATA PATH**

### 3.7.3 SBP Interface Detail Functional Description

The signals from the Am79C30A are as illustrated in figure D.3 below:

| | |
|---|---|
| SCLK | : serial clock (192KHz). |
| SFS | : serial frame sync. |
| CH2STRB | : second byte strobe. |
| SBOUT | : serial synchronous data out (MSB first). |
| SBIN | : serial synchronous data in (MSB first). |

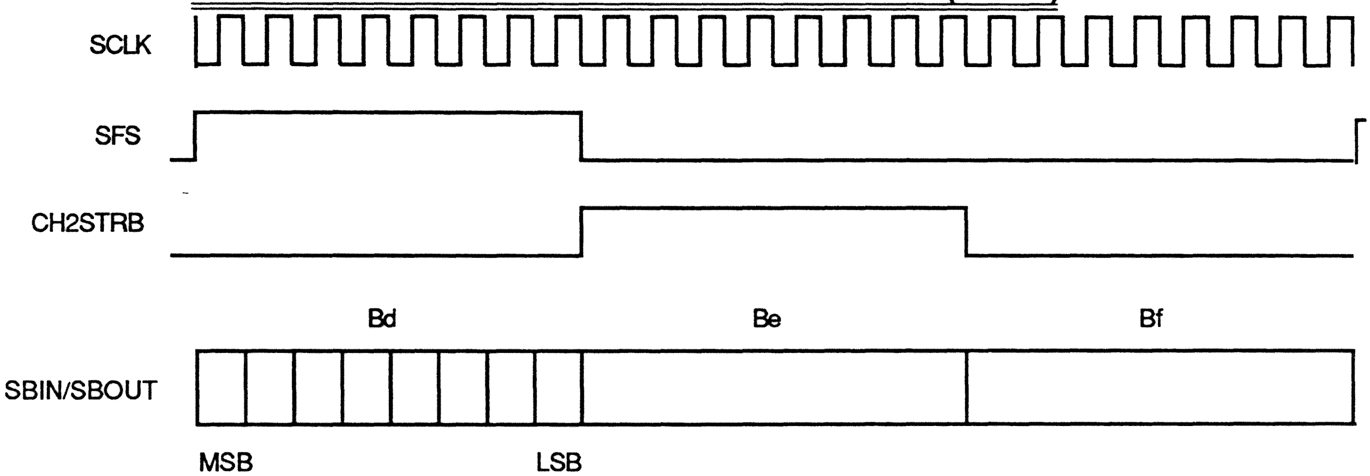



Figure D.3

When the CURIO is receiving data from the Am79C30A, the SBP interface byte packs and sends Bd, Be channels either to the 8-bit bus or routes the data directly to the internal or external ESCCs. Data from the Bf channel is always packed and sent to the bus interface port. The converse operation is performed when the CURIO is sending data to the Am79C30A. The routing path is determined by the contents of the Command/Status Register (CSR).

### 3.7.3.1 Address Table

There are seven user accessible 8-bit registers inside the SBP Interface subsection:

Command/Status Register, CSR : (Read/Write) Stores buffer status and routing commands.
Output Bd buffer, OBd : (Read only) Stores the incoming Bd channel from SBP.
Output Be buffer, OBe : (Read only) Stores the incoming Be channel from SBP.
Output Bf buffer, OBf : (Read only) Stores the incoming Bf channel from SBP.
Input Bd buffer, IBd : (Write only) Stores the outgoing Bd channel to SBP.
Input Be buffer, IBe : (Write only) Stores the outgoing Be channel to SBP.
Input Bf buffer, IBf : (Write only) Stores the outgoing Bf channel to SBP.

The MSB data occurs at the first time slot of SCLK at the serial interface. At the 8-bit data bus the Bd, Be, Bf are written or read in the byte sequence shown below.

```
SBP interface ------>| Bf |---| Be |---| Bd |--------> to SBIN
              <------| Bd |---| Be |---| Bf |<------- from SBOUT
```

Data transfer synchronization is achieved through a two stage clocking scheme based on PCLK. As a result, any read or write to the CSR and the six internal registers are only guaranteed to be completed after four PCLK periods from the rising edge of SFS (which normally runs at 8KHz).

Register Addressing Table:

| CE2# | WR8# | RD8# | A/B# | D/C# | Description |
|---|---|---|---|---|---|
| 0 | 1 | 0 | 0 | 0 | Read CSR |
| 0 | 1 | 0 | 0 | 1 | Read OBd register |
| 0 | 1 | 0 | 1 | 0 | Read OBe register |
| 0 | 1 | 0 | 1 | 1 | Read OBf register |
| 0 | 0 | 1 | 0 | 0 | Write CSR |
| 0 | 0 | 1 | 0 | 1 | Write IBd register |
| 0 | 0 | 1 | 1 | 0 | Write IBe register |
| 0 | 0 | 1 | 1 | 1 | Write IBf register |
| x | 0 | 0 | x | x | Reset CSR |
| 1 | x | x | x | x | No Operation |

Table D.4

### 3.7.3.2 Command/Status Register (CSR) Format

This register contains the buffer status and Bd, Be channel routing commands:

| EN | IDBF | ODBF | INT_SCC | IBD | IBE | OBD | OBE |
|---|---|---|---|---|---|---|---|

Bit 7 ENable : Enable SBIN output by setting this bit to 1. Tristate SBIN output by resetting this bit to 0.

Bit 6 IDBF : Set to 1 if any one of the 3 input data buffers has been written. Reset to 0 after the full 24 bit word has completed its transfer to the input shift register.

Bit 5 ODBF : Set to 1 after the full 24 bit word has been transferred from the output shift register to the output data buffers. Reset to 0 if **ANY** one of the 3 output data buffers has been read.

Bit 4 INT_SCC : 0 if external SCC (Channel C, D) is connected to ISDN SBP port.
1 if internal SCC (Channel A, B) is connected to ISDN SBP port.

Bit 3 IBD : 0 if the SBIN data will come from the first SCC channel during the Bd time slot.
1 if the SBIN data will come from the input register Bd.

Bit 2 IBE : 0 if the SBIN data will come from the second SCC channel during the Be time slot.
1 if the SBIN data will come from the input register Be.

Bit 1 OBD : 0 if the SBOUT data will be directed to the first SCC channel during Bd time slot.
1 if the SBOUT data will be directed to the output register Bd.

Bit 0 OBE : 0 if the SBOUT data will be directed to the second SCC channel during Be time slot.
1 if the SBOUT data will be directed to the output register Be.

### 3.7.3.3 SBP Interface Routing Commands

Data routing inside the SBP Interface subsection is determined by Bit[4-0] of the CSR. Only channels Bd and Be have selectable source/destinations while Bf will always be connected to the pipeline registers of the 8-bit bus port, IBf and OBf.

SBIN data source:

| Control Bits | | | | Input Source | | |
|---|---|---|---|---|---|---|
| IBD | IBE | OBD | OBE | Bd | Be | Bf |
| 0 | 0 | x | x | SCC A/C | SCC B/D | IBf |
| 0 | 1 | x | x | SCC A/C | IBe | IBf |
| 1 | 0 | x | x | IBd | SCC B/D | IBf |
| 1 | 1 | x | x | IBd | IBe | IBf |

Table D.5

SBOUT data destination:

| Control Bits | | | | Output Destination | | |
|---|---|---|---|---|---|---|
| IBD | IBE | OBD | OBE | Bd | Be | Bf |
| x | x | 0 | 0 | SCC A/C | SCC B/D | OBf |
| x | x | 0 | 1 | SCC A/C | OBe | OBf |
| x | x | 1 | 0 | OBd | SCC B/D | OBf |
| x | x | 1 | 1 | OBd | OBe | OBf |

Table D.6

### 3.7.4 External SCC interface

The SBP Interface provides a direct connection to two external SCC channels, C and D. The exact signal connection is depicted in figure D.1

In connecting to external SCC channels, the transmit and receive data clock inputs are supplied by the SBP. The SBP supplied clocks, RTxCC# and RTxCD# are internally synchronized and locked to SCLK from the Am79C30A.

## 3.8 IEEE P1149.1 (JTAG) PORT FUNCTIONAL DESCRIPTION

An IEEE 1149.1 compatible boundary scan Test Access Port is provided for board level continuity tests and diagnostics. All digital input, output and input/output pins are tested. Analog pins, including the AUI differential driver (DO±) and receivers (DI±, CI±), and the crystal input (XTAL1/XTAL2) pins, are not part of the scan path.

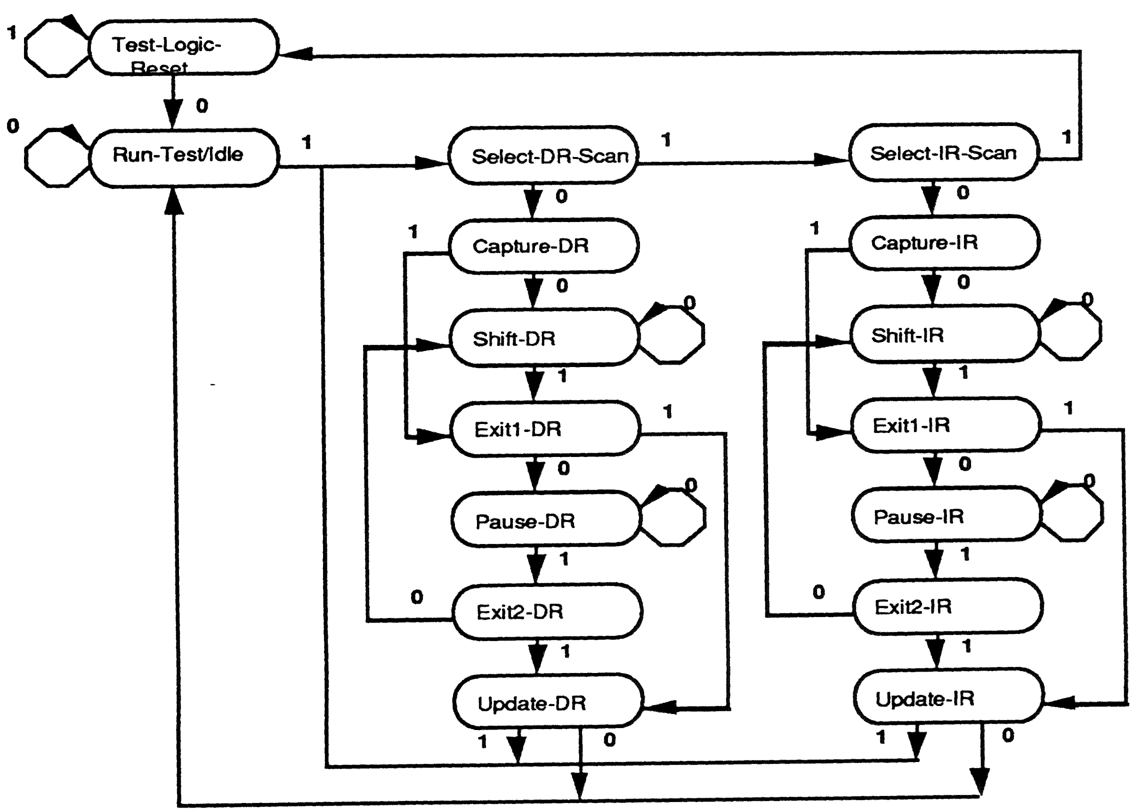
The following is a brief summary of the IEEE 1149.1 compatible test functions implemented in the CURIO.

### 3.8.1 Boundary Scan Circuit

The boundary scan test circuit requires five extra pins (TCK, TMS, TDI, TDO and TRST#), defined as the Test Access Port (TAP). It includes a finite state machine (FSM), an instruction register and a data register array. Internal pull-up resistors are provided for the TCK, TMS, TDI and TRST# pins.

### 3.8.2 TAP FSM

The TAP engine is a 16 state FSM, driven by the Test Clock (TCK) and the Test Mode Select (TMS) pins. The state diagram is shown below:



Note: The value shown adjacent to each state transition in this figure represents the signal present at TMS at the time of a rising edge at TCK.

Figure E.1 TAP Controller State Diagram

### 3.8.3 Supported Instructions

In addition to the minimum IEEE 1149.1 requirements (BYPASS, EXTEST and SAMPLE instructions), three additional instructions are provided to further ease board level testing. Unused instruction code (0101) is reserved.

| INST NAME | DESCRIPTION | SELECTED DATA REG | MODE | INST CODE |
|---|---|---|---|---|
| EXTEST | EXTERNAL TEST | BSR | TEST | 0000 |
| ID_CODE | ID CODE INSPECTION | ID REG | NORMAL | 0001 |
| SAMPLE | SAMPLE BOUNDARY | BSR | NORMAL | 0010 |
| TRI_ST | FORCE TRISTATE | BYPASS | NORMAL | 0011 |
| SET_1/0 | CONTROL BOUNDARY TO 1/0 | BYPASS | TEST | 0100 |
| reserved | reserved | n.a. | n.a. | 0101 |
| BYPASS | BYPASS SCAN | BYPASS | NORMAL | 0110 to 1111 |

Table E.1 : IEEE 1149.1 Supported Instruction Summary

### 3.8.4 Instruction Register and Decoding Logic

The instruction register gets updated only at UPDATE_IR state and TEST_LOGIC_RESET state. The TEST_LOGIC_RESET state always invokes IDCODE instruction. The instruction decoding logic gives signals to control the data flow in the DATA registers according to the current instruction.

### 3.8.5 Data Register Array

1. BSR, Boundary Scan Register.

Different types of BSR cell are provided to support input, output, I/O and tri-state control PADs.

There are four possible operational modes in the BSR cell :

(1) CAPTURE
(2) SHIFT
(3) UPDATE
(4) SYSTEM FUNCTION

2. BYPASS REG (1 bit)

3. DEV ID REG (4 + 16 +11 + 1 = 32 bits)

| | |
|---|---|
| Bits 31 - 28: | Version |
| Bits 27 - 12: | Part number |
| Bits 11 - 1: | Manufacturer ID. The 11 bit manufacturer ID code for AMD is 00000000001 according to JEDEC Publication 106-A. |
| Bit 0: | Always a logic 1 |

### 3.8.6 The TAP Reset Pin, TRST#

An independent reset pin, TRST#, is provided for TAP so that the FSM can be forced into TEST_LOGIC_RESET state after power up. It will also allow JTAG activities be independent of the CURIO operating modes (SLEEP or NORMAL MODE.) The TAP FSM is independent of other system reset pins.

## 4 ELECTRICAL SPECIFICATIONS

### ABSOLUTE MAXIMUM RATINGS

| | |
|---|---|
| Storage Temperature | -65C to +150C |
| Ambient Temperature under bias | 0 to +70C |
| Supply Voltage referenced to AVSS or DVSS ($AV_{DD}$, $DV_{DD}$) | -0.3 to +6V |

### OPERATING RANGES

| | |
|---|---|
| Temperature | 0 to +70C |
| Supply Voltage ($AV_{DD}$, $DV_{DD}$) | 5V ± 5% |

1. Stress above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

2. Operating ranges define those limits between which the functionality of the device is gauranteed.

## 4.1 DC CHARACTERISTICS

| PARAM | DESCRIPTION | TEST CONDITIONS | MIN | MAX | UNITS |
|---|---|---|---|---|---|
| $V_{IL}$ | Input LOW voltage | Vss = 0.0V | -0.5 | 0.8 | V |
| $V_{IH}$ | Input HIGH voltage | | 2.0 | $V_{DD}$+0.5 | V |
| $V_{OL}$ | Output LOW voltage | $I_{OL1}$ = 3.2mA<br>$I_{OL2}$ = 48mA<br>(note 1) | | 0.4 | V |
| $V_{OH}$ | Output HIGH voltage (note 2) | $I_{OH}$ = -0.4mA | 2.4 | | V |
| $I_{IX}$ | Input leakage current (note 3) | $V_{DD}$ = 5V, $V_{IN}$ = 0V | -10 | 10 | µA |
| $I_{IAXD}$ | Input Current at DI+ and DI- | -1V < $V_{IN}$ < $AV_{DD}$+0.5V | -500 | +500 | µA |
| $I_{IAXC}$ | Input current at CI+ and CI- | -1V < $V_{IN}$ < $AV_{DD}$+0.5V | -500 | +500 | µA |
| $I_{ILX}$ | XTAL1 Input LOW Current | $V_{IN}$ = $V_{SS}$ | | 10 | µA |
| $I_{IHX}$ | XTAL1 Input HIGH Current | $V_{IN}$ = $V_{DD}$ | | 10 | µA |
| $I_{OZ}$ | Output Leakage Current (note 4) | 0.4V < $V_{OUT}$ < $V_{DD}$ | -10 | 10 | µA |
| $V_{OD}$ | Differential Output Voltage \|(DO+)-(DO-)\| | $R_L$ = 78Ω<br>VO<br>VO* | <br>630<br>-630 | <br>870<br>-870 | <br>mV<br>mV |
| $V_{ODOFF}$ | Transmit Differential Output Idle Voltage | $R_L$ = 78Ω<br>(note 7) | -20 | 20 | mV |

| | | | | | |
|---|---|---|---|---|---|
| $I_{ODOFF}$ | Transmit Differential Output Idle Current | $R_L = 78\Omega$ (note 6) | -325 | 325 | µA |
| $V_{CMT}$ | Transmit Output Common Mode Voltage | $R_L = 78\Omega$ | 2.5 | 4.2 | V |

| PARAM | DESCRIPTION | TEST CONDITIONS | MIN | MAX | UNITS |
|---|---|---|---|---|---|
| $V_{ODI}$ | DO± Transmit Differential Output Voltage Imbalance | $R_L$ = 78Ω (note 5) | | 20 | mV |
| $V_{IRD}$ | Receive Data Differential Input Threshold | | -35 | 35 | mV |
| $V_{IDC}$ | DI± and CI± Differential Input Threshold | | -160 | -275 | mV |
| $V_{IRDVD}$ | DI± and CI± Differential Mode Input Voltage Range | | | 1.5 | V |
| $V_{ICM}$ | DI± and CI± Input Bias Voltage | $I_{IN}$ = 0 mA | | TBD | V |
| $V_{OPD}$ | DO± Undershoot Voltage at zero differential on transmit return to zero (ETD) | (note 7) | | -100 | mV |
| $I_{DD}$ | Power Supply Current | ESCLK = 25 MHz XTAL1 = 20 MHz | | 300 | mA |
| $I_{DDSLEEP}$ | Power Supply Current | ESLEEP# active (note 8) | | 500 | µA |

NOTES:

1. $I_{OL1}$ = 3.2mA:
   - MACE: TXEN#, TXDAT+/-, RXDAT, RXCRS, DATAW<15-0>, ADD<4-0>, RDTREQ#, TDTREQ#, EOF#, PWRDN#, EDTV#, EINTR#, STDCLK, SRDCLK, CLSN.
   - SCSI: SDREQ, SINT#, DATAW<15-0>.
   - ESCC: TRxCA#, TRxCB#, DTR/REQA#, DTR/REQB#, SYNCA#, SYNCB#, TxDA, TxDB, RTSA#, RTSB#, INT1#, ~W/REQA#, ~W/REQB#, REQA#, REQB#, DATAB<7-0>.
   - SBP: SBIN, RxDC, RxDD, RTxCC, RTxCD, DATAB<7-0>.
   - IEEE: TDO.
   - General: XCLK.

