National Semiconductor

DP8490 Enhanced Asynchronous SCSI Interface (EASI)

General Description

The DP8490 EASI is a CMOS device designed to provide a low cost, high performance Small Computer Systems Interface. It complies with the ANS X3.131-1986 SCSI standard as defined by the ANSI X3T9.2 committee. It can act as both INITIATOR and TARGET, making it suitable for any application. The EASI supports selection, reselection, arbitration and all other bus phases. High-current open-drain drivers on chip reduce application chip count by interfacing direct to the SCSI bus. An on-chip oscillator provides all timing delays.

The DP8490 is pin and program compatible with the NMOS NCR5380 and CMOS DP5380 devices. NCR5380, DP5380 or AM5380 applications should be able to use it with no changes to hardware or software. The DP8490 includes new features which make this part more attractive for new designs and performance upgrades. These new features include μP data bus parity, programmable parity for both SCSI and μP busses, loopback test mode, improved arbitration support, faster timing and extended interrupt control logic. The DP8490 is available in a 40-pin DIP or a 44-pin PCC.

The EASI is intended to be used in a microprocessor based application, and achieves maximum performance with a DMA controller. The device is controlled by reading and writing several internal registers. A standard non-multiplexed address and data bus easily fits any μP environment.

Data transfers can be performed by programmed-I/O, pseudo-DMA or via a DMA controller. The EASI easily interfaces to a DMA controller using normal or Block Mode. The EASI can be used in either a polled or interrupt-driven environment. The EASI includes enhanced features for interrupt control.

Features

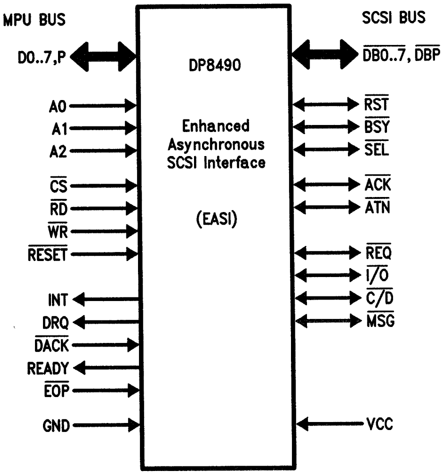
SCSI Interface

- Supports TARGET and INITIATOR roles
- Parity generation with optional checking
- Programmable parity polarity (ODD/EVEN)
- Arbitration support—can interrupt when done
- Direct control/monitoring of all SCSI signals
- High current outputs drive SCSI bus directly
- Faster and improved timing
- Very low SCSI bus loading

μP Interface

- Memory or I/O-mapped control transfers
- Programmed-I/O or DMA data transfers
- Normal or Block-mode DMA
- Fast DMA handshake timing
- Individually maskable interrupts
- Active interrupts identified in one register
- Optional data bus parity generation/checking
- Programmable parity polarity (ODD/EVEN)
- Loopback test mode

Connection Diagram



TL/F/9387-1

Table of Contents

1.0 Functional Description

1.1 OVERVIEW

The EASI is designed to be used as a peripheral device in a μP-based application and appears as a number of read/write registers. Write registers are programmed to select desired functions. Status registers provide indication of operating conditions. In an application extensive use of interrupts is desirable. The EASI incorporates an improved interrupt structure which enables fully interrupt-driven operation. In the enhanced mode interrupts can be individually masked or enabled, and a status register identifies all active interrupt requests.

For best performance a DMA controller can be easily interfaced directly to the EASI. The EASI provides request/acknowledge and wait-state signals for the DMA interface.

The SCSI bus is easily controlled via the EASI registers. Any bus signal may be asserted or deasserted via a bit in the appropriate register, and the state of every signal is available by reading registers. This direct control over SCSI signals allows the user to implement all or part of the protocol in firmware. The EASI provides hardware support for much of the protocol, and all speed-critical steps are handled by the EASI.

The EASI provides the following SCSI support:

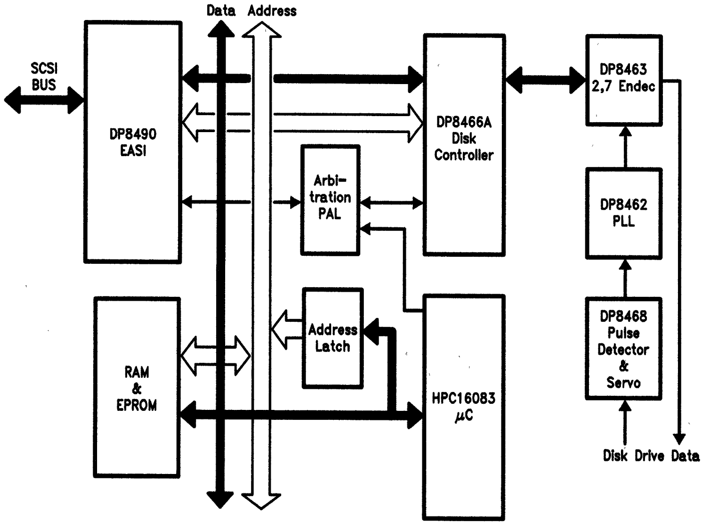
- Programmed-I/O transfers for all eight information transfer types, with or without parity.
- Data transfers via DMA, in either block or non-block mode. The DMA interface supports most devices.
- Individual setting/resetting and monitoring of every SCSI bus signal.
- Automatic release of the bus for BSY loss from a TARGET, SCSI RST, and lost arbitration.
- Automatic bus arbitration with an optional interrupt upon completion—the μP has only to check for highest priority. The 2.2 μs arbitration delay can be optionally performed by the EASI.
- Selection or Reselection of any bus device. The EASI will respond to both Selection and Reselection.
- Optional automatic monitoring of the $\overline{\text{BSY}}$ signal from a TARGET with an interrupt after releasing control of the bus.
- Optional parity polarity selection. Default after reset is ODD, but EVEN generation and checking can be programmed for diagnostic purposes and to determine whether a device supports parity when first making a connection.

Figure 1 shows an EASI in a typical application, a low cost embedded SCSI disk controller. In this application the 8051 single-chip μP acts as the controller and the dual DMA channels in the DP8475 allow one for the disk data and the other for SCSI data. The PAL provides chip selection as well as determining who has control of the bus. The advantage of using a μP with on-board ROM is that there is more free time on the external bus.

1.2 μP INTERFACE

Figure 2 shows a block diagram of the EASI. Key blocks within the EASI are Read/Write registers with associated decode and control logic, interrupt and DMA logic, SCSI bus arbitration logic, SCSI drivers/receivers with parity and the SCSI data input and output registers. The EASI has three interfaces, one to SCSI, one to a DMA controller and the third to a μP. The internal registers control all operation of the EASI.

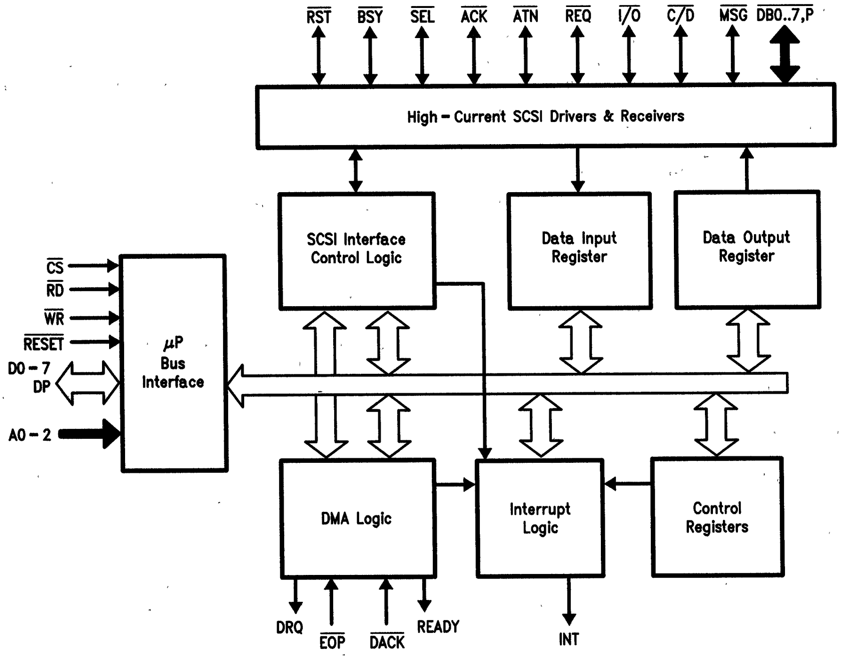
The μP interface consists of non-multiplexed address and data busses with associated control signals. The data bus can be programmed to use either ODD or EVEN parity. Address decode logic selects a register for reading or writing. The address lines A0–A2 select the register for μP accesses while for DMA accesses the address lines are ignored. The decode logic also selects different registers or functions to be mapped into address 7, according to the programmed mode (see Section 8).



TL/F/9387-2

FIGURE 1. EASI Application

1.0 Functional Description (Continued)



TL/F/9387-3

FIGURE 2. EASI Block Diagram

The register bank consists of twelve registers mapped into an address space of eight locations. Upon an external chip reset the registers are cleared (all zeroes)—the same as the NCR5380. Once the ENHANCED MODE bit in the INITIATOR COMMAND REGISTER is set, three new registers can be accessed to utilize the extra features of the DP8490 EASI.