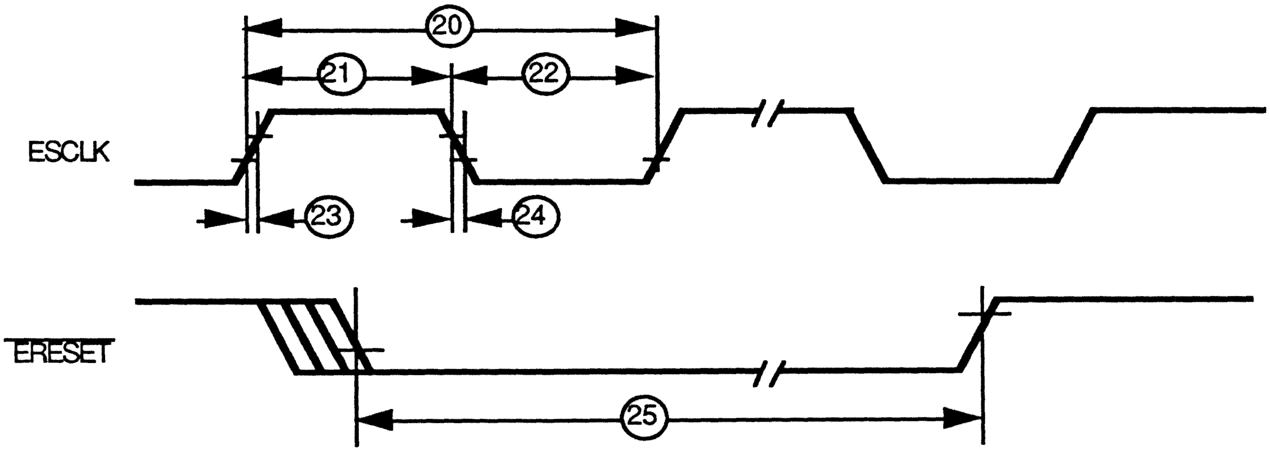
   $I_{OL2}$= 48mA:
   - SCSI: SDIO<7-0>#, SDP#, BSY#, SEL#, RST#, REQ#, ACK#, ATN#, MSG#, C/D#, I/O#.

2. $V_{OH}$ does not apply to open-drain output pins.
3. $I_{IX}$ applies to all input only pins except DI+/-, CI+/-, and XTAL1.
4. $I_{OZ}$ applies to all three-state output pins and bi-directional pins.
5. Tested, but to values in excess of limits. Test accuracy not sufficient to allow screening guard bands.
6. Correlated to other tested parameters - not tested directly.
7. Test not implemented to data sheet specification.

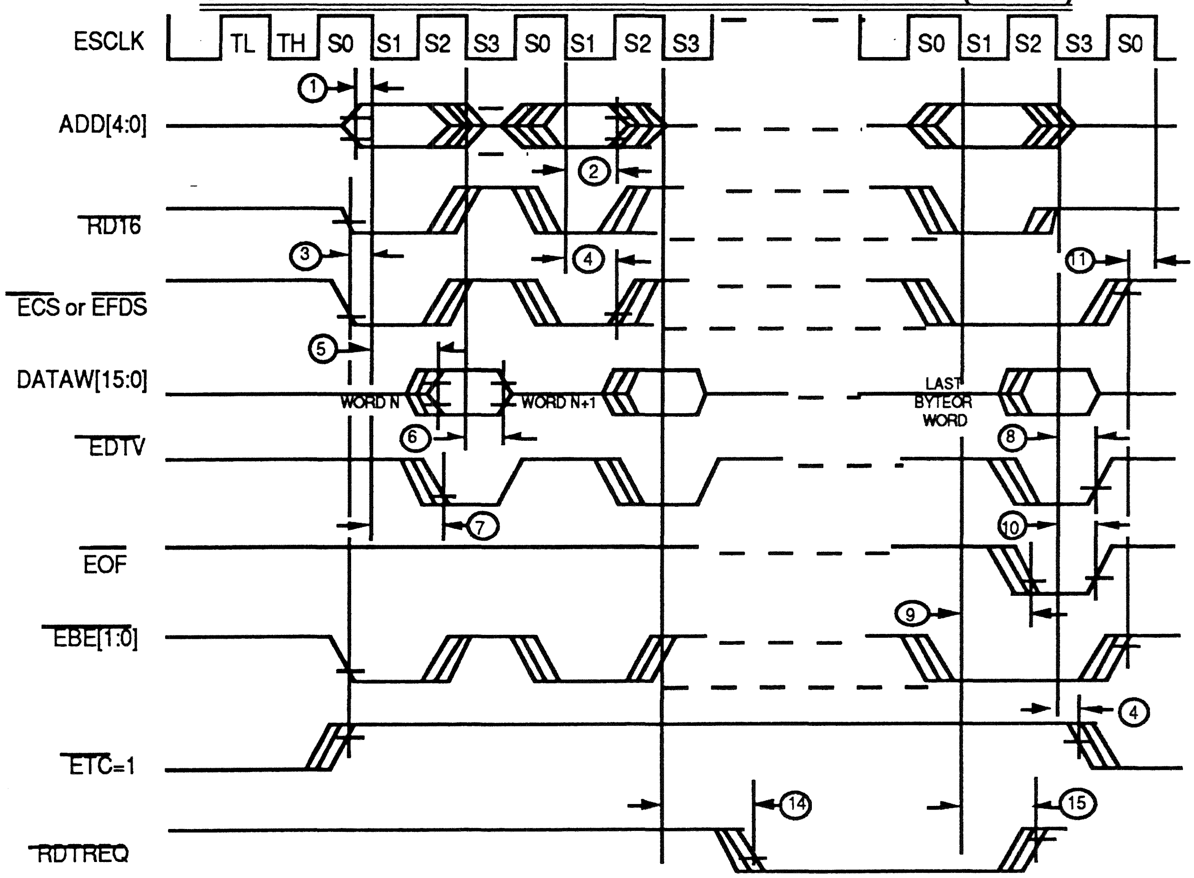
8. During the activation of ESLEEP#:
   (i) The following pins are placed in a high impedance state :
   TXDAT-, XCLK, PWRDN#, EDTV#, TDTREQ#, RDTREQ#, EINTR#.
   (ii) The following I/O pins are placed in a high impedance mode and have their internal TTL level translators disabled :
   EOF#, SRDCLK, RXCRS, RXDAT, CLSN, TXEN#, STDCLK and TXDAT+.
   (iii) The following pins are pulled low :
   XTAL1 (XTAL2 feedback is cut off from XTAL1), DO+ and DO-.
   (iv) The ESLEEP# attributes of the following pins are to be determined:
   DI+, DI-, CI+ and CI-.
9. Input hysteresis applies to all SCSI bus inputs.

## 4.2 AC CHARACTERISTICS

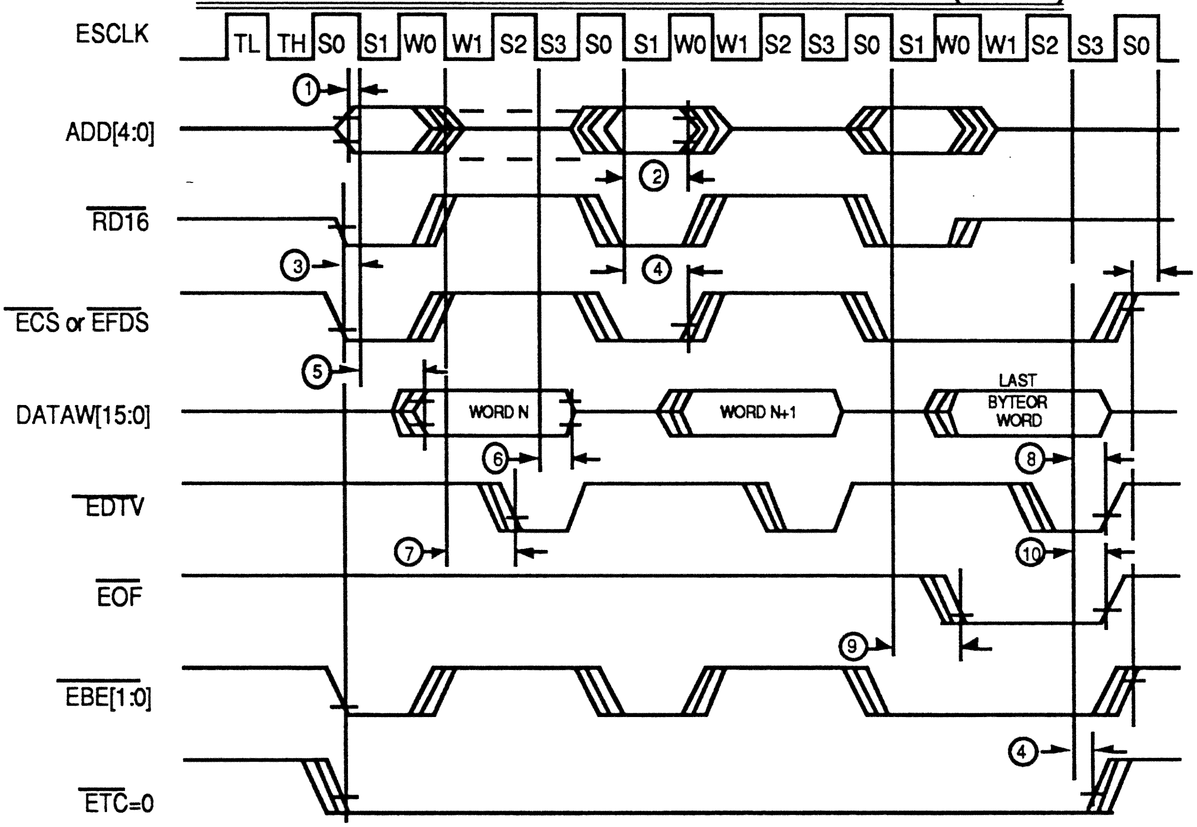
### 4.2.1 MACE AC Timing



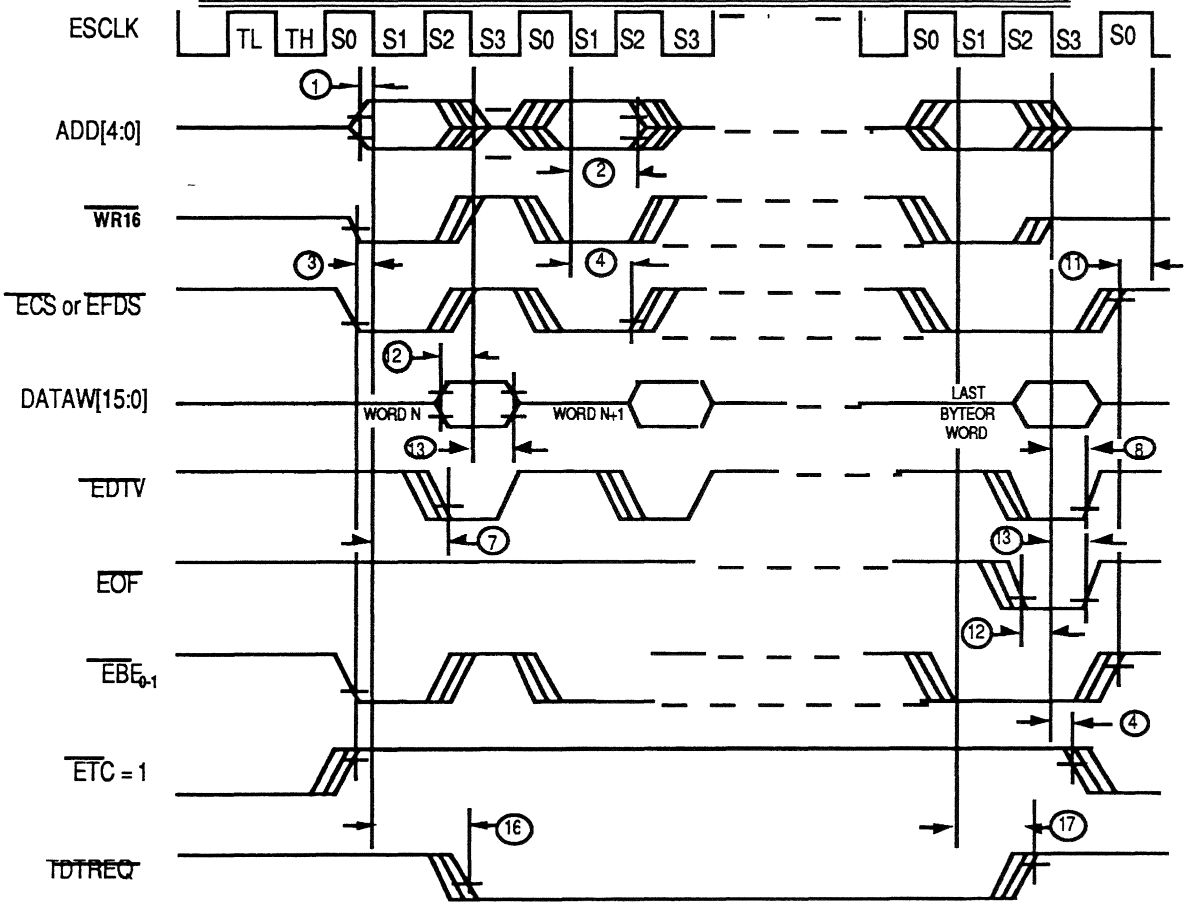

Clock and Reset Timing



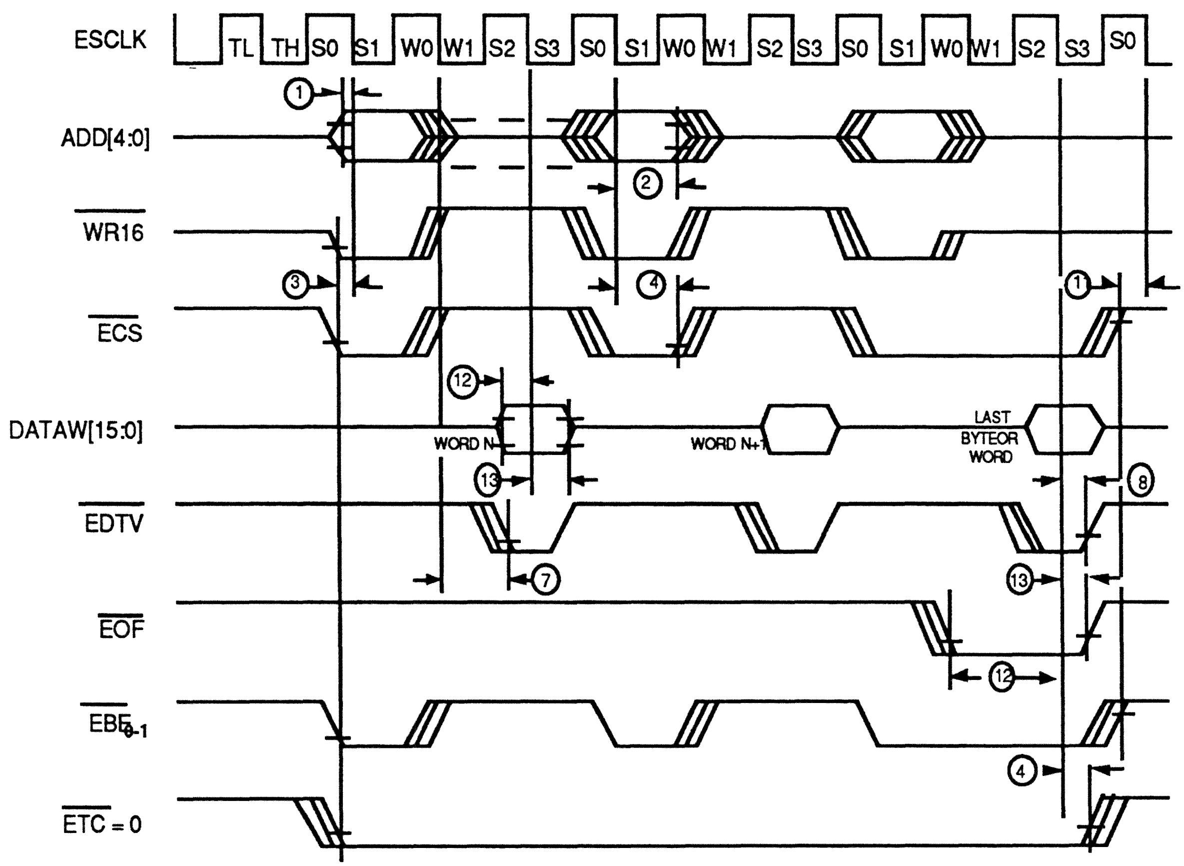

Host System Interface - 2 Cycle Receive FIFO/Register Read Timing




Host System Interface - 3 Cycle Receive FIFO/Register Read Timing




Host System Interface - 2 Cycle Transmit FIFO/Register Write Timing



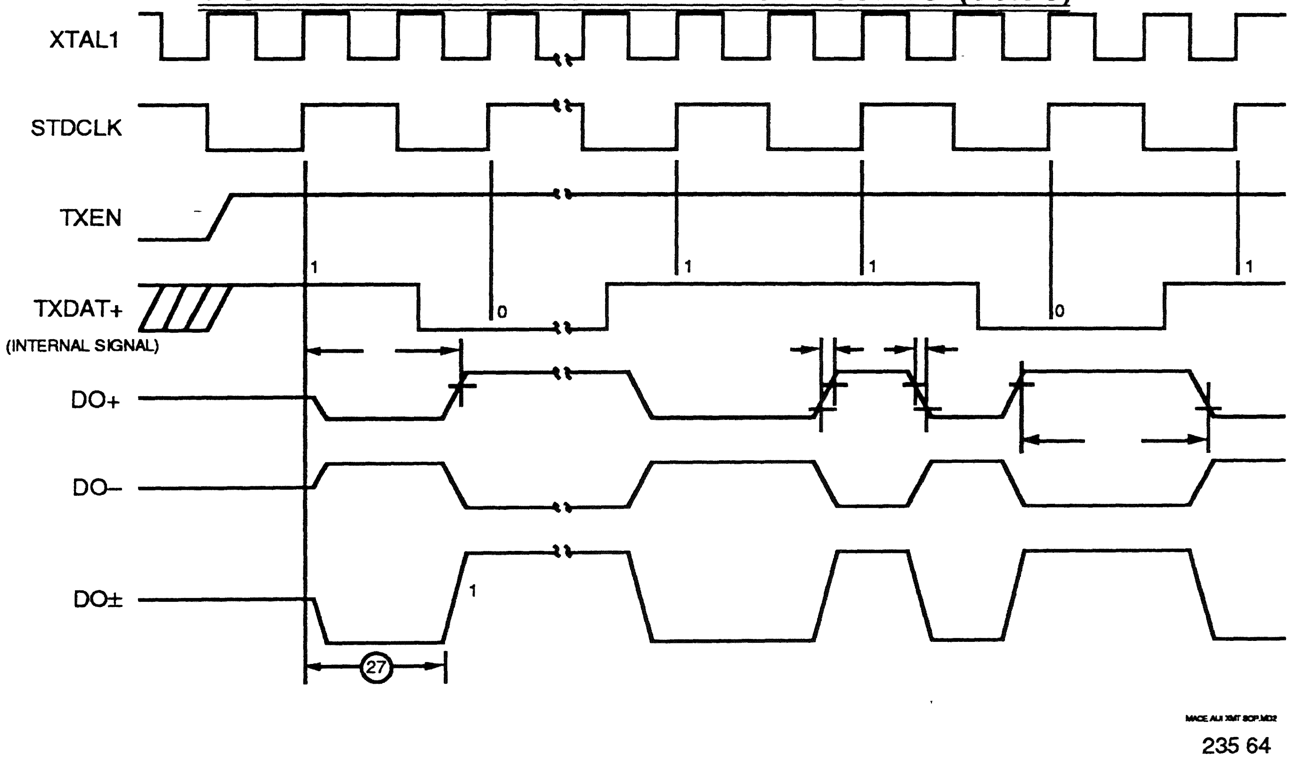

Host System Interface - 3 Cycle Transmit FIFO/Register Write Timing

# THIS PAGE INTENTIONALLY LEFT BLANK

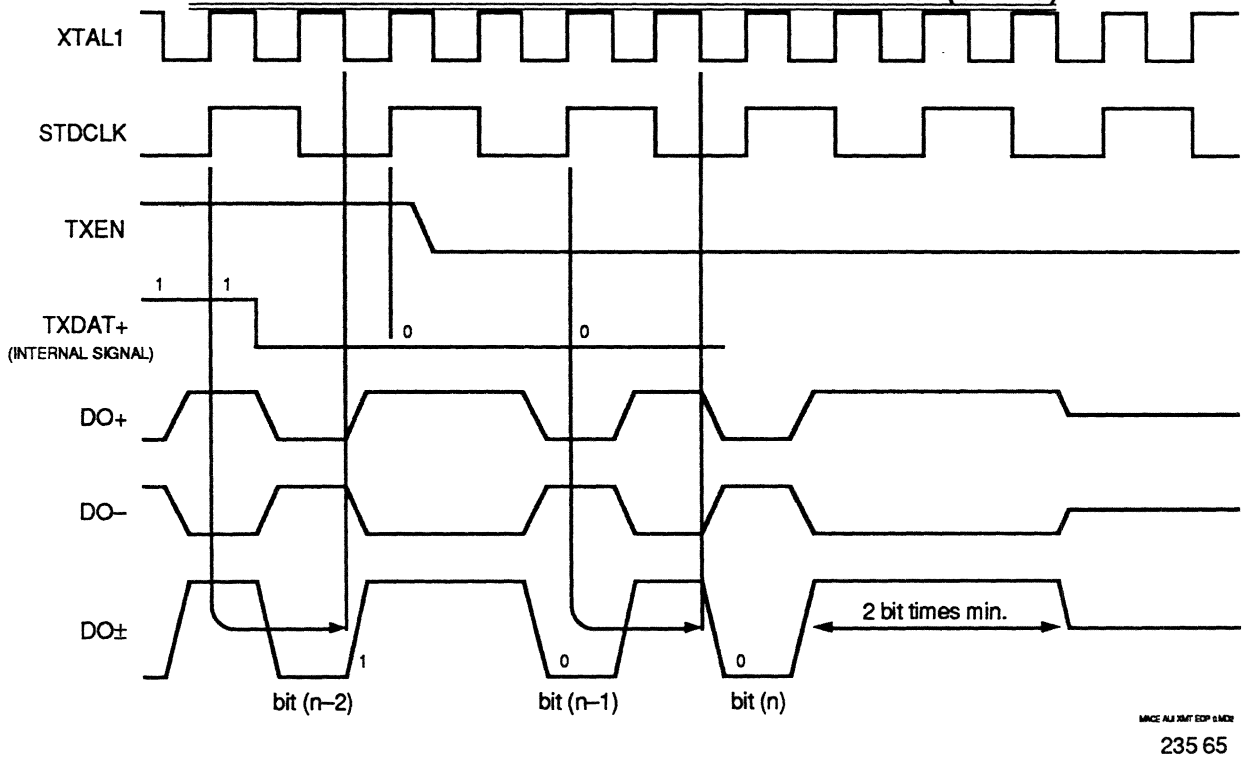
Host System Interface - Detailed RDTREQ# Timing

# THIS PAGE INTENTIONALLY LEFT BLANK

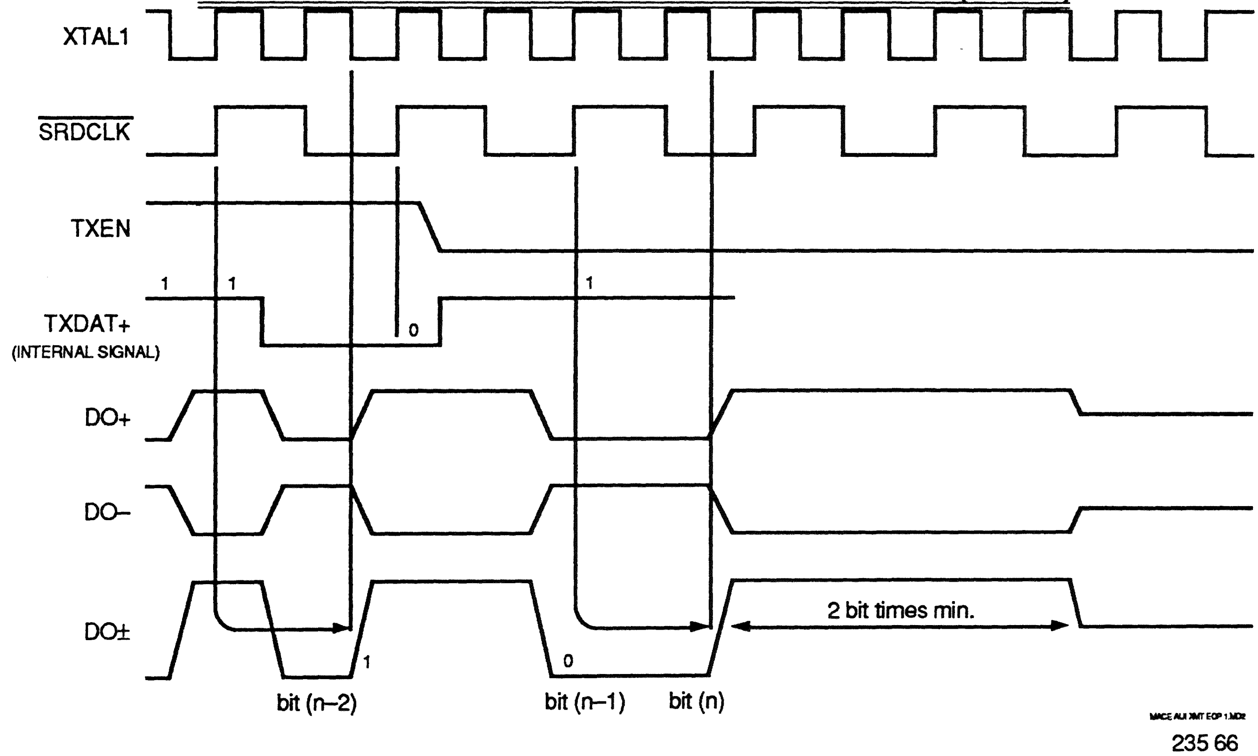
Host System Interface - Detailed TDTREQ# Timing




AUI Transmit Timing - Start of Packet



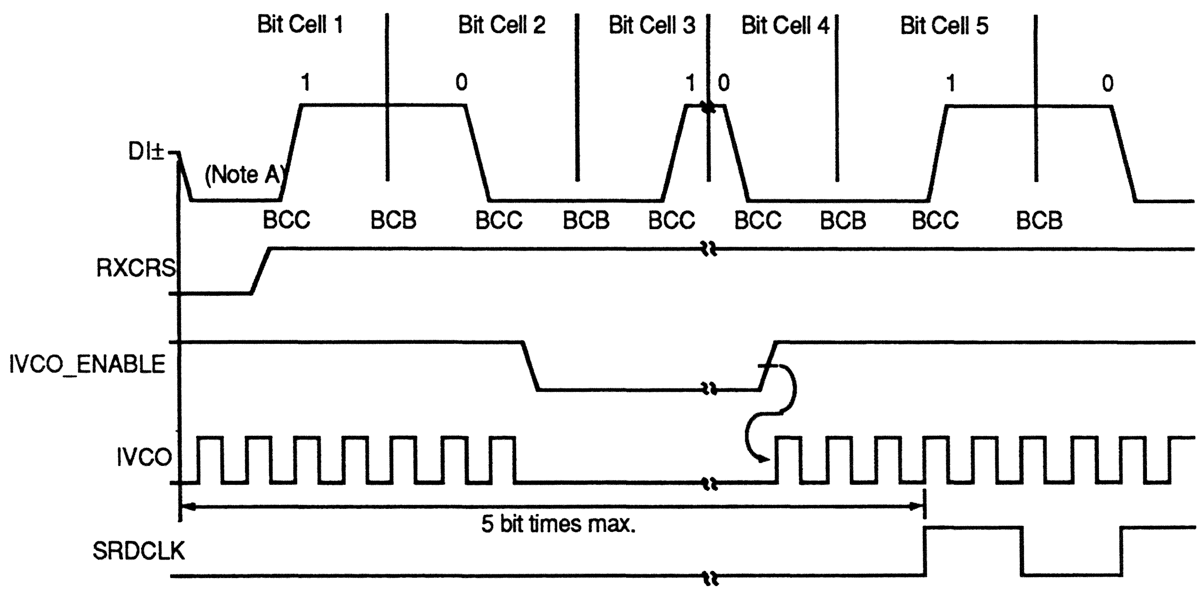

AUI Transmit Timing - End of Packet (Last Bit = 0)




AUI Transmit Timing - End of Packet (Last Bit = 1)

# THIS PAGE INTENTIONALLY LEFT BLANK

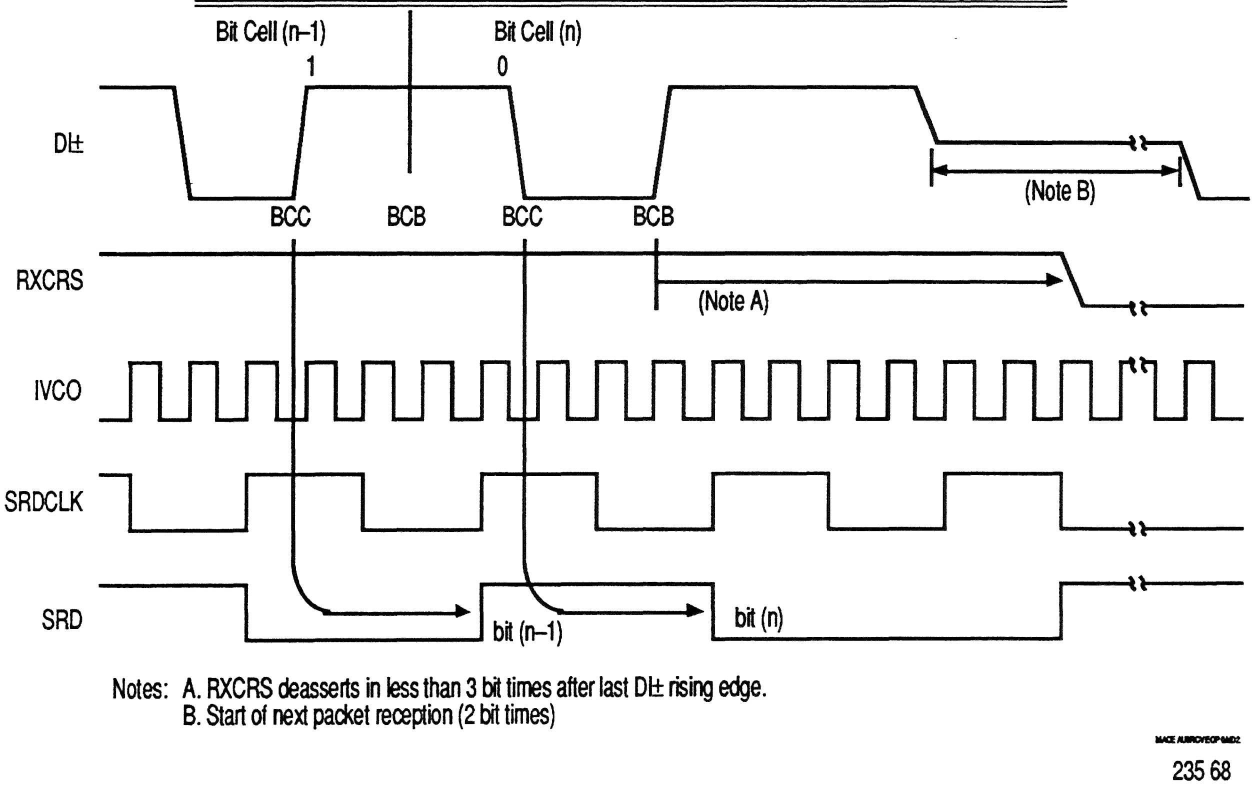
AUI Transmit Timing - End Transmit Delimiter (ETD)



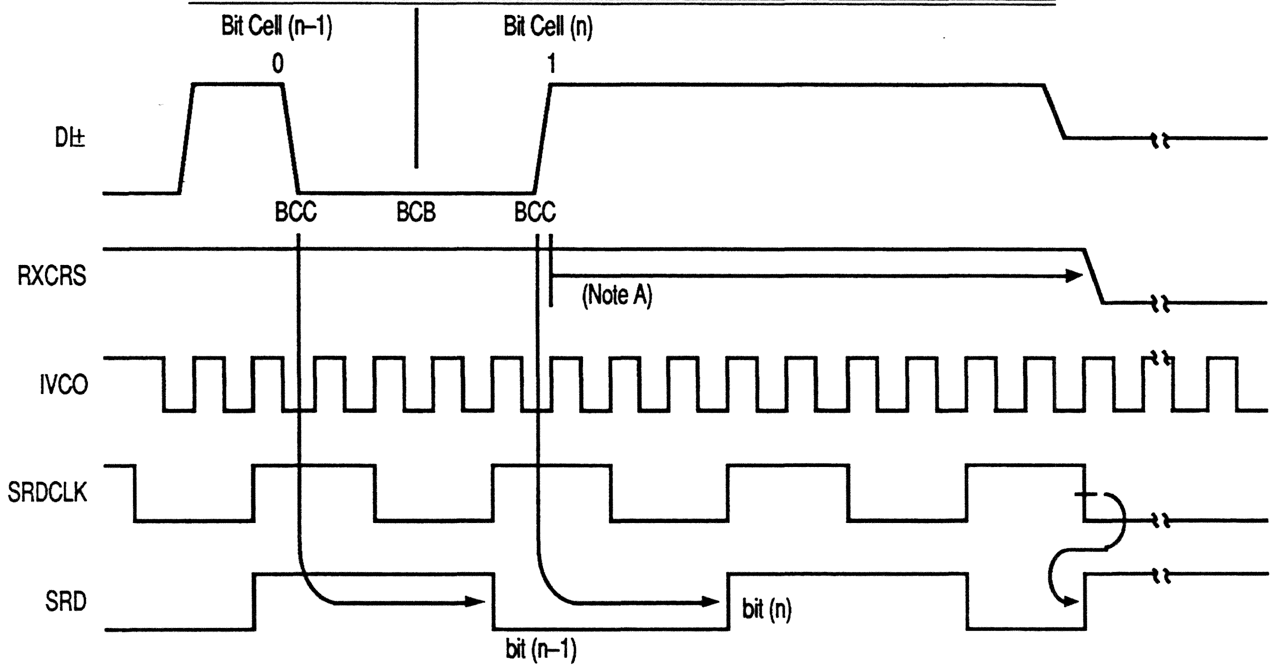

Note A: Min. pulse width > 45 ns with amplitude > -160mV.

235 67

AUI Receive Timing - Start of Packet




AUI Receive Timing - End of Packet (Last Bit = 0)




Notes: A. RXCRS deasserts in less than 3 bit times after last DI± rising edge.