1.3 DMA INTERFACE

The DMA logic interfaces to single-cycle, block mode, flow-through or fly-by controllers. Single byte transfers are accomplished via the DRQ/$\overline{\text{DACK}}$ handshake signals. Block mode transfers use the READY output to control the speed (insert wait-states). An End Of Process ($\overline{\text{EOP}}$) input from the DMA controller signals the EASI to halt DMA transfers. An interrupt can be generated for DMA completion or an error (see Section 5). All DMA data passes through the SCSI data input and output registers, automatically selected during DMA cycles.

1.4 SCSI INTERFACE

The EASI contains all logic required to interface directly to the SCSI bus. Direct control and monitoring of all SCSI signals is provided. The state of each SCSI signal may be determined by reading a register which continuously reflects the state of the bus. Each signal may be asserted by writing a ONE to the appropriate bit.

The EASI includes logic to automatically handle SCSI timing sequences too fast for μP control. In particular there is hardware support for DMA transfers, bus arbitration, selection/reselection, bus phase monitoring, $\overline{\text{BSY}}$ monitoring for bus disconnection, bus reset and parity generation and checking.

The EASI arbitration logic controls arbitration for use of the SCSI bus. The μP programs the SCSI device ID into the EASI, then sets the ARBITRATE bit. The EASI will interrupt the μP when one of three events occurs: arbitration is lost; arbitration has completed and the ID priorities need to be checked; or arbitration is complete and the 2.2 μs SCSI Arbitration delay has expired. Arbitration can be invoked with the enhanced feature of an interrupt on completion or the expiration of the SCSI Arbitration delay. These extra steps are programmed via the EXTRA MODE REGISTER (EMR). The INITIATOR COMMAND REGISTER (ICR) is read to determine whether arbitration has been won or lost.

The $\overline{\text{BSY}}$ signal is continuously monitored to detect bus disconnection and bus free phases. The EASI incorporates an on-board oscillator to determine Bus Settle, Bus Free and Arbitration Delays. The oscillator tolerance guarantees all timing to be within the SCSI specification.

The EASI incorporates high-current drivers and SCHMITT trigger receivers for interfacing directly to the SCSI bus. This feature reduces the chip count of any SCSI application. The driver/receivers also incorporate loopback logic which is enabled by an EMR bit. The Loopback mode enables testing of all EASI functions without interfering with the SCSI bus.

1.5 PARITY

The EASI provides for parity protection on both the μP and SCSI interfaces. Each data bus has eight data bits and one parity bit (only the PCC part provides μP parity, both the DIP and PCC provide SCSI parity). In each case the parity may

1.0 Functional Description (Continued)

be enabled via a register bit. A parity error can be programmed to cause an interrupt. Additionally the parity may be programmed to be either ODD or EVEN. This has a particular use on the SCSI interface where programming EVEN parity allows diagnostics, or determining whether a device supports parity. The inclusion of μP parity allows development of controllers that maintain data integrity right from the media to the host system.

1.6 INTERRUPTS

The EASI is intended to be used in an interrupt-driven environment. Each function can be programmed to cause an interrupt. In ENHANCED MODE two registers are used to control interrupts—the INT STATUS REGISTER (ISR) and the INT MASK REGISTER (IMR). Each interrupt can be masked from interrupting via the IMR. When an interrupt is recognized by the μP, reading the ISR will display all active interrupt sources. The ISR contents remain unchanged until an interrupt reset is programmed. A shadow register behind the ISR guarantees that interrupts occurring while others are serviced will not be lost.

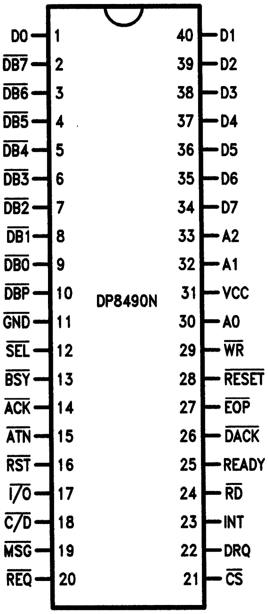
2.0 Pin Descriptions

Symbol	DIP	PCC	Type	Function
$\overline{\text{CS}}$	21	24	I	**Chip Select:** an active low enable for read or write operations, accessing the register selected by A0–A2.
A0–A2	30, 32 33	33, 36 37	I	**Address 0–2:** these three signals are used with $\overline{\text{CS}}$, $\overline{\text{RD}}$, and $\overline{\text{WR}}$ to address a register for read or write.
$\overline{\text{RD}}$	24	27	I	**Read:** an active low enable for reading an internal register selected by A0–A2 and enabled by $\overline{\text{CS}}$. It also selects the Input Data Register when used with $\overline{\text{DACK}}$.
$\overline{\text{WR}}$	29	32	I	**Write:** an active low enable for writing an internal register selected by A0–A2 and enabled by $\overline{\text{CS}}$. It also selects the Output Data Register when used with $\overline{\text{DACK}}$.
$\overline{\text{RESET}}$	28	31	I	**Reset:** an active low input with a Schmitt trigger. Clears all internal registers. (SCSI $\overline{\text{RST}}$ unaffected.)
D0–D7, P	1, 40–34	2, 44–38 1	I/O	**Data 0–7, P:** bidirectional TRI-STATE® signals connecting the active high μP data bus to the internal registers. The PCC part offers an optional parity on the μP data bus. If the parity option is not enabled the pin is TRI-STATE.
INT	23	26	O	**Interrupt:** an active high output to the μP when an error has occurred, an event requires service or has completed.
DRQ	22	25	O	**DMA Request:** an active high output asserted when the data register is ready to be read or written. DRQ occurs only if DMA mode is enabled. The signal is cleared by $\overline{\text{DACK}}$.
$\overline{\text{DACK}}$	26	29	I	**DMA Acknowledge:** an active low input that resets DRQ and addresses the data registers for input or output transfers. $\overline{\text{DACK}}$ is used instead of $\overline{\text{CS}}$ by the DMA controller.
READY	25	28	O	**Ready:** an active high output used to control the speed of block mode DMA transfers. Ready goes active when the chip is ready to send/receive data and remains inactive after the transfer until the byte is sent or until the DMA mode bit is reset.
$\overline{\text{EOP}}$	27	30	I	**End of Process:** an active low signal that terminates a block of DMA transfers. It should be asserted during the transfer of the last byte.
$\overline{\text{DB0}}$–$\overline{\text{DB7}}$ $\overline{\text{DBP}}$	9–2, 10	10–3, 11	I/O	**DB0–DB7, DBP:** SCSI data bus with parity. $\overline{\text{DB7}}$ is the MSB and is the highest priority during arbitration. Parity is default ODD but can be programmed EVEN. Parity is always generated and can be optionally checked. Parity is not valid during arbitration.
$\overline{\text{RST}}$	16	18	I/O	**Reset:** SCSI reset, monitored and can be set by EASI.
$\overline{\text{BSY}}$	13	15	I/O	**Busy:** indicates the SCSI bus is being used. Can be driven by TARGET or INITIATOR.

2.0 Pin Descriptions (Continued)

Symbol	DIP	PCC	Type	Function
$\overline{\text{SEL}}$	12	14	I/O	**Select:** used by the INITIATOR to select a TARGET or by the TARGET to reselect an INITIATOR.
$\overline{\text{ACK}}$	14	16	I/O	**Acknowledge:** driven by the INITIATOR and received by the TARGET as part of the $\overline{\text{REQ}}/\overline{\text{ACK}}$ handshake.
$\overline{\text{ATN}}$	15	17	I/O	**Attention:** driven by the INITIATOR to indicate an attention condition to the TARGET.
$\overline{\text{REQ}}$	20	22	I/O	**Request:** driven by the TARGET and received by the INITIATOR as part of the $\overline{\text{REQ}}/\overline{\text{ACK}}$ handshake.
$\overline{\text{I/O}}$	17	19	I/O	**Input/Output:** driven by the TARGET to control the direction of transfers on the SCSI bus. This signal also distinguishes between selection and reselection.
$\overline{\text{C/D}}$	18	20	I/O	**Command/Data:** driven by the TARGET to indicate whether command or data bytes are being transferred.
$\overline{\text{MSG}}$	19	21	I/O	**Message:** driven by the TARGET during message phase to identify message bytes on the bus.
V_{CC} GND	31 11	35 12, 13	—	**V_{CC}, GND:** +5 V_{DC} is required. Because of very large switching currents, good decoupling and power distribution is mandatory.

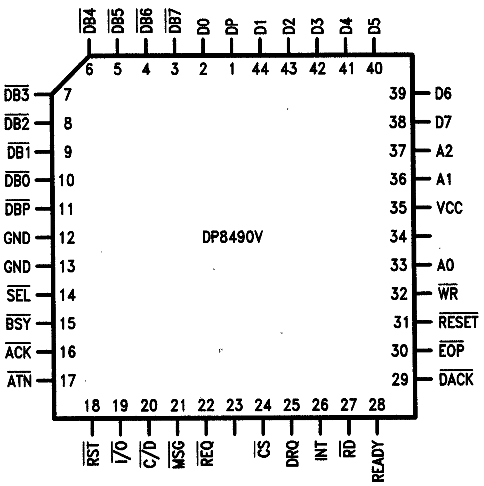
2.1 Connection Diagrams



D0 1
$\overline{\text{DB7}}$ 2
$\overline{\text{DB6}}$ 3
$\overline{\text{DB5}}$ 4
$\overline{\text{DB4}}$ 5
$\overline{\text{DB3}}$ 6
$\overline{\text{DB2}}$ 7
$\overline{\text{DB1}}$ 8
$\overline{\text{DB0}}$ 9
$\overline{\text{DBP}}$ 10
$\overline{\text{GND}}$ 11
$\overline{\text{SEL}}$ 12
$\overline{\text{BSY}}$ 13
$\overline{\text{ACK}}$ 14
$\overline{\text{ATN}}$ 15
$\overline{\text{RST}}$ 16
$\overline{\text{I/O}}$ 17
$\overline{\text{C/D}}$ 18
$\overline{\text{MSG}}$ 19
$\overline{\text{REQ}}$ 20
21 $\overline{\text{CS}}$
22 DRQ
23 INT
24 $\overline{\text{RD}}$
25 READY
26 $\overline{\text{DACK}}$
27 $\overline{\text{EOP}}$
28 $\overline{\text{RESET}}$
29 $\overline{\text{WR}}$
30 A0
31 VCC
32 A1
33 A2
34 D7
35 D6
36 D5
37 D4
38 D3
39 D2
40 D1
DP8490N

TL/F/9387-4

Order Number DP8490N
See NS Package Number N40A



$\overline{\text{DB4}}$ 6
$\overline{\text{DB5}}$ 5
$\overline{\text{DB6}}$ 4
$\overline{\text{DB7}}$ 3
D0 2
DP 1
D1 44
D2 43
D3 42
D4 41
D5 40
$\overline{\text{DB3}}$ 7
$\overline{\text{DB2}}$ 8
$\overline{\text{DB1}}$ 9
$\overline{\text{DB0}}$ 10
$\overline{\text{DBP}}$ 11
GND 12
GND 13
$\overline{\text{SEL}}$ 14
$\overline{\text{BSY}}$ 15
$\overline{\text{ACK}}$ 16
$\overline{\text{ATN}}$ 17
18 $\overline{\text{RST}}$
19 $\overline{\text{I/O}}$
20 $\overline{\text{C/D}}$
21 $\overline{\text{MSG}}$
22 $\overline{\text{REQ}}$
23
24 $\overline{\text{CS}}$
25 DRQ
26 INT
27 $\overline{\text{RD}}$
28 READY
29 $\overline{\text{DACK}}$
30 $\overline{\text{EOP}}$
31 $\overline{\text{RESET}}$
32 $\overline{\text{WR}}$
33 A0
34
35 VCC
36 A1
37 A2
38 D7
39 D6
DP8490V

TL/F/9387-5

Order Number DP8490V
See NS Package Number V44A

3.0 Register Description

3.1 GENERAL

The DP8490 EASI is a register-based device with eight addressable locations used to access twelve registers. Some addresses have dual functions depending upon whether they are being read from or written to. Two basic operating "modes" result in differences in the registers accessed through address 7. Device operation is described in Section 4 but mode differences are highlighted in this section and Section 8.

The EASI operates in one of two modes—NORMAL (MODE N) and ENHANCED (MODE E). Switching between the modes is performed by setting or resetting bit 6 in the Initiator Command Register.

In MODE N, EASI registers appear the same as the DP5380. In MODE E, address 7 accesses enhanced logic features. To help identify these differences the register description is split into two subsections. The first describes the registers in MODE N. The second describes the register differences when in MODE E.

Figure 3.1 summarizes the register map in MODE N. Note that for registers reading or writing SCSI signals the SCSI name is used for each bit. Although the SCSI bus is active low, the registers invert the SCSI bus. This means an active SCSI signal is represented by a ONE in a register and an inactive signal by a ZERO.

Hex Adr	Register	Mnemonic	Bits	R/W
0	Output Data Register	ODR	8	WO
0	Current SCSI Data	CSD	8	RO
1	Initiator Command Register	ICR	8	RW
2	Mode Register 2	MR2	8	RW
3	Target Command Register	TCR	4	RW
4	Select Enable Register	SER	8	WO
4	Current SCSI Bus Status	CSB	8	RO
5	Bus and Status	BSR	8	RO
5	Start DMA Send	SDS	0	WO
6	Start DMA Target Receive	SDT	0	WO
6	Input Data Register	IDR	8	RO
7	Start DMA Initiator Receive	SDI	0	WO
7	Reset Parity/Interrupts	RPI	0	RO

FIGURE 3.1. Normal Mode Registers

3.2 NORMAL MODE REGISTERS

OUTPUT DATA REGISTER (ODR)

8 Bits HA 0 Write Only

This is a transparent latch used to send data to the SCSI bus. The register can be written by μP cycles or via DMA. DMA writes automatically select the ODR at Hex Address 0 (HA 0). This register is also written with the ID bits required during arbitration and selection/reselection phases. Data is latched at the end of the write cycle.

Bit 7 **Bit 0**

$\overline{\text{DB7}}$	$\overline{\text{DB6}}$	$\overline{\text{DB5}}$	$\overline{\text{DB4}}$	$\overline{\text{DB3}}$	$\overline{\text{DB2}}$	$\overline{\text{DB1}}$	$\overline{\text{DB0}}$

Output Data Register

CURRENT SCSI DATA (CSD)

8 Bits HA 0 Read Only

This register enables reading of the current SCSI data bus. If SCSI parity checking is enabled it will be checked at the beginning of the read cycle. The register is also used for μP accesses of SCSI data during programmed-I/O or ID checking during arbitration. Parity is not valid during arbitration. DMA transfers select the IDR (HA 6) instead of the CSD register.

Bit 7 **Bit 0**

$\overline{\text{DB7}}$	$\overline{\text{DB6}}$	$\overline{\text{DB5}}$	$\overline{\text{DB4}}$	$\overline{\text{DB3}}$	$\overline{\text{DB2}}$	$\overline{\text{DB1}}$	$\overline{\text{DB0}}$

Current SCSI Data

INITIATOR COMMAND REGISTER (ICR)

8 Bits HA 1 Read/Write

This register is used to control the INITIATOR and some other SCSI signals, and to monitor the progress of bus arbitration. Most of the SCSI signals may also be asserted in TARGET mode. Bits 5 to 0 are reset when $\overline{\text{BSY}}$ is lost (see MR2 description).

Bit 7 **Bit 0**

$\overline{\text{RST}}$	AIP/ MODE	LA/ DIFF	$\overline{\text{ACK}}$	$\overline{\text{BSY}}$	$\overline{\text{SEL}}$	$\overline{\text{ATN}}$	DBUS

Initiator Command Register

DBUS: Assert Data Bus Bit 0

0 Disable SCSI data bus driving.

1 Enable contents of Output Data Register onto the SCSI data bus. SCSI parity is also generated and driven on DBP.

This bit should be set when transferring data out of the EASI in either TARGET or INITIATOR mode, for both DMA or programmed-I/O. In INITIATOR mode the drivers are only enabled if: Mode Register 2 TARGET MODE bit is 0, and I/$\overline{\text{O}}$ is false, and C/$\overline{\text{D}}$, I/$\overline{\text{O}}$, $\overline{\text{MSG}}$ match the contents of the Target Command Register (phasematch is true). In TARGET mode only the MR2 bit needs to be set with this bit.

Reading the ICR reflects the state of this bit.

$\overline{\text{ATN}}$: Assert Attention Bit 1

0 Deassert $\overline{\text{ATN}}$

1 Assert SCSI $\overline{\text{ATN}}$ signal. The MR2 TARGET MODE bit must also be false to assert the signal.

Reading the ICR reflects the state of this bit.

$\overline{\text{SEL}}$: Assert Select Bit 2

0 Deassert $\overline{\text{SEL}}$

1 Assert SCSI $\overline{\text{SEL}}$ signal. Can be used in INITIATOR or TARGET mode.

Reading the ICR reflects the state of this bit.

$\overline{\text{BSY}}$: Assert Busy Bit 3

0 Deassert $\overline{\text{BSY}}$.

1 Assert SCSI $\overline{\text{BSY}}$ signal. Can be used in INITIATOR or TARGET mode.

Reading the ICR reflects the state of this bit.

3.0 Register Description (Continued)

$\overline{\text{ACK}}$: Assert Acknowledge Bit 4

0 Deassert $\overline{\text{ACK}}$.

1 Assert SCSI $\overline{\text{ACK}}$ signal. The MR2 TARGET MODE bit must also be false to assert the signal.

Reading the ICR reflects the state of this bit.

DIFF: Differential Enable Bit 5 Write

0 This bit must be reset to 0.

LA: Lost Arbitration Bit 5 Read

0 Normally reset to 0 to show arbitration not lost or not enabled.

1 Will be set when the EASI loses arbitration, i.e. when SEL is true during arbitration AND the Assert $\overline{\text{SEL}}$ bit of this register is false.

A 1 in this bit means the EASI has arbitrated for the bus, asserted $\overline{\text{BSY}}$ and its ID on the data bus and another device has asserted $\overline{\text{SEL}}$. The ARBITRATE bit in MR2 or the EMR must be set to enable arbitration.