235 69

AUI Receive Timing - End of Packet (Last Bit = 1)

# THIS PAGE INTENTIONALLY LEFT BLANK

AUI Collision Timing

# THIS PAGE INTENTIONALLY LEFT BLANK

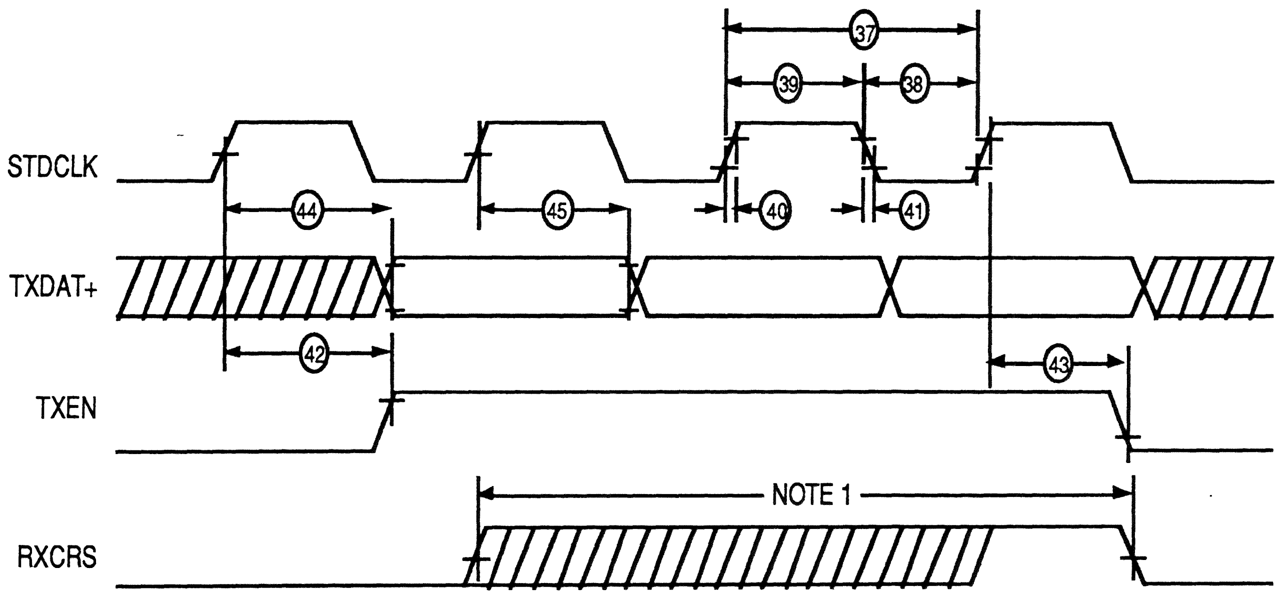
DAI™ Transmit Timing

# THIS PAGE INTENTIONALLY LEFT BLANK

DAI™ Receive Timing

# THIS PAGE INTENTIONALLY LEFT BLANK

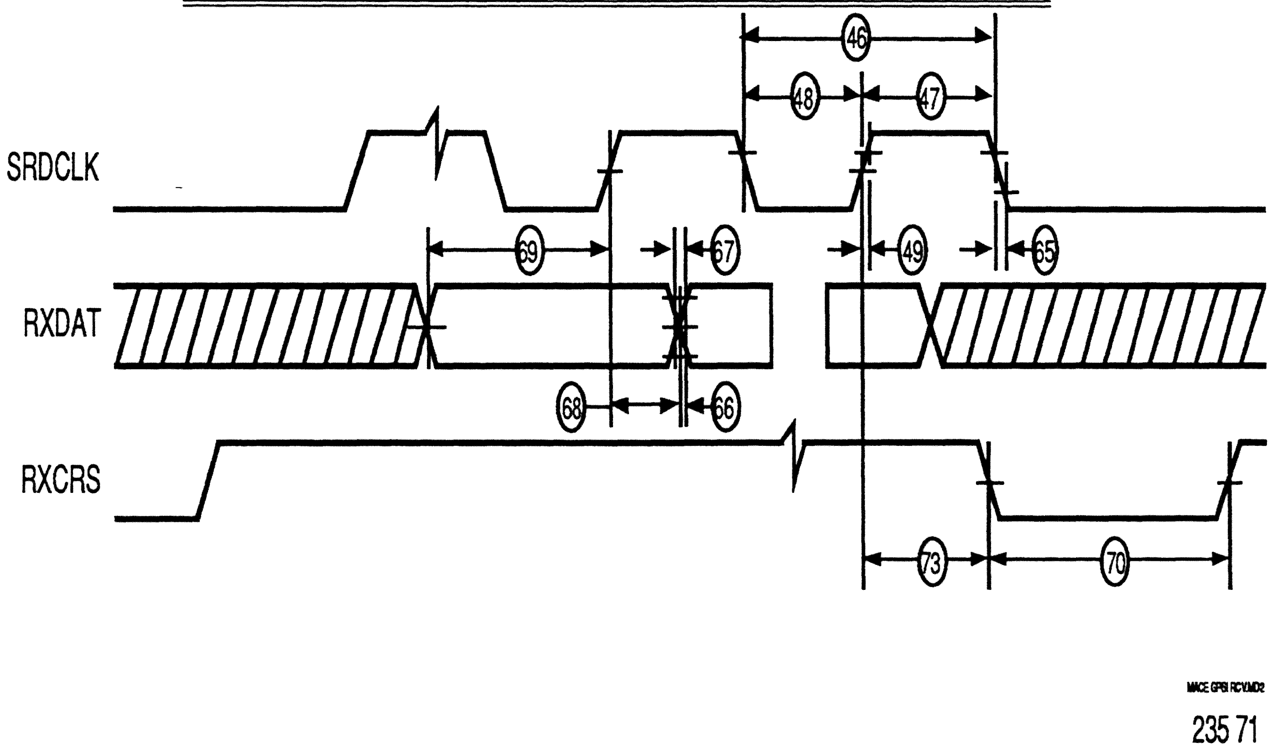
DAI™ Collision Timing



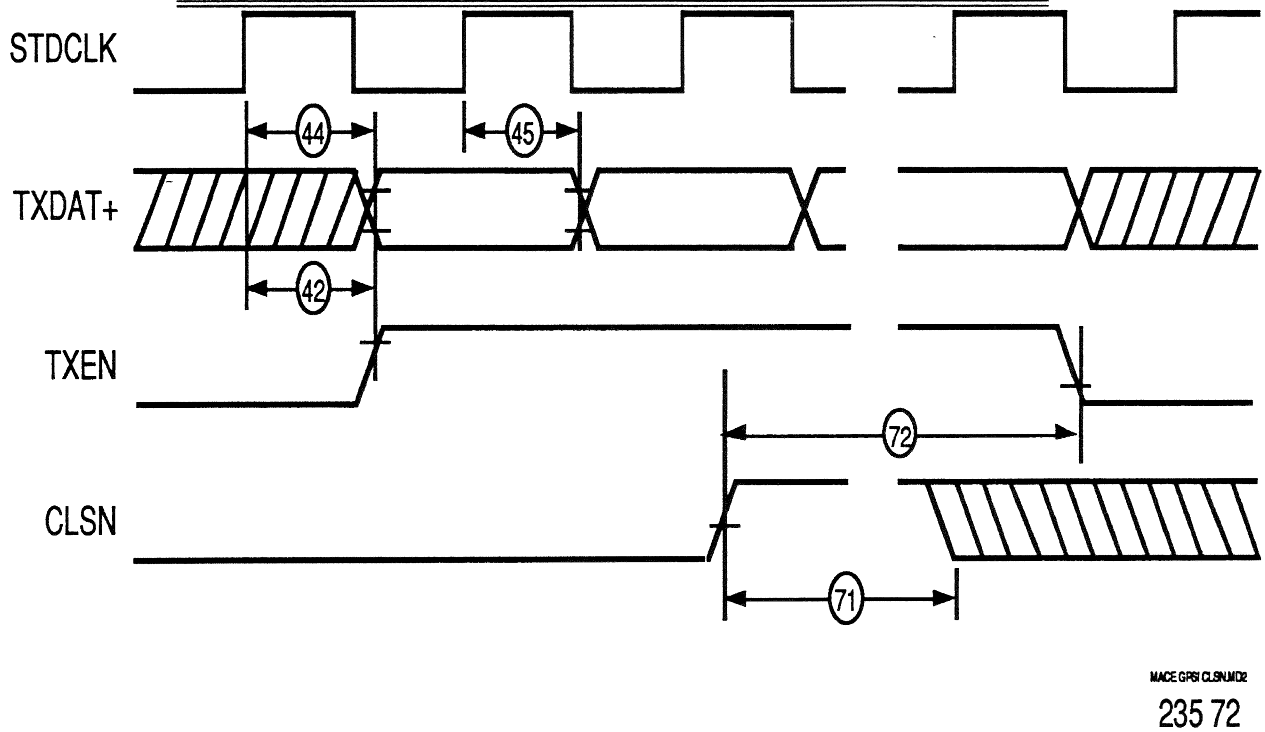

*During transmit, the RXCRS input must be asserted (high) and remain active-high after TXEN goes active (high). If RXCRS is deasserted before TXEN is deasserted, LCAR will be reported (Transmit Frame Status) after the transmission is completed by the MACE.

235 70

GPSI Transmit Timing




GPSI Receive Timing




GPSI Collision Timing

4.2.1.1 Clock and Reset Timing

| # | PARAM | DESCRIPTION | TEST CONDITIONS | MIN (ns) | MAX (ns) |
|---|---|---|---|---|---|
| 20 | $t_{ESCLK}$ | ESCLK period : | | 40 | 200 |
| 21 | $t_{ESCLKH}$ | ESCLK HIGH pulse width | | 0.4*$t_{ESCLK}$ | 0.6*$t_{ESCLK}$ |
| 22 | $t_{ESCLKL}$ | ESCLK LOW pulse width | | 0.4*$t_{ESCLK}$ | 0.6*$t_{ESCLK}$ |
| 23 | $t_{ESCLKR}$ | ESCLK Rise Time | | | 5 |
| 24 | $t_{ESCLKF}$ | ESCLK Fall Time | | | 5 |
| 25 | $t_{RST}$ | ERESET# pulse width | | 15*$t_{ESCLK}$ | |
| | $t_{BT}$ | Network Bit Time (=2*$t_{XCLK}$ or $t_{ESTDCLK}$) | | 99 | 101 |

4.2.1.2 BIU Timing

| # | PARAM | DESCRIPTION | TEST CONDITIONS | MIN (ns) | MAX (ns) |
|---|---|---|---|---|---|
| 1 | | Address valid setup to ESCLK↓ | | 7 | |
| 2 | | Address valid hold after ESCLK↓ | | 7 | |
| 3 | | ECS# or EFDS# and ETC#, $EBE_{1-0}$#, RD16#, WR16# setup to ESCLK↓ | | 7 | |
| 4 | | ECS# or EFDS# and ETC#, $EBE_{1-0}$#, RD16#, WR16# hold after ESCLK↓ | | 7 | |
| 5 | | Data out valid delay from ESCLK↓ | | | 33 |
| 6 | | Data out valid hold after ESCLK↓ | | 4 | TBD |
| 7 | | EDTV# valid delay from ESCLK↓ | | | 33 |
| 8 | | EDTV# valid hold after ESCLK↓ | | 4 | |
| 9 | | EOF# output valid delay from ESCLK↓ | | | 33 |
| 10 | | EOF# output valid hold after ESCLK↓ | | 4 | |
| 11 | | ECS# inactive prior to ESCLK↓ | | 7 | |
| 12 | | Data in, EOF# input valid setup to ESCLK↓ | | 7 | |
| 13 | | Data in, EOF# input valid hold after ESCLK↓ | | 7 | |
| 14 | | RDTREQ# valid delay from ESCLK↓ | | | 33 |
| 15 | | RDTREQ# valid hold from ESCLK↓ | | | 33 |
| 16 | | TDTREQ# valid delay from ESCLK↓ | | | 33 |
| 17 | | TDTREQ# valid hold from ESCLK↓ | | | 33 |

4.2.1.3 AUI Timing

THIS PAGE INTENTIONALLY
LEFT BLANK

4.2.1.4 DAI™ Timing

THIS PAGE INTENTIONALLY
LEFT BLANK

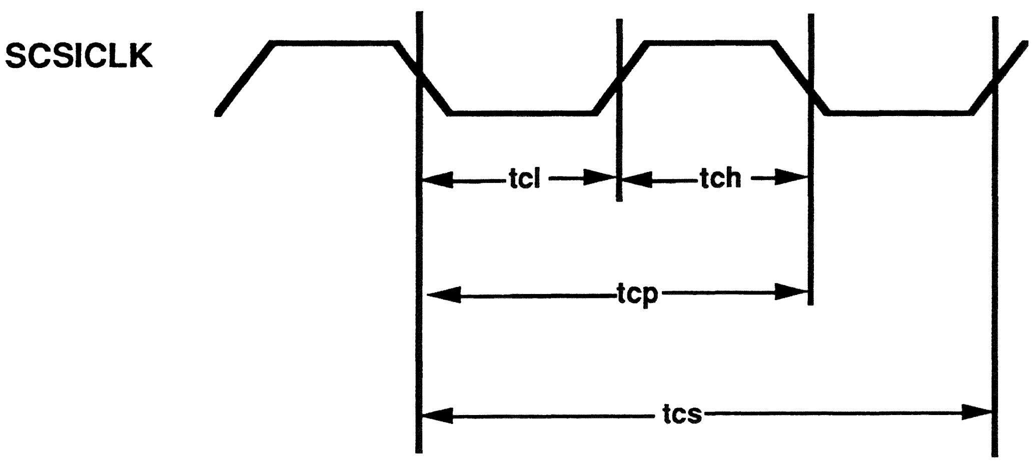
4.2.1.5 GPSI Timing

| # | PARAM | DESCRIPTION | TEST CONDITIONS | MIN (ns) | MAX (ns) |
|---|---|---|---|---|---|
| 37 | $t_{STDC}$ | STDCLK period | | 99 | 101 |
| 38 | $t_{STDCL}$ | STDCLK low pulse width | | 45 | 55 |
| 39 | $t_{STDCH}$ | STDCLK high pulse width | | 45 | 55 |
| 40 | $t_{STDCR}$ | STDCLK rise time | (Note 2) | | 8 |
| 41 | $t_{STDCF}$ | STDCLK fall time | (Note 2) | | 8 |
| 42 | | STDCLK≠ delay to TXEN↑ | ($C_L$=50pF) | | 70 |
| 43 | | TXEN hold time from STDCLK↑ | ($C_L$=50pF) | 5 | |
| 44 | | STDCLK↑ delay to TXDAT+ | ($C_L$=50pF) | | 70 |
| 45 | | TXDAT+ hold time from STDCLK↑ | ($C_L$=50pF) | 5 | |
| 46 | $t_{SRDC}$ | SRDCLK period | | 85 | 115 |
| 47 | $t_{SRDCH}$ | SRDCLK HIGH pulse width | | 45 | 55 |
| 48 | $t_{SRDCL}$ | SRDCLK LOW pulse width | | 45 | 55 |
| 49 | $t_{SRDCR}$ | SRDCLK rise time | (Note 2) | | 8 |
| 50 | $t_{SRDCF}$ | SRDCLK fall time | (Note 2) | | 8 |
| 66 | | RXDAT rise time | (Note 2) | | 8 |
| 67 | | RXDAT fall time | (Note 2) | | 8 |
| 68 | | RXDAT hold time (SRDCLK↑ to RXDAT change) | | 25 | |
| 69 | | RXDAT setup time (RXDAT stable to SRDCLK↑) | | 0 | |
| 70 | | RXCRS low time | | $t_{BT}$+20 | |
| 71 | | CLSN high time | | $t_{TxC}$+30 | |
| 72 | | TXEN hold time from CLSN↑ | | 32*$t_{BT}$ | 40*$t_{BT}$ |
| 73 | | RXCRS hold time from SRDCLK↑ | | 0 | |

### 4.2.2 SCSI AC Timing

The A.C. characteristics described herein apply over the operating voltage and temperature range of 4.75V ≤ Vdd ≤ 5.25V and 0° C ≤ Ta ≤ 70° C. Output timing is based on simulation under worst case conditions (4.75V, 70° C) and worst case processing using the following termination:

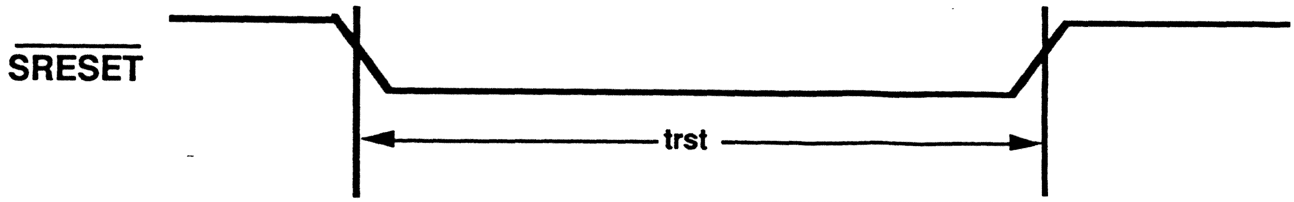
| Pins | Termination |
|---|---|
| REQ#, SDP#,SDIO#<7:0> | 50 pF |
| SINT# | 50 pF, 1KΩ pull-up |
| DATAW<15:0> | 80 pF |
| SDP#, SDIO#<7:0>, RST#, SEL#, BSY#, ATN#, MSG#, C/D#, I/O#, REQ#, ACK# | 200 pF, 110Ω pullup, 165Ω pulldown |



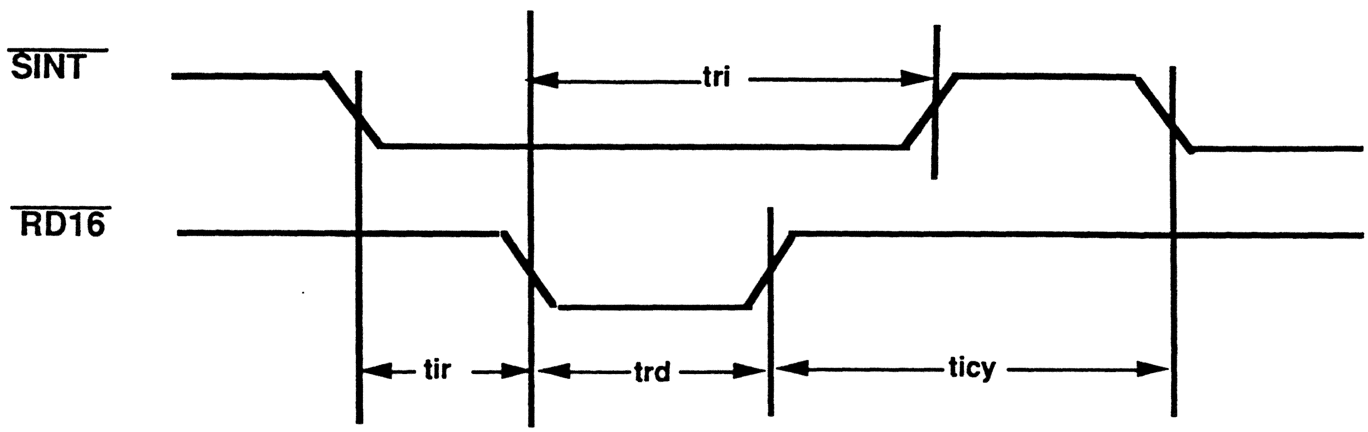

| Clock Input | | | | | |
|---|---|---|---|---|---|
| Symbol | Description | Min | Max | Units | notes |
| fcpa | Clock frequency, asynch SCSI | 10 | 25 | Mhz | |
| fcps | Clock frequency, synch SCSI | 12 | 25 | Mhz | |
| tch | Clock high time | 16.0 | | ns | 1 |
| tcl | Clock low time | 16.0 | | ns | 1 |
| tcp | Clock period | 40 | 100 | ns | |
| tcs | Synchronization latency = tcp + tcl | | | | |