MODE: Operating Mode Bit 6 Write

0 Normal Mode (MODE N) is selected.

1 Enhanced Mode (MODE E) is selected.

AIP: Arbitration In Progress Bit 6 Read

0 Normally 0 to show no arbitration in progress.

1 Set when the EASI has detected BUS FREE phase and asserted $\overline{\text{BSY}}$ and the Output Data Register contents onto the SCSI data bus. This bit remains set until arbitration is disabled.

$\overline{\text{RST}}$: Assert $\overline{\text{RST}}$ Bit 7

0 Deassert $\overline{\text{RST}}$.

1 Assert SCSI $\overline{\text{RST}}$ signal. $\overline{\text{RST}}$ is asserted as long as this bit is 1, or until a μP Reset ($\overline{\text{RESET}}$).

After this bit is set the INT pin goes active and internal registers are reset (except for the interrupt latch, MR2 TARGET MODE bit, and this bit). Reading the ICR reflects the state of this bit.

MODE REGISTER 2 (MR2)

8 Bits HA 2 Read/Write

This register is used to program basic operating conditions in the EASI. Operation as TARGET or INITIATOR, DMA mode and type as well as some interrupt controls are set via this register. This is a read/write register and when read the value reflects the state of each bit.

Bit 7							**Bit 0**
BLK	TARG	PCHK	PINT	EOP	BSY	DMA	ARB

Mode Register 2

ARB: Arbitrate Bit 0

0 Disable arbitration

1 Enable arbitration. The EASI will wait for a BUS FREE phase then arbitrate for the bus. Before setting this bit the Output Data Register should contain the SCSI device ID—a single bit set only. The status of the arbitration process is given in the AIP and LA bits (6,5) in the Initiator Command Register.

DMA: DMA Mode Bit 1

0 Disable DMA mode

1 Enable DMA operation. This bit should be set then one of address 5 to 7 written to start DMA. The TARGET MODE bit in the ICR and the phase lines in the TCR should have been set appropriately. The DBUS bit in the ICR must be set for DMA send operations. $\overline{\text{BSY}}$ must be active in order to set this bit. The phase lines must match the contents of the TCR during the actual transfers. In DMA mode EASI logic automatically controls the $\overline{\text{REQ}}/\overline{\text{ACK}}$ handshakes.

This bit should be reset by a μP write to stop any DMA transfer. An $\overline{\text{EOP}}$ signal will not reset this bit. During DMA, $\overline{\text{CS}}$ and $\overline{\text{DACK}}$ should not be active simultaneously.

This bit will be reset if $\overline{\text{BSY}}$ is lost during DMA mode.

BSY: Monitor Busy Bit 2

0 Disable $\overline{\text{BSY}}$ monitor.

1 Monitor SCSI $\overline{\text{BSY}}$ signal and interrupt when $\overline{\text{BSY}}$ goes inactive. When this bit goes active the lower 6 bits of the ICR are reset and all signals removed from the SCSI bus. This is used to check for valid TARGET connection.

EOP: Enable $\overline{\text{EOP}}$ Interrupt Bit 3

0 No interrupt for $\overline{\text{EOP}}$.

1 Interrupt after valid $\overline{\text{EOP}}$ condition.

PINT: Enable SCSI Parity Interrupt Bit 4

0 No Interrupt on SCSI parity error.

1 When SCSI parity is enabled via the PCHK bit, setting this bit enables an interrupt upon a SCSI parity error.

PCHK: Enable SCSI Parity Checking Bit 5

0 No SCSI parity checking.

1 Enable checking of SCSI parity during read operations. This applies to either programmed I/O or DMA mode.

TARG: Target Mode Bit 6

0 Initiator Mode

1 Target Mode

BLK: Block Mode DMA Bit 7

0 Non-block DMA

1 When set along with DMA bit (1), enables block mode DMA transfers. In block mode the READY line is used to handshake each byte with the DMA controller instead of the DRQ/$\overline{\text{DACK}}$ handshake used in non-block mode.

TARGET COMMAND REGISTER (TCR)

4 Bits HA 3 Read/Write

This register is used to control TARGET SCSI signals and to program the desired phase during INITIATOR mode. During DMA transfers the SCSI phase lines ($\overline{\text{C/D}}$, $\overline{\text{MSG}}$, $\overline{\text{I/O}}$) must match the contents of the TCR for transfers to occur. A phase mismatch halts DMA transfers and generates an interrupt.

Bit 7							**Bit 0**
X	X	X	X	$\overline{\text{REQ}}$	$\overline{\text{MSG}}$	$\overline{\text{C/D}}$	$\overline{\text{I/O}}$

Target Command Register

3.0 Register Description (Continued)

This is a read/write register and the value read reflects the state of each bit, except bits 4–7 which always read 0.

$\overline{\text{I/O}}$: Assert $\overline{\text{I/O}}$ Bit 0

0 Deassert $\overline{\text{I/O}}$.

1 Assert SCSI $\overline{\text{I/O}}$ signal. The MR2 TARGET MODE bit must also be active.

$\overline{\text{C/D}}$: Assert $\overline{\text{C/D}}$ Bit 1

0 Deassert $\overline{\text{C/D}}$

1 Assert SCSI $\overline{\text{C/D}}$ signal. The MR2 TARGET MODE bit must also be active.

$\overline{\text{MSG}}$: Assert $\overline{\text{MSG}}$ Bit 2

0 Deassert $\overline{\text{MSG}}$.

1 Assert SCSI $\overline{\text{MSG}}$ signal. The MR2 TARGET MODE bit must also be active.

$\overline{\text{REQ}}$: Assert $\overline{\text{REQ}}$ Bit 3

0 Deassert $\overline{\text{REQ}}$.

1 Assert SCSI $\overline{\text{REQ}}$ signal. The MR2 TARGET MODE bit must also be active. This bit is used to handshake SCSI data via programmed-I/O.

SELECT ENABLE REGISTER (SER)

8 Bits HA 4 Write Only

This write-only register is used to program the SCSI device ID for the EASI to respond to during Selection or Reselection phases. Only one bit in the register should be set. When $\overline{\text{SEL}}$ is true, $\overline{\text{BSY}}$ false and the SER ID bit active an interrupt will occur.

This interrupt is reset or can be disabled by writing zero to this register. Parity will also be checked during Selection or Reselection if the PCHK bit in MR2 is set.

Bit 7							Bit 0
$\overline{\text{DB7}}$	$\overline{\text{DB6}}$	$\overline{\text{DB5}}$	$\overline{\text{DB4}}$	$\overline{\text{DB3}}$	$\overline{\text{DB2}}$	$\overline{\text{DB1}}$	$\overline{\text{DB0}}$

Select Enable Register

CURRENT SCSI BUS STATUS (CSB)

8 Bits HA 4 Read Only

This read-only register is used to monitor SCSI control signals and the SCSI parity bit. The SCSI lines are monitored during programmed-I/O transfers and after an interrupt in order to determine the cause. A bit is 1 if the corresponding SCSI signal is active.

Bit 7							Bit 0
$\overline{\text{RST}}$	$\overline{\text{BSY}}$	$\overline{\text{REQ}}$	$\overline{\text{MSG}}$	$\overline{\text{C/D}}$	$\overline{\text{I/O}}$	$\overline{\text{SEL}}$	$\overline{\text{DBP}}$

Current SCSI Bus Status

BUS AND STATUS REGISTER (BSR)

8 Bits HA 5 Read Only

This read-only register is used to monitor SCSI signals not included in the CSB, and internal status bits. This register is read after an interrupt (in MODE N) to determine the cause of an interrupt. Bit 0 or 1 are set to 1 if the SCSI signal is active.

Bit 7							Bit 0
EDMA	DRQ	SPER	INT	PHSM	BSY	$\overline{\text{ATN}}$	$\overline{\text{ACK}}$

Bus and Status Register

$\overline{\text{ACK}}$: Acknowledge Bit 0

This bit reflects the state of the SCSI $\overline{\text{ACK}}$ Signal.

$\overline{\text{ATN}}$: Attention Bit 1

This bit reflects the state of the SCSI $\overline{\text{ATN}}$ Signal.

BSY: Busy Error Bit 2

0 No error

1 The SCSI $\overline{\text{BSY}}$ signal has become inactive while the MR2 $\overline{\text{BSY}}$ (Monitor $\overline{\text{BSY}}$) bit is set. This will cause an interrupt, remove all EASI signals from the SCSI bus and reset the DMA MODE bit in MR2.

PHSM: Phase Match Bit 3

0 Phase Match. The SCSI $\overline{\text{C/D}}$, $\overline{\text{I/O}}$ and $\overline{\text{MSG}}$ phase lines are continuously compared with the corresponding bits in the TCR. The result of this comparison is reflected in this bit. This bit must be 1 (phase matches) for DMA transfers. A phase mismatch will stop DMA transfers and cause an interrupt.

INT: Interrupt Request Bit 4

0 No interrupt

1 Interrupt request active. Set when an enabled interrupt condition occurs. This bit reflects the state of the INT pin. INT may be reset by performing a Reset Parity/Interrupt (RPI) function.

SPER: SCSI Parity Error Bit 5

0 No SCSI parity error

1 SCSI parity error occurred. This bit remains set once an error occurs until the RPI function clears it. The PCHK bit in MR2 must be set for a parity error to be checked and registered.

DRQ: DMA Request Bit 6

0 No DMA request

1 DMA request active. This bit reflects the state of the DRQ pin. DRQ is reset by asserting $\overline{\text{DACK}}$ during a DMA cycle or by resetting the DMA bit in MR2. A Busy error will reset the MR2 DMA bit and thus will also clear DRQ. A phase mismatch will not reset DRQ.

EDMA: End of DMA Bit 7

0 Not end of DMA

1 Set when $\overline{\text{DACK}}$, $\overline{\text{EOP}}$ and either $\overline{\text{RD}}$ or $\overline{\text{WR}}$ are active simultaneously. Normally occurs when the last byte is transferred by the DMA. During DMA send operations the last byte transferred by the DMA may not have been transferred on SCSI so $\overline{\text{REQ}}$ and $\overline{\text{ACK}}$ should be monitored to verify when the last SCSI transfer is complete. This bit is reset when the MR2 DMA bit is reset.

Note: In MODE E the EASI presents a true EDMA bit in bit 7 of the TCR. This feature removes the need to poll the $\overline{\text{REQ}}$ and $\overline{\text{ACK}}$ signals.

START DMA SEND (SDS)

0 Bits HA 5 Write Only

This write-only register is used to start a DMA send operation. A write of don't care data should be the last thing done by the μP. The MR2 DMA, BLK and TARG bits must have been programmed previously.

Bit 7							Bit 0
X	X	X	X	X	X	X	X

Start DMA Send

3.0 Register Description (Continued)

START DMA TARGET RECEIVE

0 Bits HA 6 Write Only

This write-only register is used to start a DMA Target Receive operation. Same comments as SDS apply.

INPUT DATA REGISTER (IDR)

8 Bits HA 6 Read Only

This read-only register contains the SCSI data last latched during a DMA receive. Each byte from SCSI is latched into this register automatically by the EASI DMA logic. A DMA read ($\overline{\text{DACK}}$ and $\overline{\text{RD}}$) automatically selects this register. Programmed I/O SCSI data reads should use the CSD (HA 8).

START DMA INITIATOR RECEIVE (SDI)

0 Bits HA 7 Write Only

This write-only register is used to start DMA INITIATOR Receive Operation. Same comments as SDS apply. An alternative method of performing the SDI function is available in the Enhanced Mode Register.

RESET PARITY/INTERRUPT (RPI)

0 Bits HA 7 Read Only

This read-only register is used to reset the parity and interrupt latches. Reading this register resets the SCSI parity, μP parity, Busy Loss and Interrupt Request latches. It also resets the interrupt latches presented in the Interrupt Status Register (available in MODE E).

An alternative method of performing the Reset Parity Interrupt function is available in the Enhanced Mode Register. In MODE E writing a value of 01 to bits 2 and 1 of the EMR will reset the same bits as a read from HA 7 in MODE N. The EMR RPI will also reset enhanced logic that has bits set in the Interrupt Status Register.

3.3 ENHANCED MODE REGISTERS

Addresses 0 to 6 remain the same as MODE N except for bit 7 of TCR, as described below. Address 7 is the SDI and RPI functions in MODE N, but in MODE E it directly accesses the Enhanced Mode Register (EMR) and indirectly accesses the Interrupt Mask Register (IMR) and the Interrupt Status Register (ISR).

When bit 6 in the ICR (HA 1) is set, HA 7 accesses the read/write EMR. The SDI and RPI functions performed when writing/reading HA 7 in MODE N are disabled. To perform these functions the EMR is used instead.

Note that EMR functions are intended to be used in an interrupt-driven environment. Reading this register reflects the state of each bit.

TARGET COMMAND REGISTER (TCR)

5 Bits HA 3 Read/Write

This is the same as MODE N except for bit 7 which is described below. Note bits 4–6 always read 0.

Bit 7 **Bit 0**

(true) EDMA	X	X	X	$\overline{\text{REQ}}$	$\overline{\text{MSG}}$	$\overline{\text{C/D}}$	$\overline{\text{I/O}}$

Target Command Register

EDMA: True End of DMA Bit 7

0 Not End of DMA

1 Set when the last byte of data has been transferred. This bit is not set until $\overline{\text{REQ}}$ and $\overline{\text{ACK}}$ both go inactive following the DMA cycle during which $\overline{\text{EOP}}$ was asserted. Note that unlike the BSR EDMA bit, this bit reflects the true completion of DMA transfers.

ENHANCED MODE REGISTER (EMR)

8 Bits HA 7 Read/Write

This register is accessed at HA 7 when the ICR MODE bit (6) is set. The register controls operation of enhanced logic and timing. Normally the application will leave the EASI permanently in MODE N or MODE E.

Bit 7 **Bit 0**

APHS	MPEN	MPOL	SPOL	LOOP	EFN1	EFN0	ARB

Enhanced Mode Register

ARB: Extended Arbitration Bit 0

0 Disable extended arbitration

1 Enable extended arbitration. This is an alternative bit to the MR2 ARB function. The EASI waits for a BUS FREE phase then arbitrates for the bus, asserting the contents of the ODR onto the SCSI data bus and asserting $\overline{\text{BSY}}$. The EASI will then wait the 2.2 μs SCSI Arbitration Delay, and set the Arbitration Complete bit in the ISR and cause an interrupt. As for the MR2 ARB function, the ICR LA and AIP bit can be examined to determine arbitration status.

EFN1,0: Enhanced Function Bits 2,1

00 No operation—these bits ALWAYS read 00.

01 When this pattern is written the Parity and Interrupt latches are reset. This pattern should be followed by any other pattern to remove the reset (usually 00). This function replaces the RPI function performed when reading HA 7 in MODE N. Only the latches with ISR bits set to 1 will be reset.

10 Start DMA Initiator Receive. At the end of the write cycle the SDI function is performed. There is no need to follow with another pattern as required with the RPI value (01). This function replaces the SDI function performed when writing HA 7 in MODE N.

11 Read/Write ISR/IMR. When this pattern is written the NEXT read of HA 7 will access the ISR, or the NEXT write to HA 7 will access the IMR. This state ONLY lasts for the ONE following read OR write cycle. Further cycles will then access the EMR.

LOOP: Loopback Mode Bit 3

0 Normal operation

1 When set: SCSI drivers are disabled and the SCSI I/O's looped back inside the EASI, and both the TARGET and INITIATOR signals may be driven simultaneously. This enables the μP to check EASI operation without affecting the SCSI bus.

SPOL: SCSI Parity Polarity Bit 4

0 SCSI parity is ODD (as per SCSI specification).

1 SCSI parity is EVEN. This allows diagnostics to be performed.

3.0 Register Description (Continued)

MPOL: μP Parity Polarity Bit 5

0 μP parity is ODD.

1 μP parity is EVEN.

MPEN: μP Parity Enable Bit 6

0 μP parity checking and generation disabled.

1 Enable checking of μP data bus parity during μP and DMA writes. Generate parity during μP and DMA reads. Parity errors will cause an interrupt and set the ISR MPE bit if not masked.

APHS: Any Phase Mismatch Bit 7

0 Disable phase mismatch detection

1 Detect SCSI requests with a phase mismatch present. Is set when $\overline{\text{REQ}}$ goes active AND the SCSI phase lines do not match the contents of the TCR. Can be used in INITIATOR mode to interrupt on TARGET phase changes.

INTERRUPT STATUS REGISTER (ISR)

8 Bits HA 7 (EFN = 11) Read Only

This register is accessed during the first read cycle after the EFN bits in the EMR have both been set to 1. Once read, successive accesses of HA 7 go to the EMR. This register provides all interrupt status within one register. This is intended to make determination of interrupt sources easier in MODE E. In MODE N two registers must be read—the BSR and CSB. In MODE E only the ISR needs to be read. Additionally each interrupt (except SCSI $\overline{\text{RST}}$) has a corresponding status bit in the ISR.

When the ISR is read all unmasked, enabled, active interrupt sources set their corresponding ISR bits to a ONE. The interrupt status is sampled at the beginning of the ISR read cycle. When an RPI function is performed via the EMR, only each interrupt latch with an ISR bit set will be reset. This means interrupts occurring since the last ISR read will not be lost.

ISR bits may be individually masked via their corresponding bits in the IMR.

Bit 7 **Bit 0**

SPE	MPE	EDMA	DPHS	APHS	BSY	SEL	ARB

Interrupt Status Register

ARB: Arbitration Complete Bit 0

This bit is set when arbitration enabled by EMR ARB bit (0) has completed. Completion occurs in two ways: when the EASI loses arbitration and has asserted the LA bit in the ICR; or when the EASI has asserted the ID contained in the ODR onto the data bus, asserted $\overline{\text{BSY}}$, then waited for the 2.2 μs SCSI Arbitration Delay.

SEL: Selection/Reselection Bit 0

This bit is set when $\overline{\text{BSY}}$ is false, $\overline{\text{SEL}}$ is active, and any SER bit set to 1 has an active corresponding SCSI data bus bits. This situation occurs during Selection or Reselection phases.

BSY: Busy Loss Bit 2

This bit is the same as BSR bit 2. Set when the SCSI $\overline{\text{BSY}}$ signal becomes inactive while the MR2 BSY (monitor $\overline{\text{BSY}}$) bit is set.