Notes 1. SCSICLK must also meet the following for synchronous SCSI data transmission:

$$2tcp + tcl \geq 97.92 \text{ ns}$$
$$2tcp + tch > 97.92 \text{ ns}$$



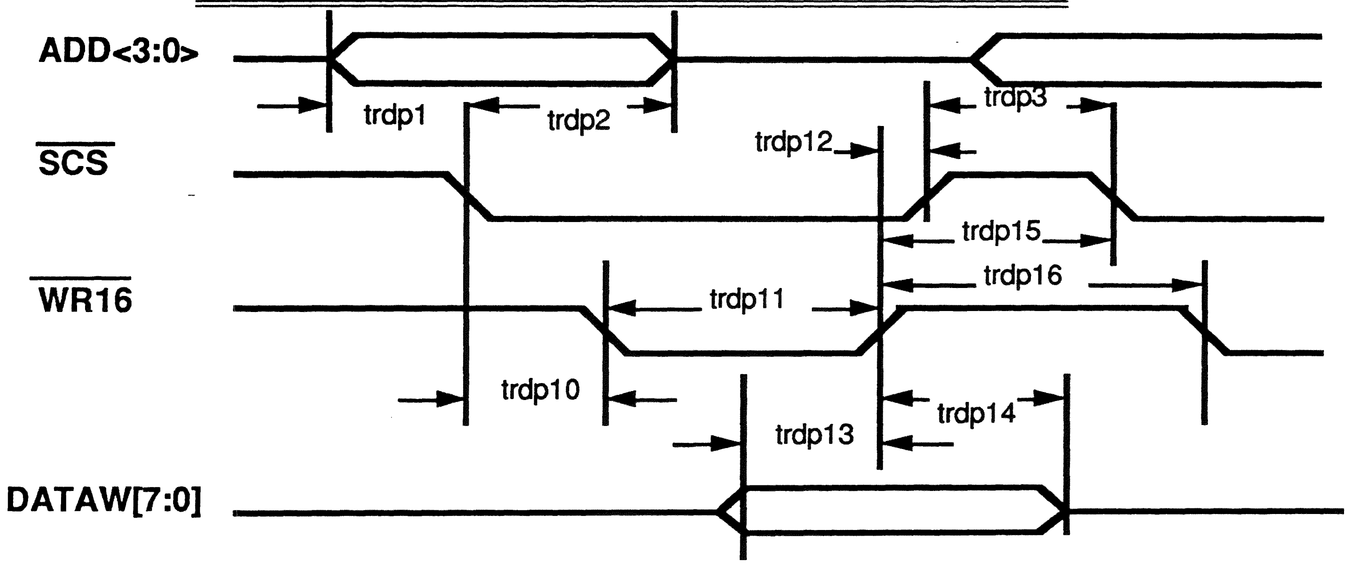

| Reset Input | | | | |
|---|---|---|---|---|
| Symbol | Description | Min | Max | Units |
| trst | SRESET# pulse width | 500 | | ns |



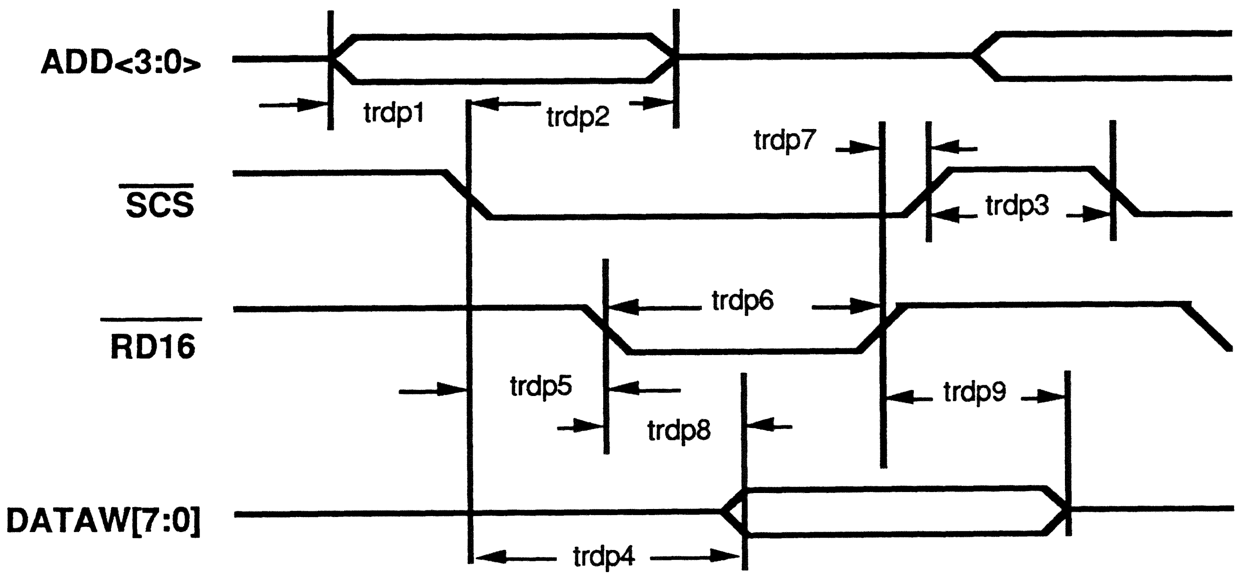

**Interrupt Output** (note 1.)

| Symbol | Description | Min | Max | Units | Notes |
|---|---|---|---|---|---|
| tir | SINT# low to Interrupt Register Read | 0 | | ns | 2 |
| trd | RD16# pulse width | 50 | | ns | |
| tri | RD16# low to SINT# high | | 100 | ns | |
| ticy | RD16# high to SINT# low | tcs-tri | | ns | |

Notes:
1. For the timing requirements of SCS#, RD16# and Address signals please refer to the register read specification.
2. When SINT# is not asserted, the interrupt register should not be read.




**Register Write**

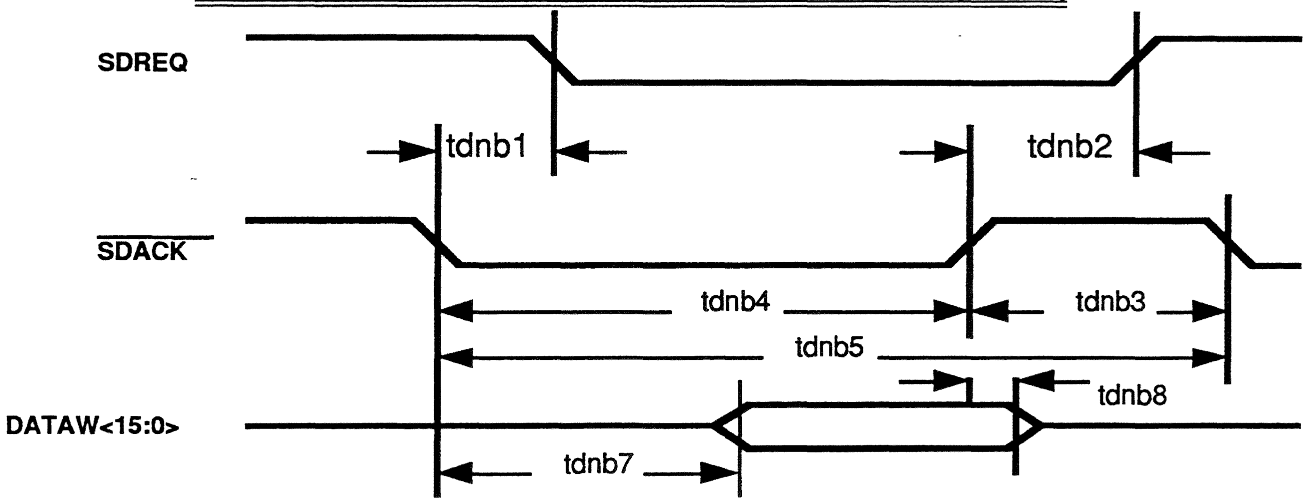



**Register Read**

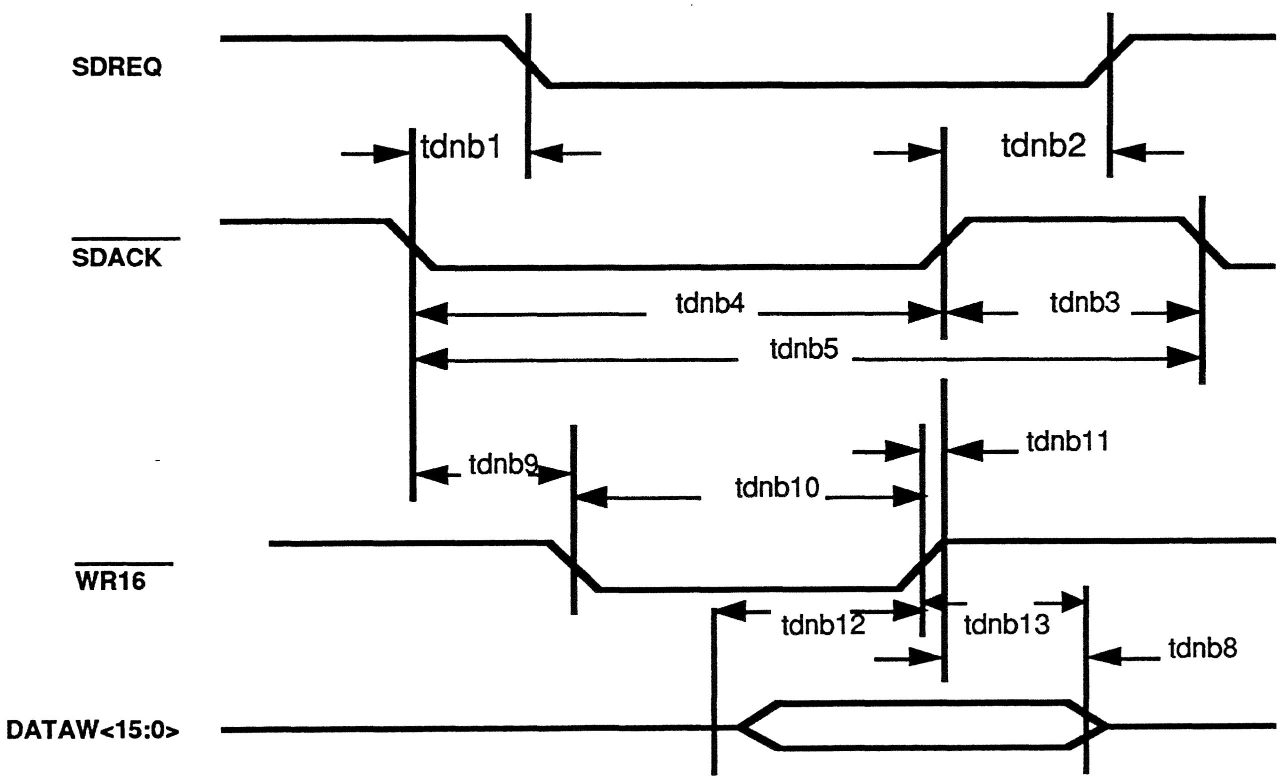
| Register Interface | | | | | |
|---|---|---|---|---|---|
| Symbol | Description | Min | Max | Units | notes |
| trdp1 | Address Setup to SCS# low | 0 | | ns | 1,2 |
| trdp2 | Address hold from SCS# low | 50 | | ns | 1,2 |
| trdp3 | SCS# high to SCS# low | 40 | | ns | |
| trdp4 | SCS# low to read data valid | | 90 | ns | 4 |
| trdp5 | SCS# setup to RD16# low | 0 | | ns | 5 |
| trdp6 | RD16# pulse width | 50 | | ns | |
| trdp7 | RD16# high to SCS# high | 0 | | ns | 5 |
| trdp8 | RD16# low to data valid | | 50 | ns | 6 |
| trdp9 | Read data output disable | 2 | 40 | ns | |
| trdp10 | SCS# setup to WR16# low | 0 | | ns | 7 |
| trdp11 | WR16# pulse width | 40 | | ns | |
| trdp12 | WR16# high to SCS# high | 0 | | ns | 7 |
| trdp13 | Data setup to WR16# high | 15 | | ns | |
| trdp14 | Data hold after WR16# high | 0 | | ns | |
| trdp15 | WR16# high to SCS# low | 60 | | ns | |
| trdp16 | WR16# high to WR16# low | 60 | | ns | |

Notes:

1. The FIFO must not be accessed when DMA or SCSI is active.
2. A new register address is latched at the following edge of the SCS# signal.
3. SDACK# must remain high for all register accesses.
4. trdp8 must be satisfied simultaneously.
5. If RD16# is held low, the time from SCS# low to stable data is trdp4; and the output disable time from SCS# high is trdp9. RD16# edges may come before or after SCS# edges. Recommended values are trdp5>trdp4-trdp8 and trdp7 > trdp9.
6. trdp4 must be satisfied simultaneously.
7. WR16# edges may come before or after SCS# edges. Recommended values are trdp10 ≥0 and trdp 12 ≥ 0. If WR16# is held low, the data setup to SCS# high is 25ns minimum; data hold from SCS# high is 60 ns minimum; trdp3 is 60 ns minimum.




DMA Read




DMA Write

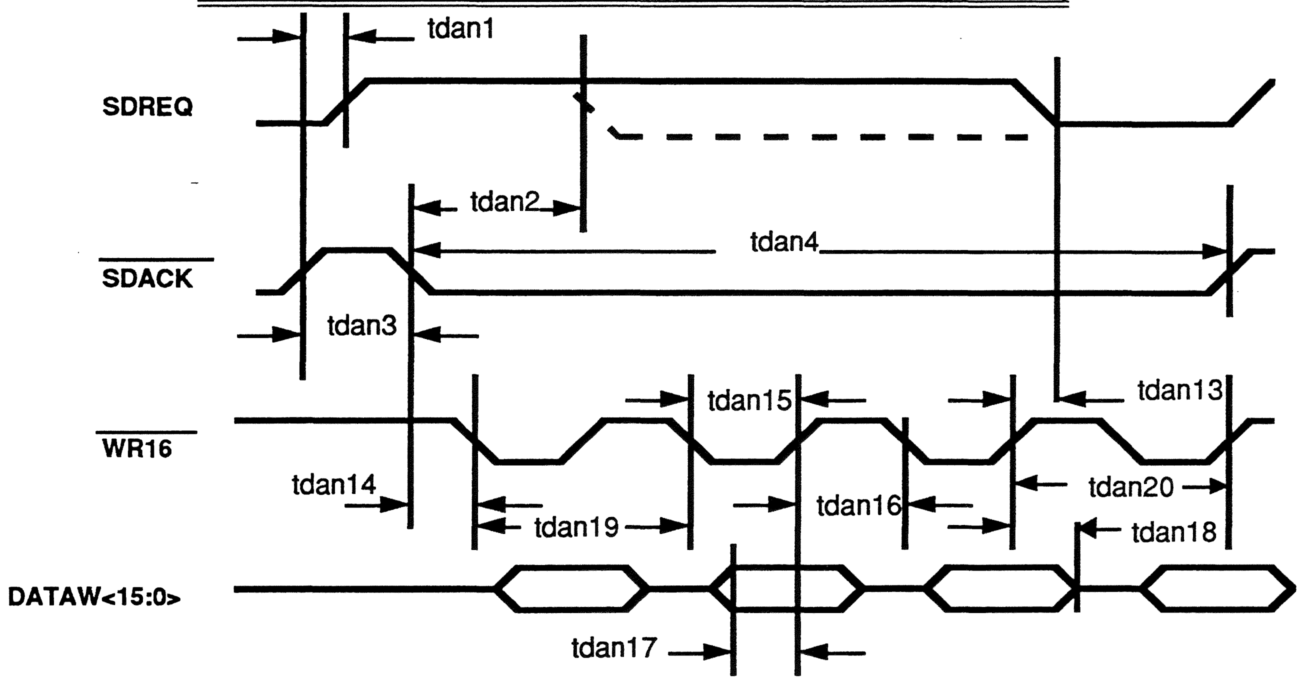
## DMA Interface

(notes 1,2,3)

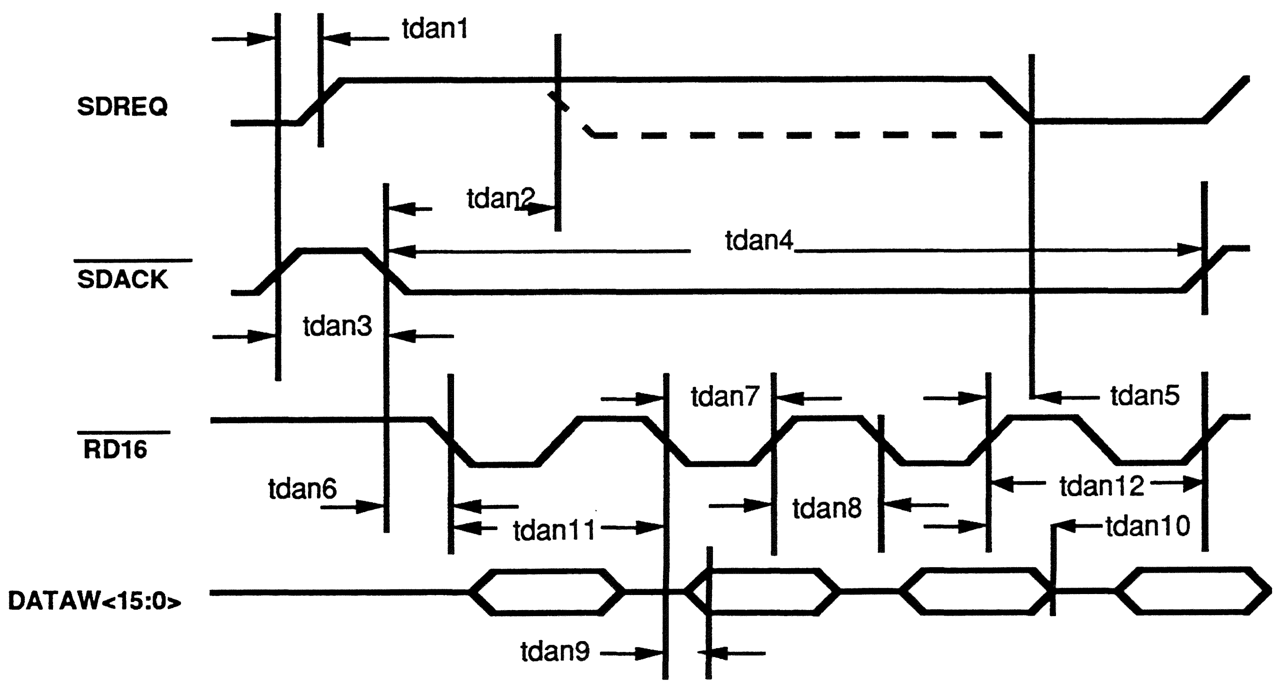
| Symbol | Description | Min | Max | Units | notes |
|---|---|---|---|---|---|
| tdnb1 | SDACK# low to SDREQ low | | 38 | ns | 5 |
| tdnb2 | SDACK# high to SDREQ high | | 40 | ns | |
| tdnb3 | SDACK# high to SDACK# low | 12 | | ns | 4 |
| tdnb4 | SDACK# pulse width | 60 | | ns | |
| tdnb5 | SDACK# period (low to low) | 100 | | ns | |
| tdnb6 | SDACK# period (high to high) | | | ns | 6 |
| tdnb7 | SDACK# low to data valid | | 41 | ns | |
| tdnb8 | data release time | 2 | 40 | ns | |
| tdnb9 | SDACK# low to WR16# low | 0 | | ns | 4 |
| tdnb10 | WR16# pulse width | 50 | | ns | |
| tdnb11 | WR16# high to SDACK# high | 0 | | ns | 4 |
| tdnb12 | Data setup to WR16# | 15 | | ns | |
| tdnb13 | Data hold from WR16# | 0 | | ns | |
| tdnb14 | WR16# high to WR16# low | 40 | | ns | |

Notes:

1. Alternate DMA mode is disabled.
2. SCS# and SDACK# shall not be active at the same time.
3. Toggle SDACK# once for every access.
4. WR16# edges may come before or after SDACK# edges. Recommended values are: tdnb9≥0 and tdnb11≥0. If WR16# is held low, the data setup to SDACK# high 15 ns minimum; data hold from SDACK# high is 15 ns minimum; and tdnb3 is 40ns minimum.
5. SDREQ may stay high if there is room to accept another word into the FIFO during DMA write, or send another word during DMA read. If the current DMA acknowledge cycle fills the FIFO during a write operation, or empties the FIFO during a read operation, then SDREQ will be de-asserted.
6. Minimum high to high SDACK# period for synchronous SCSI transfer is: tcs+50-tdnb3 for asynchronous SCSI and 2tcp for synchronous SCSI.