APHS: Any Phase Mismatch Bit 3

Set when a $\overline{\text{REQ}}$ occurs while the SCSI phase lines do not match. Bit is set to enable detection of the contents of the TCR. EMR APHS (bit 7) must be 1 to allow this bit to be set.

DPHS: DMA Phase Mismatch Bit 4

Set when a SCSI DMA mode operation occurs with a phase mismatch. Similar to the APHS condition but restricted to DMA mode only. In MODE N this condition is not as easily determined.

EDMA: End of DMA Bit 5

Set when $\overline{\text{REQ}}$ and $\overline{\text{ACK}}$ are both false following the DMA cycle during which $\overline{\text{EOP}}$ was asserted. This represents the true end of DMA operation.

MPE: μP Parity Error Bit 6

Set when μP parity error is detected at the end of a μP or DMA write cycle. The MPEN bit in the EMR must be set for μP parity to be checked.

SPE: SCSI Parity Error Bit 7

Set when a SCSI parity error is detected when reading the CSD, during selection/reselection, or when the IDR is loaded during DMA operation. The MR2 PCHK bit must be set for SCSI parity to be checked, and the MR2 PINT bit must be set to enable the interrupt.

INTERRUPT MASK REGISTER (IMR)

8 Bits HA 7 (EFN = 11) Write Only

This register is accessed during the first write cycle after the EFN bits in the EMR have been set to 11. Once written, successive reads or writes at HA 7 access the EMR.

This register has the same bit definition as the ISR. If a bit in the IMR is set to 1 that interrupt will be masked. The interrupt will be captured internally (if enabled) but will not cause an active INT signal. A bit reset to 0 will enable that interrupt to occur (if enabled) and will enable the ISR bit to be set to 1.

Bit 7 **Bit 0**

SPE	MPE	EDMA	DPHS	APHS	BSY	SEL	ARB

Interrupt Mask Register

4.0 Device Operation

4.1 GENERAL

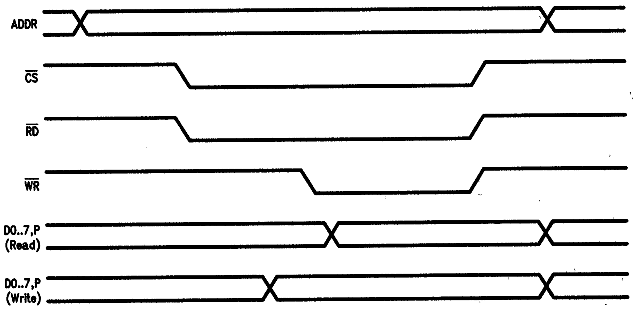
This section describes overall operation of the EASI. More detailed information of data transfers, interrupts and reset conditions are covered in later sections. The operation description covers μP accesses, SCSI bus monitoring, arbitration, selection, reselection, programmed-I/O, DMA interrupts. Programming and timing details are covered.

For information regarding interfacing to μPs and DMA controllers refer to Section 9.

In the descriptions following, program examples are given in pseudo-C. This processor-independent approach should be clearest. These are backed up by flow charts in Appendix A1.

For each section where appropriate the description is split into two, MODE N and MODE E.

4.0 Device Operation (Continued)



TL/F/9387-6

FIGURE 4.2. μP Cycles

4.2 μP ACCESSES

The μP accesses the EASI via the $\overline{\text{CS}}$, $\overline{\text{RD}}$, $\overline{\text{WR}}$ and address and data lines in order to read/write the registers. *Figure 4.1* shows typical timing. Note the use of non-multiplexed address and data lines.

4.3 SCSI BUS MONITORING/DRIVING

The SCSI bus may be monitored or driven at any time. Each bus signal is buffered and inverted by the EASI and can be read via the CSB, BSR and CSD registers. An active SCSI signal reads a 1 in the status registers.

Each SCSI signal may be asserted by setting a bit in the TCR or ICR. Setting the bit to 1 asserts the SCSI signal.

The following code demonstrates a byte transferred via programmed-I/O in INITIATOR mode.

```
{    /* Transfer one byte as Initiator */
    while (NOT (TCR: REQ));
     /* wait till TARGET asserts REQ */
    data = input (CSD);
     /* parity is checked if enabled */
    output (ICR, Assert_ACK);
    while (TCR:REQ);
     /* wait till TARGET deasserts REQ */
    output (ICR, 0);
     /* deassert ACK, ready for next byte */
}
```

4.4 ARBITRATION

This sub-section describes the arbitration support provided by the EASI and how to program it.

4.4.1 MODE N Arbitration

Since the SCSI arbitration process requires signal sequencing too fast for μPs, hardware support is provided by the EASI. The arbitration process is enabled by bit 0 MR2 (ARB). Prior to setting this bit the ODR should be programmed with the device's SCSI ID—a single bit.

The EASI will monitor the bus for a BUS FREE phase. The $\overline{\text{BSY}}$ signal is continuously monitored. If continuously inactive for at least a SCSI Bus Settle Delay (400 ns) and $\overline{\text{SEL}}$ is inactive, a valid Bus Free Phase exists. After a period of SCSI Bus Free Delay (800 ns) the EASI asserts $\overline{\text{BSY}}$ and the ODR onto the SCSI data bus. The μP should poll the ICR to determine when arbitration has started. The AIP bit in the ICR is set when the Bus Free Phase is detected and the EASI is beginning the Bus Free Delay. Following the Bus Free Delay a 2.2 μs SCSI Arbitration Delay is required before examining the data bus to resolve the priorities of the ID bits. This delay must be implemented in firmware. The ICR Lost Arbitration (LA) bit must be examined to determine whether arbitration is lost. The LA bit is set if another device asserts $\overline{\text{SEL}}$ during arbitration. If the LA bit is 0 the data bus is read via the CSD register. The data is examined to resolve ID priorities. If this device is the highest ID, assert $\overline{\text{SEL}}$ by setting ICR bit 2 to a 1. After waiting Bus Clear + Bus Settle Delays (1200 ns), the Selection Phase begins. These 2 delays must be implemented in firmware.

4.4.2 MODE E Arbitration

The extended arbitration in MODE E is enabled by bit 0 of the EMR (ARB). This alternative offers two significant advantages over MODE N. First the 2.2 μs SCSI Arbitration Delay is implemented by the EASI. Second the arbitration process may be interrupt driven.

In MODE N the EASI must be polled to see when the ICR AIP bit is set. After the appropriate delays, the LA bit and data bus are examined. If arbitration is lost or this device is not the highest priority the MR2 ARB bit must be reset to 0, then set to 1 again and the whole process restarted. This means the EASI MUST be polled until arbitration is won—potentially many SECONDS, typically ms. This ties up the host μP.

In MODE E when the EMR ARB bit is set the EASI will: wait for BUS FREE phase; delay SCSI Bus Free Delay; assert $\overline{\text{BSY}}$ and the ODR onto $\overline{\text{DB0}}$–$\overline{\text{DB7}}$; delay SCSI Arbitration Delay; interrupt the μP.

The μP should read the ISR and if the ISR bit 0 (ARB complete) is set examine ICR bits 5 and 6 (LA and AIP) to determine whether arbitration is lost. If not lost the data bus is examined to resolve ID priorities. As for MODE N if arbitration has failed the EMR ARB bit should be reset and then set again after first resetting the ARB interrupt. Note that the EMR ARB bit allows the μP to carry on with other tasks while the EASI arbitrates. This means there is NO NEED to poll the EASI.

4.5 SELECTION/RESELECTION

The EASI can be used to select or reselect a device. The EASI will also respond to selection or reselection. Selecting or reselecting a device is the same in MODE N or MODE E. Response to selection or reselection can differ between the bus modes.

4.0 Device Operation (Continued)

4.5.1 Selecting/Reselecting

Selection requires programming the ODR with the desired and own device IDs; the data bus via ICR DBUS (bit 0); asserting ATN if required via ICR bit 1; asserting SEL via ICR bit 2; then resetting the MR2/EMR ARB bit.

The SER should have been cleared to zero before Selection/Reselection to ensure the EASI does not respond. If Reselection is desired the I/O line should also be asserted before SEL via TCR bit 0.

Resetting the ARB bit causes the EASI to remove BSY and the ODR from the data bus. Thus the ICR Assert data bus bit is required to assert the bits for desired and own device IDs.

BSY is then monitored to determine when the device has responded to (re)selection. If the device fails to respond an error handler should sequence the EASI off the bus. If the device responds the ICR DBUS and SEL bits should be reset to remove these signals. If this is a Reselection the ICR BSY bit (3) should be set before removing the other signals.

The bus is now ready to handle Information Transfer Phases.

4.5.2 MODE N (Re)Selection Response

The EASI responds to Selection or Reselection when the SER is non-zero. A (re)selected interrupt is generated when BSY is false for at least a Bus Settle Delay (400 ns); and SEL is true AND any non-zero bit in the SER has its corresponding SCSI data bus bit active. A Selection is disabled by zeroing the SER. If parity is supported it should be valid during (re)selection so it must be checked via the SPE bit (5) in the BSR. SCSI specification states that (re)selection is not valid if more than 2 data bits are active. This condition is checked by reading the CSD.