**Burst Mode DMA Write**



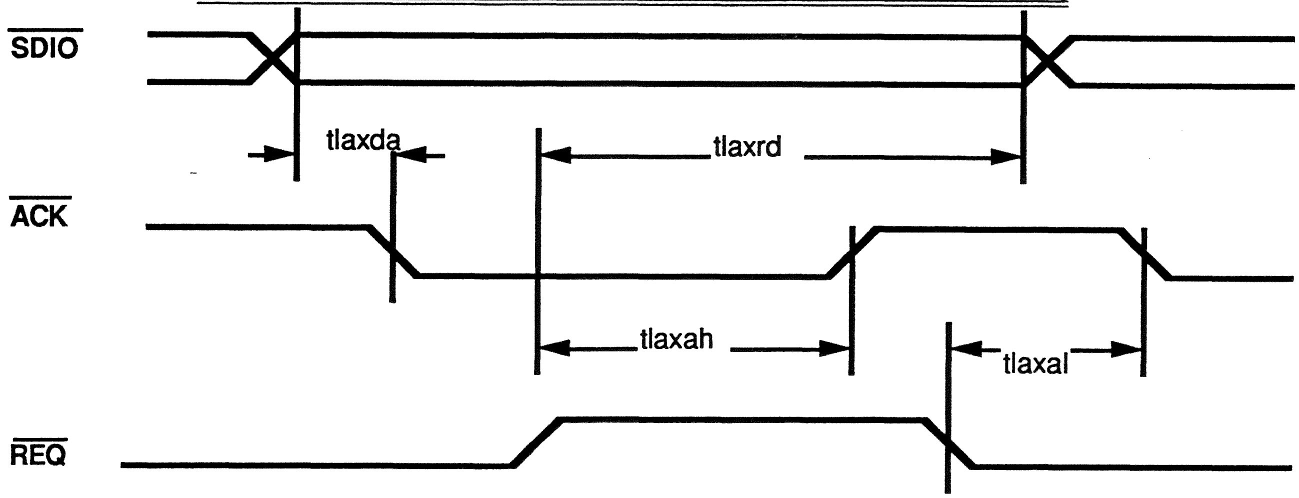

**Burst Mode DMA Read**

| Burst Mode DMA Interface | | | | | |
|---|---|---|---|---|---|
| Symbol | Description | Min | Max | Units | notes |
| tdan1 | SDACK# high to SDREQ high | | 40 | ns | 3 |
| tdan2 | SDACK# low to SDREQ low | | 45 | ns | 1 |
| tdan3 | SDACK# high to SDACK# low | 60 | | ns | |
| tdan4 | SDACK# pulse width | 100 | | ns | |
| tdan5 | RD16# high to SDREQ low | | 140 | ns | 2 |
| tdan6 | SDACK# low to RD16# low | 0 | | ns | |
| tdan7 | RD16# pulse width | 70 | | ns | |
| tdan8 | RD16#high to RD16# low | 60 | | ns | |
| tdan9 | RD16# low to data valid | 70 | | ns | |
| tdan10 | Data release time | 2 | | ns | |
| tdan11 | RD16# low to RD16# low | 130 | | ns | |
| tdan12 | RD16# high to RD16# high | 130 | | ns | |
| tdan13 | WR16# high to SDREQ low | | 140 | ns | 2 |
| tdan14 | SDACK# low to WR16# low | 0 | | ns | |
| tdan15 | WR16# pulse width | 100 | | ns | |
| tdan16 | WR16# high to WR16# low | 60 | | ns | |
| tdan17 | Data setup to WR16# high | 15 | | ns | |
| tdan18 | Data hold from WR16# high | 0 | | ns | |
| tdan19 | WR16# low to WR16# low | 160 | | ns | |
| tdan20 | WR16# high to WR16# high | 160 | | ns | |

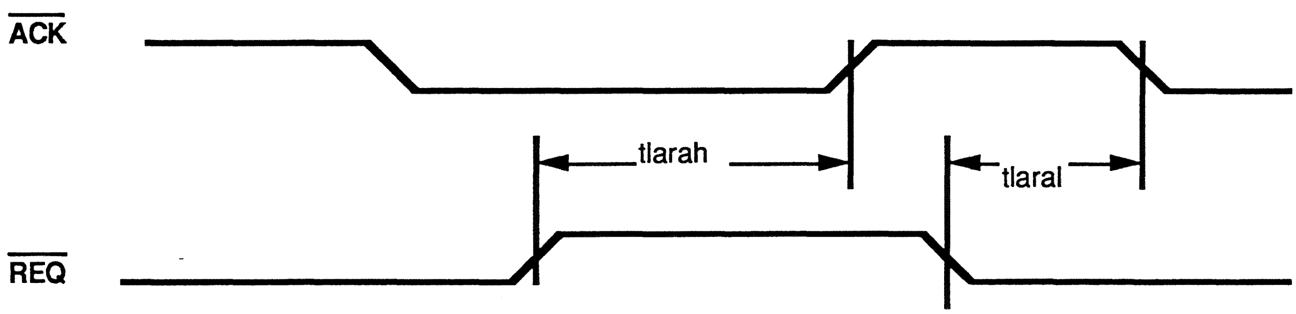
Notes:

1. tdan2 applies to single DMA transfers only.
2. These parameters apply to multiple DMA transfers.
3. SDREQ is deasserted if FIFO is empty during DMA read or if FIFO is full during DMA write.

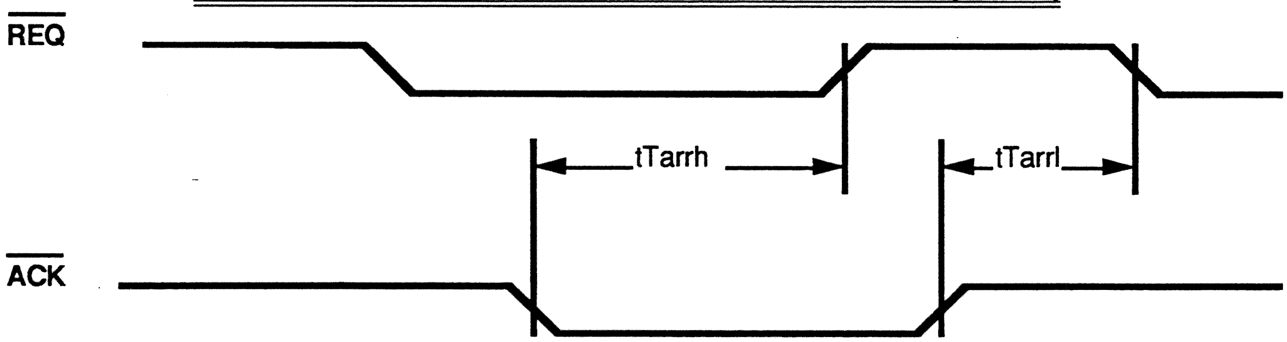
| Initiator Asynchronous Send | | | | |
|---|---|---|---|---|
| Symbol | Description | Min | Max | Unit |
| tlaxda | Data to ACK# low | 55 | | ns |
| tlaxah | REQ# high to ACK# high | | 46 | ns |
| tlaxrd | REQ# high to data (FIFO bottom full) | | 80 | ns |
| tlaxal | REQ# low to ACK# low (data already setup) | | 55 | ns |



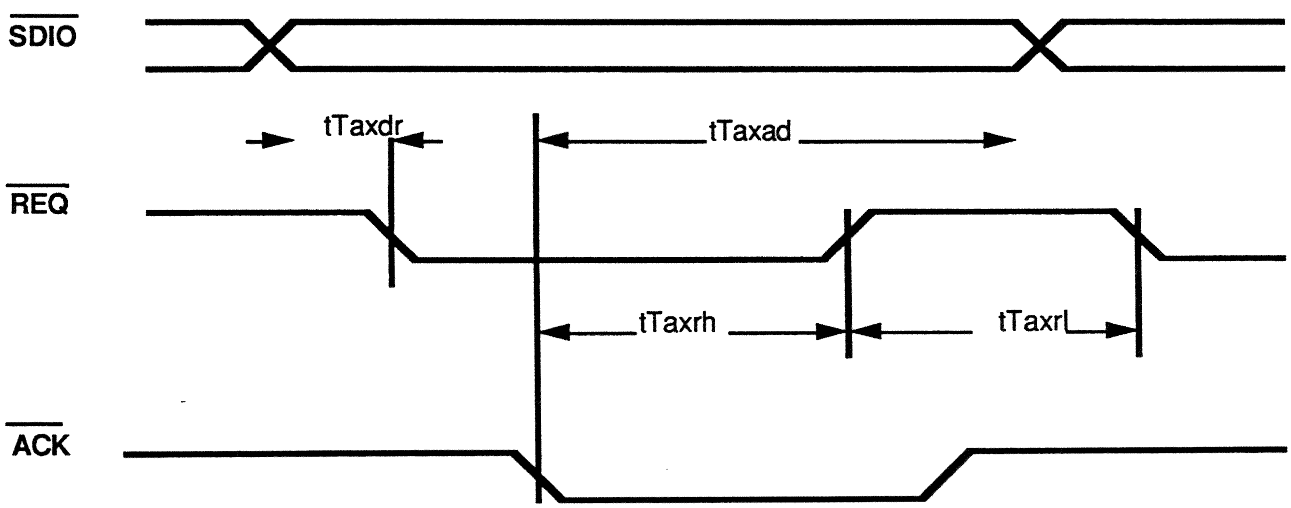

| Initiator Asynchronous Receive | | | | |
|---|---|---|---|---|
| Symbol | Description | Min | Max | Unit |
| tlarah | REQ# high to ACK# high | | 43 | ns |
| tlaral | REQ# low to ACK# low (FIFO not full) | | 47 | ns |



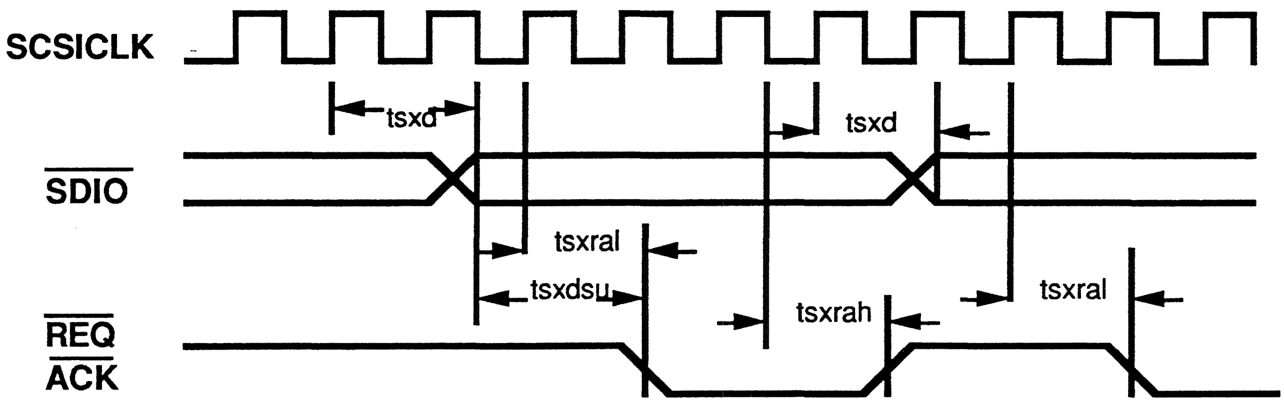

| Target Asynchronous Receive | | | | |
|---|---|---|---|---|
| Symbol | Description | Min | Max | Unit |
| tTarrh | ACK# low to REQ# high | | 43 | ns |
| tTarrl | ACK# high to REQ# low (FIFO not full) | | 45 | ns |




| Target Asynchronous Send | | | | |
|---|---|---|---|---|
| Symbol | Description | Min | Max | Unit |
| tTaxdr | Data to REQ# low | 55 | | ns |
| tTaxdh | ACK# low to REQ# high | | 43 | ns |
| tTaxad | ACK# low to data (FIFO bottom full) | | 78 | ns |
| tTaxrl | ACK# high to REQ# low (data already set up) | | 45 | ns |




| Target and Initiator Synchronous Transmit | | | | |
|---|---|---|---|---|
| Symbol | Description | Min | Max | Unit |
| tsxd | Data from SCSICLK high | 20 | 90 | ns |
| tsxral | REQ# or ACK# low from SCSICLK high | 15 | 68 | ns |
| tsxrah | REQ# or ACK# high from SCSICLK low | 17 | 70 | ns |
| tsxdsu | Data setup to ACK# or REQ# low | 55 | | ns |




**Synchronous Transfers**

### 4.2.3 ESCC AC Timing

**General Timing**

| No. | Parameter Symool | Parameter Description | 16.384 MHz Min. | 16.384 MHz Max. | Unit |
|---|---|---|---|---|---|
| 1 | TdPC(REQ) | PCLK ↓ to W#/REQ# Valid Delay | | 80 | ns |
| 2 | TdPC(W) | PCLK ↓ to Wait Inactive Delay | | 180 | ns |
| 3 | TsRXC(PC) | RxC# ↑ to PCLK ↑ Setup Time (Notes 1,4, & 8) | NA | NA | |
| 4 | TsRXD(RXCr) | RxD to RxC# ↑ Setup Time (XI Mode) (Note 1) | 0 | | ns |
| 5 | ThRXD(RXCr) | RxD to RxC# ↑ Hold Time (XI Mode) (Note 1) | 50 | | ns |
| 6 | TsRXD(RXCf) | RxD to RxC# ↓ Setup Time (XI Mode) (Notes 1, 5) | 0 | | ns |
| 7 | ThRXD(RXCf) | RxD to RxC# ↓ Hold Time (XI Mode) (Notes 1, 5) | 50 | | ns |
| 8 | TsSY(RXC) | SYNC# to RxC# ↑ Setup Time (Note 1) | -100 | | ns |
| 9 | ThSY(RXC) | SYNC to RxC# ↑ Hold Time (Note 1) | 5TcPc | | ns |
| 10 | TsTXC(PC) | TxC# ↓ to PCLK ↑ Setup Time (Notes 2, 4 & 8) | NA | NA | |
| 11 | TdTXCf(TXD) | TxC# ↓ to TxD Delay (XI Mode) (Note 2) | | 80 | ns |
| 12 | TdTXCr(TXD) | TxC# ↑ to TxD Delay (XI Mode) (Notes 2, 5) | | 80 | ns |
| 13 | TdTXD(TRX) | TxD to TRxC# Delay (Send Clock Echo) | | 80 | ns |
| 14a | TwRTXh | RTxC# High Width (Note 6) | 80 | | ns |
| 14b | TwRTXh(E) | RTxC# High Width (Note 9) | 15.6 | | ns |
| 15a | TwRTXl | RTxC# Low Width (Note 6) | 80 | | ns |
| 15b | TwRTXl(E) | RTxC# Low Width (Note 9) | 15.6 | | ns |
| 16a | TcRTX | RTxC# Cycle Time (Notes 6, 7) | 244 | | ns |
| 16b | TcRTx(E) | RTxC# Cycle Time (Note 9) | 31.25 | | ns |
| 17 | TcRTXX | Crystal Oscillator Period (Note 3) | 62 | 1000 | ns |
| 18 | TwTRXh | TRxC# High Width (Note 6) | 80 | | ns |
| 19 | TwTRXl | TRxC# Low Width (Note 6) | 80 | | ns |
| 20 | TcTRX | TRxC# Cycle Time (Notes 6, 7) | 244 | | ns |
| 21 | TwEXT | DCD# or CTS# Pulse Width | 70 | | ns |
| 22 | TwSY | SYNC# Pulse Width | 70 | | ns |

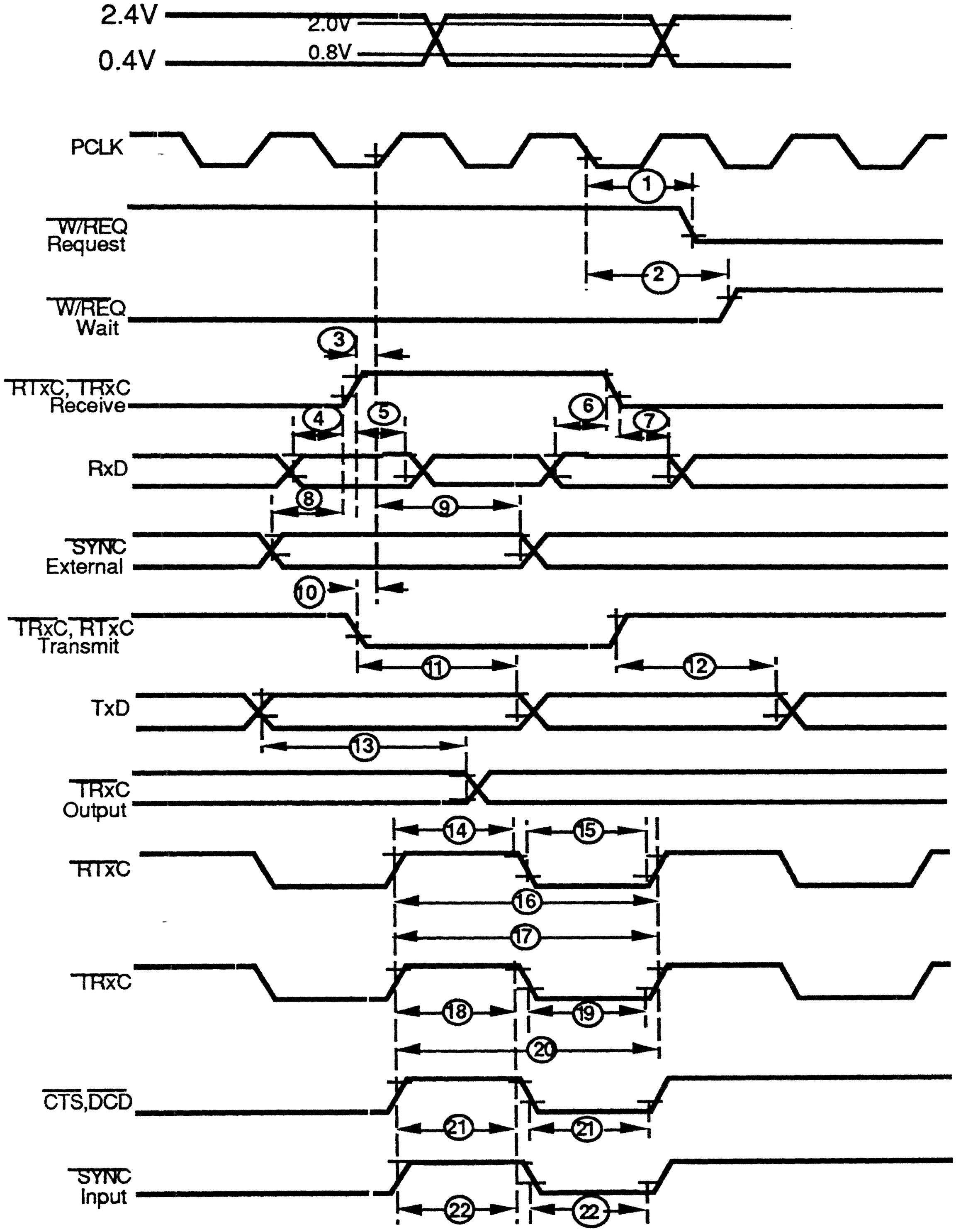
Notes:
1. RxC# is RTxC# or TRxC#, whichever is supplying the receive clock.
2. TxC# is TRxC# or RTxC#, whichever is supplying the transmit clock.
3. Both RTxC#and SYNC# have 30-pF capacitors to ground connected to them.