When the selection interrupt occurs it is determined by reading the BSR and CSB registers. There is no dedicated status bit for (re)selection in MODE N, it must be determined by the absence of other interrupts, and the active state of the SEL signal. Reselection occurs when I/O is also active. See Section 6.

4.5.3 MODE E (Re)Selection Response

The same conditions for valid (re)selection apply as for MODE N. A (re)selection interrupt is generated as per MODE N. The difference is that the interrupt sets the SEL bit (1) in the ISR. In MODE E, reading the ISR enables exact determination of interrupt sources. This interrupt can be masked via bit 1 of the IMR. The interrupt is reset by programming the RPI function to the EMR. See Section 6 for further details.

4.6 MONITORING BSY

While an INITIATOR is connected to a TARGET the TARGET must maintain an active BSY signal. During DMA operations the BSY signal is monitored by the EASI and will halt operations if it goes inactive. To enable BSY to be monitored at other times the MR2 BSY bit (2) should be set. An interrupt will be generated if BSY goes inactive while MR2 BSY is set.

In MODE N this interrupt sets bit 2 in the BSR. In MODE E this interrupt sets bit 2 in the BSR and bit 2 in the ISR. The interrupt may be masked via bit 2 of the IMR.

4.7 COMMAND/MESSAGE/STATUS TRANSFERS

Command, message and status bytes are transferred using programmed-I/O. The SCSI REQ/ACK handshake is accomplished by monitoring and setting lines individually. Data is output via the ODR and read in via the CSD register.

The following code shows INITIATOR and TARGET programming for two of these cases. See Appendix A1 for flowcharts.

Initiator Command Send

```
{
  MR2 = monitor BSY
  TCR = Command Phase   /* 02h */
  while (bytes) to do) {
        while (REQ) inactive)
              idle; /* CSB bit 5 = 0 */
        if (BSR: phase_match==0)
              phase error;
        else  {
              ODR = data byte;
              ICR = Assert_ACK;
              while (REQ active)
              idle; /* CSB bit 5==1 */
              ICR = deassert_ACK
              /* byte transfer complete */
              byte count - -;
        }
  }
  goto data phase;
}
```

Target Message Receive

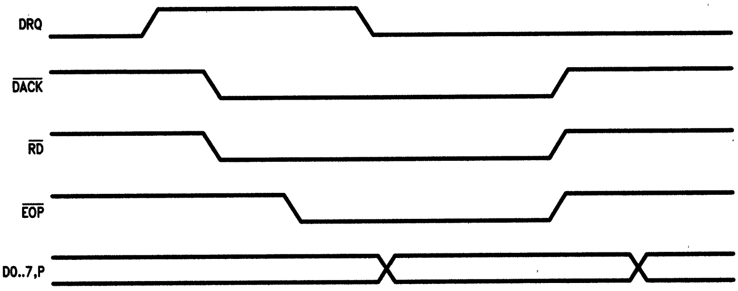
```
{
  /* assumed Assert_BSY already set in ICR */
  MR2 = TARG MODE OR PARITY CHECK
        OR PARITY INTERRUPT;
  TCR = Message_Out phase; /* 06h */
  delay (Bus Settle);
  TCR = Assert_REQ;
  while (ACK inactive)
        idle; /* BSR bit 0 */
  data = CSD; /* parity is latched */
  if (BSR: parity_error)
        error routine;
  else  {
        TCR = deassert_REQ;
        while (ACK active)
              idle;
  }
    /* message done, can change to next
       phase*/
}
```

4.8 NON-BLOCK DMA TRANSFERS

Data transfers may be effected by DMA. This method should be used for optimum performance. Two methods of DMA are available—block and non-block mode. This section describes non-block mode transfers. MODE N operation is covered first followed by MODE E.

The interface to the DMA controller uses the DRQ, DACK, EOP lines in non-block mode. Each byte is requested (DRQ) and ack'd (DACK). Representative timing for a DMA read is shown in *Figure 4.8.1*.

4.0 Device Operation (Continued)



TL/F/9387-7

FIGURE 4.8.1. Non-Block DMA Timing

4.8.1 MODE N Non-Block DMA

DMA operation involves programming the EASI with the set-up parameters, initiating the DMA cycles and checking for correct operation when the completion interrupt is received. The DMA controller should be programmed with the data byte count and the memory start address. Methods of halting a DMA operation are covered in Section 4.11.

Setting up the EASI requires enabling or disabling the following: Data bus driving, DMA mode enable, $\overline{\text{BSY}}$ monitoring, $\overline{\text{EOP}}$ interrupt, parity checking, parity interrupt, TARGET Mode, bus phase.

Once set up, DMA should be initiated by writing to address 5, 6 or 7 as appropriate. The DMA controller should assert $\overline{\text{EOP}}$ during the transfer of the last byte, although this may be done by the μP if the DMA transfers (n−1) bytes and the μP transfers the last byte. See the application guide for more details (Section 9).

Upon completion the μP should check the following as required: End of DMA, Parity Error, Phase Match, Busy Error. In MODE N the end of DMA occurs as a response to $\overline{\text{EOP}}$. SCSI transfers may still be underway so $\overline{\text{REQ}}$ and $\overline{\text{ACK}}$ must still be checked to establish when the final byte is finished.

The code below shows programming of the EASI in each of the four DMA cases. One of these cases is shown in a flow diagram in Appendix A.

```
Initiator Send    /* DATA OUT PHASE */
{
  Program DMA Controller;
  TCR = 00h;     /* phase */
  ICR = 01h;     /* Assert_DBUS */
  MR2 = 0Eh;
  SDS = 00;      /* Start DMA Send */
  while (NOT interrupt)
        idle;
  while (CSD:REQ)
        idle;    /* wait for last SCSI byte
                    transfer so phase
                    is checked */
  if(BSR:Busy_error OR NOT(BSR:End_of_DMA))
        error routine;
  else {         /* DMA END */
        MR2 = 04h;  /* reset DMA bit */
        ICR = 0;
  }
}
```

```
Initiator Receive    /* DATA IN PHASE */
{
  Program DMA Controller;
  TCR = 01h;      /* phase */
  MR2 = 3Eh;
  SDI = 0;        /* Start DMA Init Rx */
  while (NOT interrupt)
        idle;
  /* no need to wait for last SCSI handshake
    done since DMA done implies it is
    checked */
  if (BSR:parity_error OR BSR:busy_error
        or NOT (BSR End of DMA))
               do error routines;
  else  {         /* End of DMA */
        while (CSD:REQ)
               idle;  /* wait for REQ inactive
                 to deassert ACK */
        MR2 = 04h;
  }
}
```

```
Target Receive    /* DATA OUT PHASE */
{
  Program DMA controller;
  TCR = 0;        /*phase*/
  ICR = 08h;
  MR2 = 7Ah;      /*check parity*/
  SDT = 0;        /*Start DMA Targ Rx*/
  while (not interrupt)
        idle;
  /*when End of DMA occurs the last byte
    has been read and checked*/
  if(BSR:parity_error OR NOT(BSR:End_of_DMA)
        error routine;
  else  {         /* End of DMA */
  while (BSR:ACK)
        idle;
  /*Not True End of DMA in MODE N, so wait
    until SCSI bus inactive before
    changing phase */
  MR2 = 40h;
  change phase as required;
  }
}
```

4.0 Device Operation (Continued)

```
Target Send  /* DATA IN PHASE */
{
  Program DMA Controller;
  TCR = 01h;    /* phase */
  ICR = 09h;
  MR2 = 4Ah;
  SDS = 0;      /* Start DMA Send */
  while (NOT interrupt)
        idle;
  if (NOT(BSR:END_of_DMA))
        error;
  else  {       /* DMA end */
        repeat {
               while (CSB:REQ OR BSR:ACK)
               loop count = 3;
               loop count - -;}
               /* decrement */
        until (loop count == 0);
        MR2 = 40h;
  Change phase as required;
  }
}
```

Some explanation of the final part of Target Send is required. In this type of DMA operation it is very difficult to exactly determine the True End of DMA in MODE N. Simply detecting $\overline{\text{REQ}}$ and $\overline{\text{ACK}}$ simultaneously inactive is not enough.

Reference to *Figure 4.8.2* will help to understand the following text.

As shown in *Figure 4.8.2,* $\overline{\text{ACK}}$ going active causes the DRQ for the next byte and also $\overline{\text{REQ}}$ to go inactive. $\overline{\text{ACK}}$ going inactive allows $\overline{\text{REQ}}$ to go active for the next byte. If the INITIATOR is slow removing $\overline{\text{ACK}}$ the μP may sample the SCSI bus after the $\overline{\text{EOP}}$ interrupt at point A. Here both $\overline{\text{REQ}}$ and $\overline{\text{ACK}}$ will be inactive, but there is one more byte to transfer on SCSI. Due to chip timing delays this condition will not last more than 200 ns. A safe way to determine the True End of DMA is to sample $\overline{\text{REQ}}$ and $\overline{\text{ACK}}$ and ONLY when both are inactive in three successive samples will the μP be at point B in the figure.

In MODE E, True End of DMA is correctly decoded and the End of DMA interrupt occurs as a result of this True condition, thus there is no need to sample $\overline{\text{REQ}}$ and $\overline{\text{ACK}}$. For this and other reasons operation in MODE E is strongly recommended.