4. Parameter applies only if the data rate is one-fourth the PCLK rate. In all other cases, no phase relationship between RxC# and PCLK or TxC# and PCLK is required.
5. Parameter applies only to FM encoding/decoding.
6. Parameter applies only for transmitter and receiver; DPLL and baud rate generator timing requirements are identical to chip PCLK requirements.
7. The maximum receive or transmit data is 1/4 PCLK.
8. External PCLK to RxC# or TxC# synchronization requirement eliminated for PCLK divide-by-four operation.

TRxC# and RTxC# rise and fall times are identical to PCLK. Reference timing specs Tfpc and Trpc.

Tx and Rx input clock slow rates should be kept to a maximum of 30 ns. All parameters related to input PCLK edges should be referenced at the point at which the transition begins or ends, whichever is the worst case.

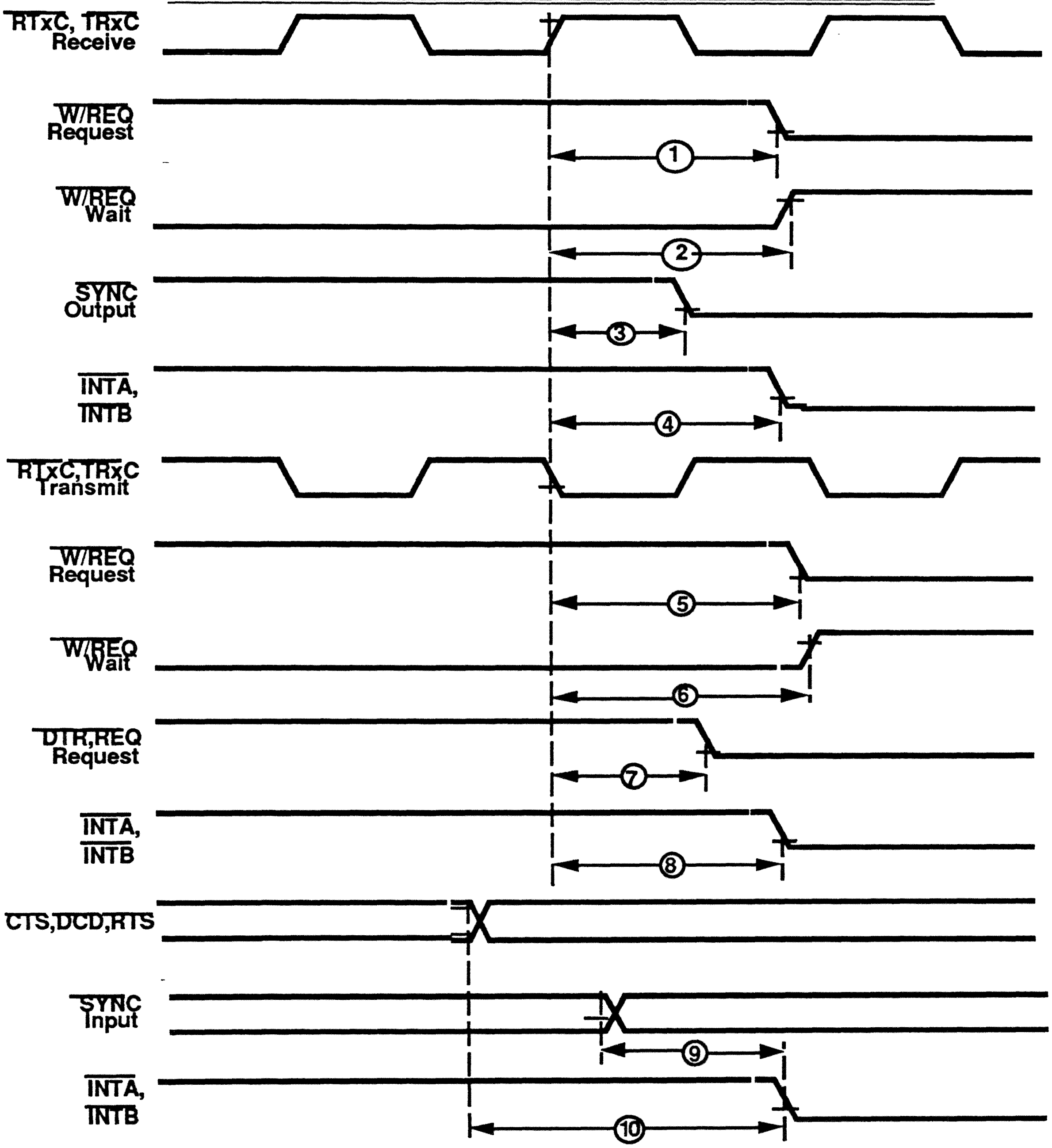
9. ENHANCED FEATURE—RTXC# used as input to internal DPLL only.


2.4V
2.0V
0.8V
0.4V
PCLK
W/REQ
Request
W/REQ
Wait
RTxC, TRxC
Receive
RxD
SYNC
External
TRxC, RTxC
Transmit
TxD
TRxC
Output
RTxC
TRxC
CTS,DCD
SYNC
Input
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22

**ESCC SWITCHING CHARACTERISTICS over COMMERCIAL operating range**
**System Timing**

| No. | Parameter Symbol | Parameter Description | 16.384 MHz Min. | 16.384 MHz Max. | Units |
|---|---|---|---|---|---|
| 1 | TdRXC(REQ) | RXC# ↑ W#/REQ# Valid Delay (Note 2) | 8 | 12 | TcPc |
| 2 | TdRXC(W) | RXC# ↑ to Wait Inactive Delay (Notes 1, 2) | 8 | 14 | TcPc |
| 3 | TdRXC(SY) | RXC# ↑ to SYNC# Valid Delay (Note 2) | 4 | 7 | TcPc |
| 4 | TdRXC(INT) | RXC# ↑ to INT# Valid Delay (Notes 1, 2) | 10 | 16 | TcPc |
| 5 | TdTXC(REQ) | TxC# ↓ to W#/REQ# Valid Delay (Note 3) | 5 | 8 | TcPc |
| 6 | TdTXC(W) | TxC# ↓ to Wait Inactive Delay (Notes 1, 3) | 5 | 11 | TcPc |
| 7a | TdTXC(DRQ) | TxC# ↓ to DTR#/REQ# Valid Delay (Note 3) | 4 | 7 | TcPc |
| 7b | TdTXC(EDRQ) | TxC# ↓ to DTR#/REQ# Valid Delay (Notes 3, 4) | 5 | 8 | TcPc |
| 8 | TdTXC(INT) | TxC# ↓ to INT# Valid Delay (Notes 1, 3) | 6 | 10 | TcPc |
| 9 | TdSY(INT) | SYNC# Transition to INT# Valid Delay (Note 1) | 2 | 6 | TcPc |
| 10 | TdEXT(INT) | DCD# or CTS# Transition to INT# Valid Delay (Note 1) | 2 | 6 | TcPc |

Notes:
1. Open-drain output, measured with open-drain test load.
2. RxC# is RTxC# or TRxC#, whichever is supplying the receive clock.
3. TxC# is TRxC# or RTxC#, whichever is supplying the transmit clock.
4. Parameter applies to Enhanced Request mode only.


RTxC, TRxC
Receive
W/REQ
Request
1
W/REQ
Wait
2
SYNC
Output
3
INTA,
INTB
4
RTxC,TRxC
Transmit
W/REQ
Request
5
W/REQ
Wait
6
DTR,REQ
Request
7
INTA,
INTB
8
CTS,DCD,RTS
SYNC
Input
9
INTA,
INTB
10

## OBJECTIVE SPECIFICATION - ADVANCED COMBO (CURIO)

**ESCC SWITCHING CHARACTERISTICS over COMMERCIAL operating range**
**Read and Write Timing**

| No. | Parameter Symbol | Parameter Description | 16.384 MHz Min. | 16.384 MHz Max. | Unit |
|---|---|---|---|---|---|
| 1 | TwPCl | PCLK Low Width | 26 | 2000 | ns |
| 2 | TwPCh | PCLK High Width | 26 | 2000 | ns |
| 3 | TfPC | PCLK Fall Time | | 8 | ns |
| 4 | TrPC | PCLK Rise Time | | 8 | ns |
| 5 | TcPC | PCLK Cycle Time | 61 | 4000 | ns |
| 6 | TsA(WR) | Address to WR8# ↓ Setup Time | 35 | | ns |
| 7 | ThA(WR) | Address to WR8# ↑ Hold Time | 0 | | ns |
| 8 | TsA(RD) | Address to RD8# ↓ Setup Time | 35 | | ns |
| 9 | ThA(RD) | Address to RD8# ↑ Hold Time | 0 | | ns |
| 10 | TsIA(PC) | INTACK# to PCLK ↑ Setup Time | 15 | | ns |
| 11 | TsIAi(WR) | INTACK# to WR8# ↓ Setup Time (Note 1 ) | 70 | | ns |
| 12 | ThIA(WR) | INTACK# to WR8# ↑ Hold Time | 0 | | ns |
| 13 | TsIAi(RD) | INTACK# to RD8# ↓ Setup Time (Note 1 ) | 70 | | ns |
| 14 | ThIA(RD) | INTACK# to RD8# ↑ Hold Time | 0 | | ns |
| 15 | ThIA(PC) | INTACK# to PCLK ↑ Hold Time | 15 | | ns |
| 16 | TsCEl(WR) | CE1# Low to WR8# ↓ Setup Time | 0 | | ns |
| 17 | ThCE(WR) | CE1# to WR8# ↑ Hold Time | 0 | | ns |
| 18 | TsCEh(WR) | CE1# High to WR8# ↓ Setup Time | 30 | | ns |
| 19 | TsCEl(RD) | CE1# Low to RD8# ↓ Setup Time (Note 1) | 0 | | ns |
| 20 | ThCE(RD) | CE1# to RD8# ↑ Hold Time (Note1 ) | 0 | | ns |
| 21 | TsCEh(RD) | CE1# High to RD8# ↓ Setup Time (Note 1) | 30 | | ns |
| 22 | TwRDl | RD8# Low Width (Note 1 ) | 75 | | ns |
| 23 | TdRD(DRA) | RD8# ↓ to Read Data Active Delay | 0 | | ns |
| 24 | TdRDr(DR) | RD8# ↑ to Read Data Not Valid Delay | 0 | | ns |
| 25 | TdRDf(DR) | RD8# ↓ to Read Data Valid Delay | | 70 | ns |
| 26 | TdRD(DRz) | RD8# ↑ to Read Data Float Delay (Note 2) | | 20 | ns |

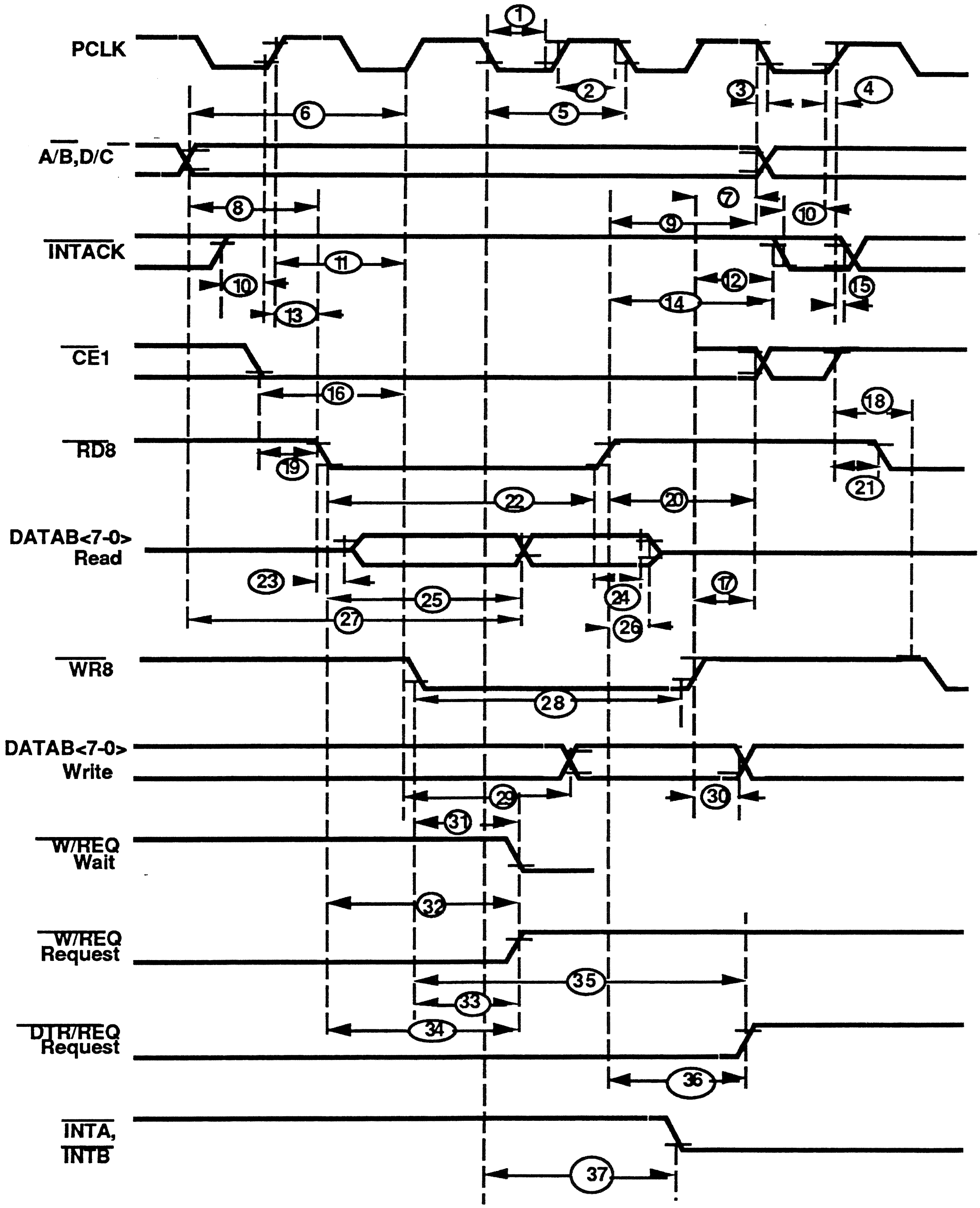
Notes: 1. Parameter does not apply to Interrupt Acknowledge transactions.
2. Float delay is defined as the time at which the data bus is released from its drive state with a maximum DC load and minimum AC load.

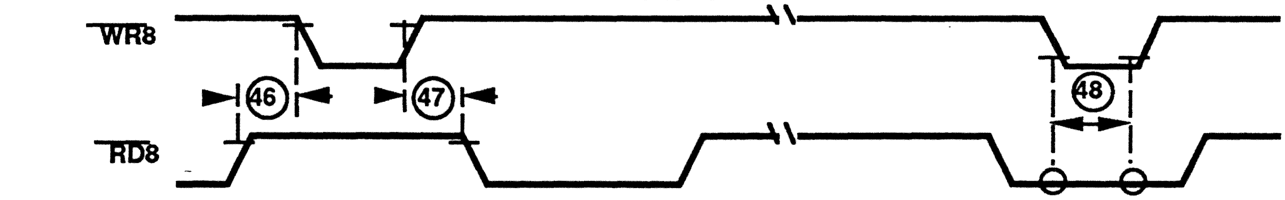
## OBJECTIVE SPECIFICATION - ADVANCED COMBO (CURIO)

**ESCC SWITCHING CHARACTERISTICS over COMMERCIAL operating range**
**Interrupt Acknowledge Timing, Reset Timing, Cycle Timing**

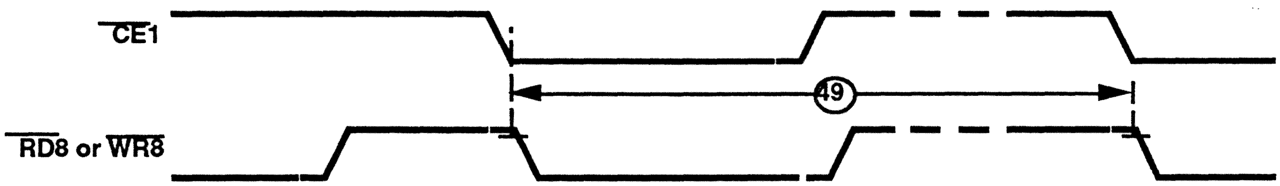
| No. | Parameter Symbol | Parameter Description | 16.384 MHz Min. | 16.384 MHz Max. | Unit |
|---|---|---|---|---|---|
| 27 | TdA(DR) | Address Required Valid to Read Data Valid Delay | | 100 | ns |
| 28 | TwWRl | WR8# Low Width | 75 | | ns |
| 29 | TdWRf(DW) | WR8# ↓ to Write Data Valid | | 20 | ns |
| 30 | ThDW(WR) | Write Data to WR8# ↑ Hold Time | 0 | | ns |
| 31 | TdWR(W) | WR8# ↓ to Wait Valid Delay (Note 2) | | 50 | ns |
| 32 | TdRD(W) | RD8# ↓ to Wait Valid Delay (Note 2) | | 50 | ns |
| 33 | TdWRf(REQ) | WR8# ↓ to W#/REQ# Not Valid Delay | | 70 | ns |
| 34 | TdRDf(REQ) | RD8# ↓ to W#/REQ# Not Valid Delay | | 70 | ns |
| 35a | TdWRr(REQ) | WR8# ↓ to DTR#/REQ# Not Valid Delay | | 4.0TcPc | ns |
| 35b | TdWRr(EREQ) | WR8# ↓ to DTR#/REQ# Not Valid Delay (Note 4) | | 70 | ns |
| 36 | TdRDr(REQ) | RD8# ↑ to DTR#/REQ# Not Valid Delay | | NA | ns |
| 37 | TdPC(INT) | PCLK ↓ to INTA#, INTB# Valid Delay (Note 2) | | 175 | ns |
| 38 | TdIAi(RD) | INTACK# to RD8# ↓ (Acknowledge) Delay (Note 3) | 50 | | ns |
| 39 | TwRDA | RD8# (Acknowledge) Width | 75 | | ns |
| 40 | TdRDA(DR) | RD8# ↓ (Acknowledge) to Read Data Valid Delay | | 70 | ns |
| 45 | TdRDA(INT) | RD8# ↓ to INTA#, INTB# Inactive Delay (Note 2) | | 200 | ns |
| 46 | TdRD(WRQ) | RD8# ↑ to WR8# ↓ Delay for No Reset | 10 | | ns |
| 47 | TdWRQ(RD) | WR8# ↑ to RD8# ↓ Delay for No Reset | 10 | | ns |
| 48 | TwRES | WR8# and RD8# Coincident Low for Reset | 75 | | ns |
| 49 | Trc | Valid Access Recovery Time (Note 1) | 3.5 | | TcPc |

Notes:
1. Parameter applies only between transactions involving the ESCC. If WR8#/RD8# falling edge is synchronized to PCLK falling edge, then TrC = 3TcPc.
2. Open-drain output, measured with open-drain test load.
3. Parameter is system dependent. For any SCC in the daisy chain, TdIAi(RD) must be greater than the sum of TdPC(IEO) for the highest priority device in the daisy chain, TsIEI(RDA) for the SCC, and TdIEI(IEO) for each device separating them in the daisy chain.
4. Parameter applies to Enhanced Request mode only.

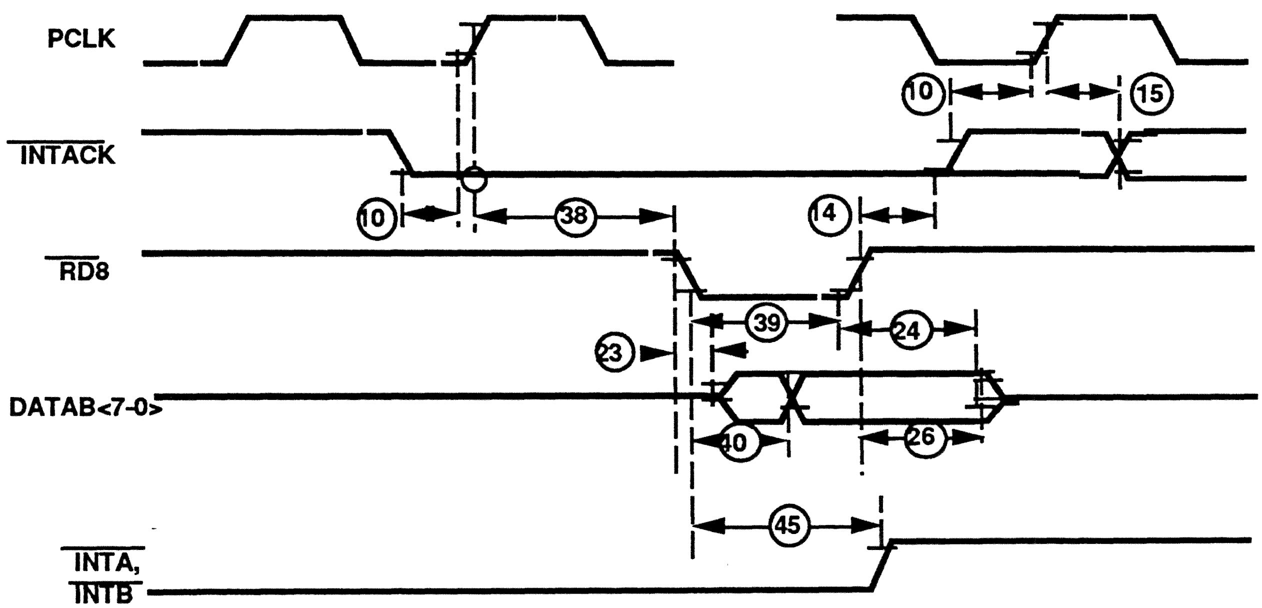

PCLK
A/B,D/C
INTACK
CE1
RD8
DATAB<7-0>
Read
WR8
DATAB<7-0>
Write
W/REQ
Wait
W/REQ
Request
DTR/REQ
Request
INTA,
INTB
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37




Reset Timing



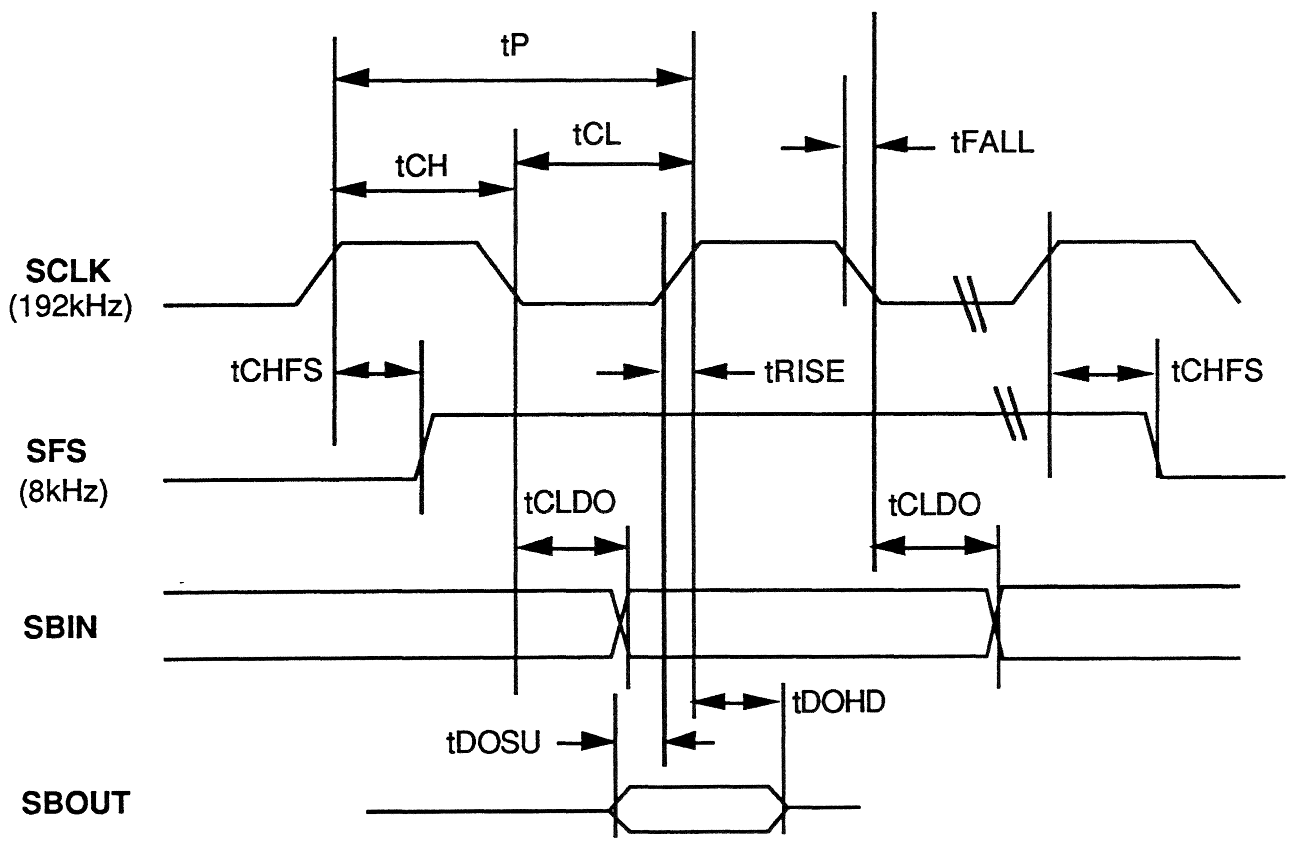

Cycle Timing




Interrupt Acknowledge Timing

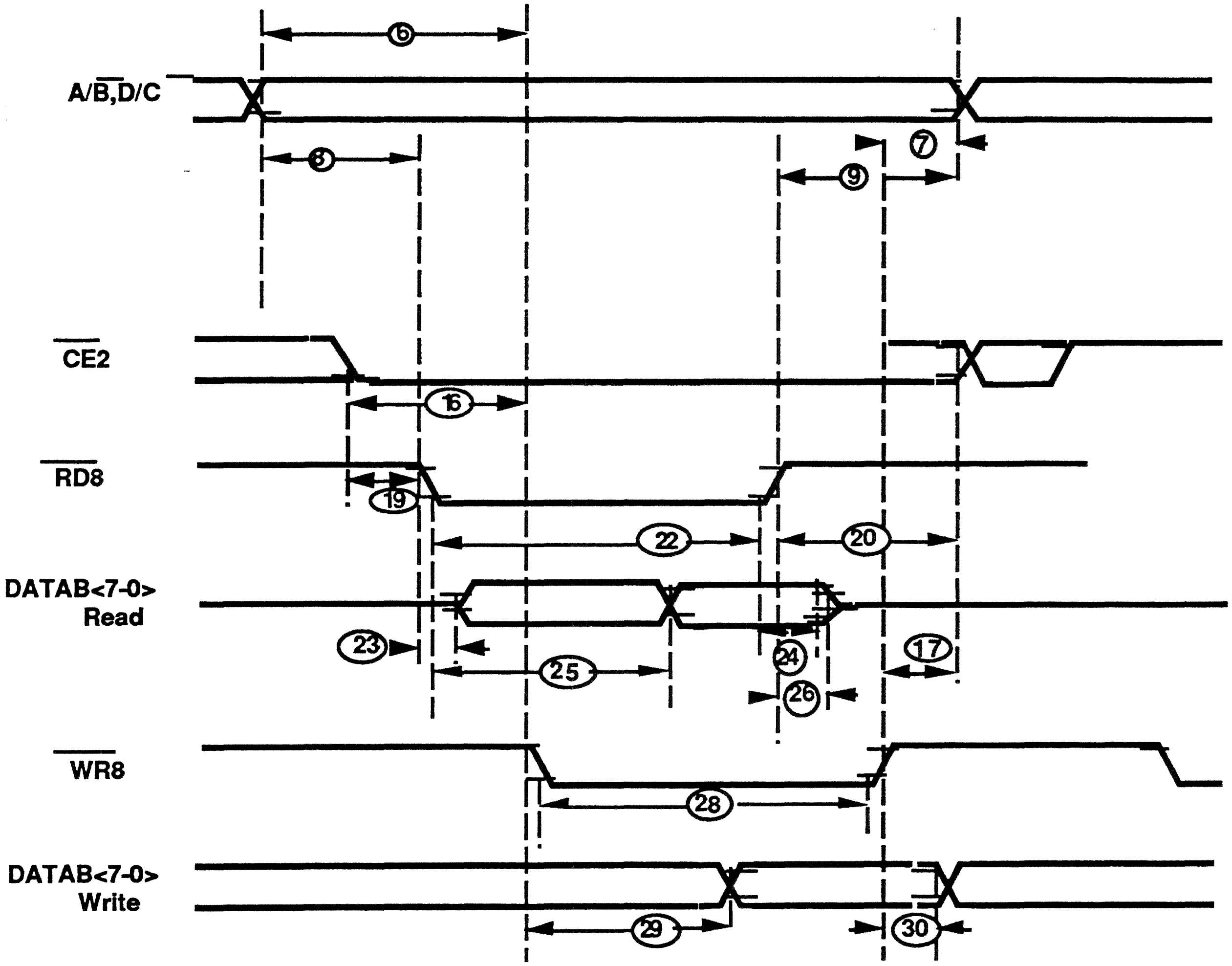
### 4.2.4 SBP AC Timing

| # | PARAM | DESCRIPTION | TEST CONDITIONS | MIN | MAX | Unit |
|---|---|---|---|---|---|---|
| | tP | SCLK period | | 4.9 | 5.5 | µs |
| | tCH | SCLK high time | | 2.5 | 2.7 | µs |
| | tCL | SCLK low time | | 2.4 | 2.8 | µs |
| | tRISE | SCLK rise time | | | 20 | ns |
| | tFALL | SCLK fall time | | | 20 | ns |
| | tCHFS | SCLK high to frame sync | | 40 | 260 | ns |
| | tCLDO | SCLK low to data out | | 50 | 250 | ns |
| | tDOSU | SBOUT setup time to SCLK | | 200 | | ns |
| | tDOHD | SBOUT hold time from SCLK | | 0 | | ns |



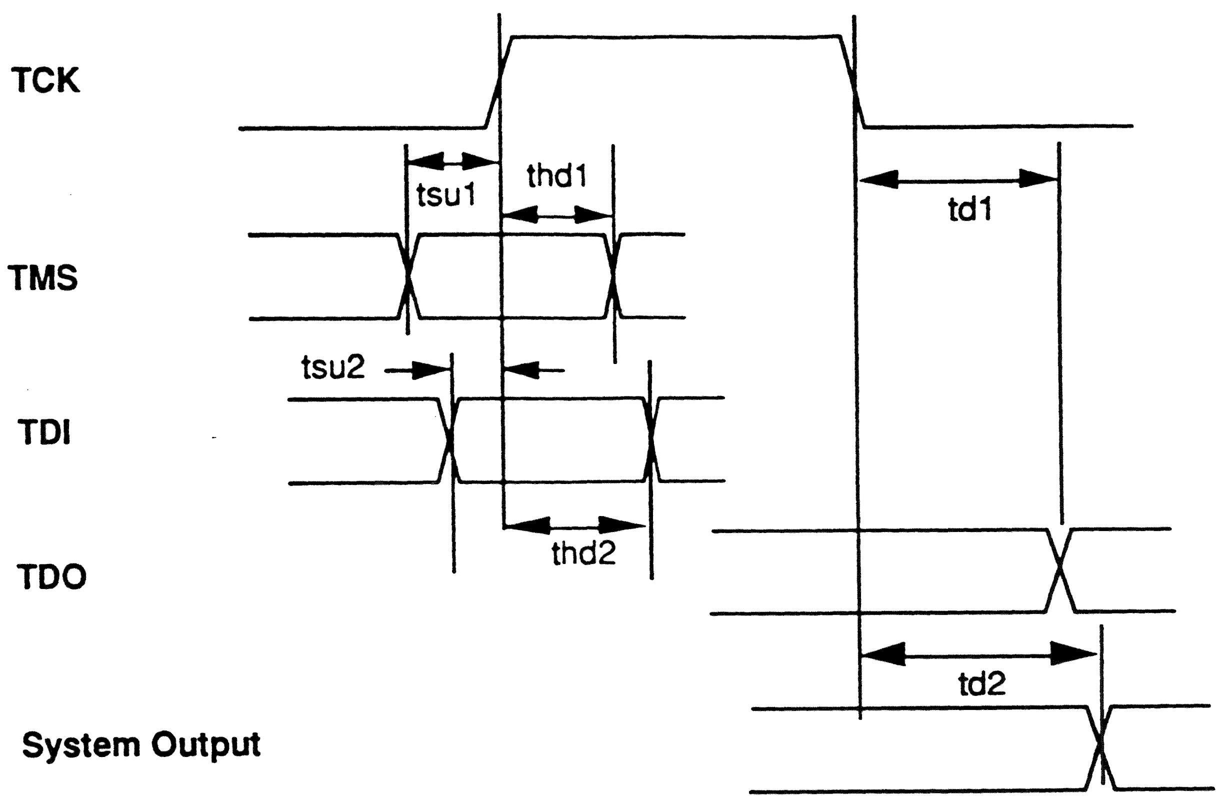
| No. | Parameter Symbol | Parameter Description | 16.384 MHz Min. | Max. | Unit |
|---|---|---|---|---|---|
| 6 | TsA(WR) | Address to WR8# ↓ Setup Time | 35 | | ns |
| 7 | ThA(WR) | Address to WR8# ↑ Hold Time | 0 | | ns |
| 8 | TsA(RD) | Address to RD8# ↓ Setup Time | 35 | | ns |
| 9 | ThA(RD) | Address to RD8# ↑ Hold Time | 0 | | ns |
| 16 | TsCEI(WR) | CE2# Low to WR8# ↓ Setup Time | 0 | | ns |
| 17 | ThCE(WR) | CE2# to WR8# ↑ Hold Time | 0 | | ns |
| 19 | TsCEI(RD) | CE2# Low to RD8# ↓ Setup Time (Note 1) | 0 | | ns |
| 20 | ThCE(RD) | CE2# to RD8# ↑ Hold Time (Note1 ) | 0 | | ns |
| 22 | TwRDI | RD8# Low Width (Note 1 ) | 75 | | ns |
| 23 | TdRD(DRA) | RD8# ↓ to Read Data Active Delay | 0 | | ns |
| 24 | TdRDr(DR) | RD8# ↑ to Read Data Not Valid Delay | 0 | | ns |
| 25 | TdRDf(DR) | RD8# ↓ to Read Data Valid Delay | | 70 | ns |
| 26 | TdRD(DRz) | RD8# ↑ to Read Data Float Delay (Note 2) | | 20 | ns |
| 28 | TwWRI | WR8# Low Width | 75 | | ns |
| 29 | TdWRf(DW) | WR8# ↓ to Write Data Valid | | 20 | ns |
| 30 | ThDW(WR) | Write Data to WR8# ↑ Hold Time | 0 | | ns |

Notes: 1. Parameter does not apply to Interrupt Acknowledge transactions.
2. Float delay is defined as the time at which the data bus is released from its drive state with a maximum DC load and minimum AC load.


A/B,D/C
CE2
RD8
DATAB<7-0>
Read
WR8
DATAB<7-0>
Write
6
8
7
9
16
19
22
20
23
25
24
26
17
28
29
30

### 4.2.5 IEEE/JTAG P1149.1 Port AC Timing

| # | PARAM | DESCRIPTION | TEST CONDITIONS | MIN | MAX | Unit |
|---|---|---|---|---|---|---|
| | tTCK | TCK frequency, 50% duty cycle (±5%). | | | 10 | MHz |
| | tsu1 | SETUP time, TMS to TCK rising edge | | 8 | | ns |
| | tsu2 | SETUP time, TDI to TCK rising edge | | 0 | | ns |
| | thd1 | HOLD time, TMS to TCK rising edge | | 2 | | ns |
| | thd2 | HOLDTIME, TDI to TCK rising edge | | 10 | | ns |
| | td1 | PROP delay, TDO after TCK falling edge | | | 30 | ns |
| | td2 | PROP_DELAY, SYSTEM OUTPUT after TCK falling edge | | | 35 | ns |




- END -