4.8.2 MODE E Non-Block DMA

Operation for the Non-Block DMA in MODE E is essentially the same as MODE N. A primary difference is that in MODE E True End of DMA is decoded internally and this causes an interrupt. This feature removes the need to check for DMA End before moving to the next phase. Additionally, the use of the ISR allows easier determination of the end condition. For more information on interrupts see Section 6.

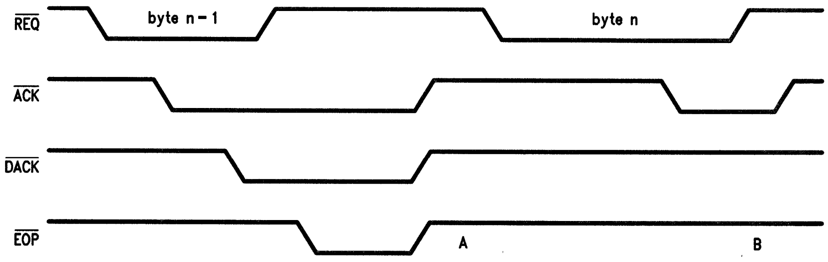
Examples of code are given below.

```
Initiator Receive  /* DATA IN PHASE */
{
  Program DMA controller;
  TCR = 01h;   /* phase */
  ICR = 40h;   /* MODE E */
  MR2 = BEh;
  EMR = 04h;   /* Start DMA INIT Rx */
  while (NOT interrupt)
        idle;
  if (ISR ! = 20h)
        error;
else           /* DMA end */
  MR2 = 04h;
}

Target Send     /* DATA IN PHASE */
{
  Program DMA controller;
  TCR = 01h;   /* phase */
  ICR = 49h;   /* MODE E */
  MR2 = CAh;
  SDS = 0;     /* Start DMA Send */
  while (NOT interrupt)
        idle;
  if (ISR! = 20h)
        error;
  else  {      /* DMA end */
  MR2 = 40h;
  Change phase as required;
  }
}
```

4.9 BLOCK MODE DMA TRANSFERS

In Block Mode the DMA interface uses the DRQ, $\overline{\text{DACK}}$, $\overline{\text{EOP}}$ and READY lines, DRQ is asserted once at the begin-



TL/F/9387-8

FIGURE 4.8.2. Target Send DMA

4.0 Device Operation (Continued)

ning of transfers and deasserted once $\overline{\text{DACK}}$ is received. $\overline{\text{DACK}}$ should be asserted continuously for the duration of all the transfer. $\overline{\text{EOP}}$ should be asserted during the last DMA byte signal when the next DMA byte transfers. The EASI asserts the READY signal when the next DMA byte should be transferred. As for non-block mode the End of DMA interrupt is just $\overline{\text{EOP}}$ in MODE N, but is a True End of DMA in MODE E.

The block mode is intended for systems where the overhead of handing the system busses to and from the μP and DMA controller is too great. The block mode handshake is not necessarily faster than non-block (it may be) but the overall transfer rate is improved once the bus exchange overhead is removed. Of course the μP is prevented from executing for the whole DMA operation.

If a phase mismatch occurs the READY signal is left in the inactive state. The DMA controller must hand back the bus to the μP and the inactive READY signal may need to be gated off. For more detail see Section 9.

When performing DMA as an INITIATOR the $\overline{\text{EOP}}$ signal does not deassert $\overline{\text{ACK}}$ on the SCSI bus in MODE N. Firmware must determine when $\overline{\text{REQ}}$ is inactive after the last SCSI transfer then reset the MR2 DMA bit to deassert $\overline{\text{ACK}}$. In MODE E the EASI correctly handles the deassertion of $\overline{\text{ACK}}$ after True End of DMA.

Programming the EASI in block mode is the same as non-block mode except bit 7 in MR2 should also be set.

4.10 PSEUDO DMA

The system design can utilize EASI DMA logic for non data transfers. This removes the need to poll $\overline{\text{REQ}}/\overline{\text{ACK}}$ and program the assertion/deassertion of the handshake signal. The μP can emulate a DMA controller by asserting $\overline{\text{DACK}}$ and $\overline{\text{EOP}}$ signals. DRQ may be sampled by reading the BSR. In most cases the chip decode logic can be adapted to this use for little or no cost. See Section 9 for further details.

4.11 HALTING A DMA OPERATION

There are three ways to halt a DMA operation apart from a chip or SCSI reset. These methods are: $\overline{\text{EOP}}$, phase mismatch and resetting the DMA MODE bit in MR2.

4.11.1 End of Process

$\overline{\text{EOP}}$ is asserted for a minimum period during the last DMA cycle. The $\overline{\text{EOP}}$ signal generates the End of DMA interrupt in MODE N, and enables the True End of DMA to occur later in MODE E. $\overline{\text{EOP}}$ does not cause the MR2 DMA mode bit to be reset.

4.11.2 DMA Phase Mismatch

If a $\overline{\text{REQ}}$ goes active while there is a phase mismatch the DMA will be halted and an interrupt generated. The EASI will stop driving the SCSI bus when the mismatch occurs. A phase mismatch is when the TCR phase bits do not match the SCSI bus values.

4.11.3 DMA Mode Bit

If $\overline{\text{EOP}}$ is not used, the best method is to reset the MR2 DMA Mode bit. This bit may be reset at any time, and should be reset after an End of DMA interrupt or a phase mismatch. Resetting the bit disables all DMA logic and thus should only be reset at the True End of DMA condition. Additionally, all DMA logic is reset, so this bit must be reset then set again to carry out the next DMA phase.

5.0 Interrupts

5.1 OVERVIEW

The EASI is intended to be used in an interrupt driven environment. MODE E has greatly enhanced the use of interrupts to relieve firmware overhead. This section describes the conditions for and use of each interrupt. Each description explains MODE N then MODE E operation.

Before individually describing each interrupt, an explanation of the use of interrupts is required.

5.2 USING INTERRUPTS

MODE N: In this mode interrupts are controlled by bits in MR2 if control is provided. Not all interrupts can be disabled under software control. When an interrupt occurs, both the BSR and CSD register must be read and analyzed to determine the source of interrupt. Since status is NOT provided for each interrupt, great care should be exercised when determining the interrupt source.

MODE E: In this mode every interrupt can be individually masked and enabled or disabled. In addition, when an interrupt occurs a single register (ISR) can be read to determine the source(s) of interrupt. An associated register (IMR) allows a mask bit to be set for each interrupt. Finally, the design of the logic prevents loss of interrupts occurring after the INT signal goes Active and before the Reset Parity/Interrupt function. In MODE N loss of interrupts can occur.

5.3 SCSI PARITY ERROR

MODE N: If SCSI parity checking is enabled via MR2 bit 5, an interrupt can occur as a result of a read from CSD, a selection/(re)selection, or a DMA receive operation. The parity error bit (bit 5) in the BSR will be set if checking is enabled. An interrupt will occur if Enable Parity Interrupt (bit 4) of MR2 is set. The interrupt is reset by reading HA 7. Following an interrupt the BSR and CSB should contain the values shown below.

Bit 7 **Bit 0**

X	X	1	1	X	X	X	X
EDMA	DRQ	SPER	INT	PHSM	BSY	$\overline{\text{ATN}}$	$\overline{\text{ACK}}$

BSR

Bit 7 **Bit 0**

0	1	X	X	X	X	0	X
$\overline{\text{RST}}$	$\overline{\text{BSY}}$	$\overline{\text{REQ}}$	$\overline{\text{MSG}}$	$\overline{\text{C/D}}$	$\overline{\text{I/O}}$	$\overline{\text{SEL}}$	$\overline{\text{DBP}}$

CSB

5.4 μP PARITY ERROR

MODE N: μP parity is not available under this mode.

MODE E: If bit 6 of the EMR is 1, μP parity will be checked on a write to the EASI via the μP or DMA controller. The parity polarity is determined by bit 5 of the EMR. If bit 6 of the IMR is zero, a μP parity error will set bit 6 of the ISR and cause an interrupt. The interrupt may be reset by writing 01 to bits 2 and 1 of the EMR followed by any other pattern.

5.5 END OF DMA

MODE N: If $\overline{\text{EOP}}$ is asserted during a DMA transfer, bit 7 of the BSR will be set and an interrupt generated if bit 3 of MR2 is 1. $\overline{\text{EOP}}$ is recognized when $\overline{\text{EOP}}$, $\overline{\text{DACK}}$ and either $\overline{\text{IOR}}$ or $\overline{\text{IOW}}$ are all simultaneously active for a minimum period. The interrupt may be reset by reading HA 7. Following an interrupt the BSR and CSB should contain the values shown below.

5.0 Interrupts (Continued)

Bit 7							Bit 0
1	X	X	1	X	X	0	X
EDMA	DRQ	SPER	INT	PHSM	BSY	$\overline{\text{ATN}}$	$\overline{\text{ACK}}$

BSR

Bit 7							Bit 0
0	1	X	X	X	X	0	X
$\overline{\text{RST}}$	$\overline{\text{BSY}}$	$\overline{\text{REQ}}$	$\overline{\text{MSG}}$	$\overline{\text{C/D}}$	$\overline{\text{I/O}}$	$\overline{\text{SEL}}$	$\overline{\text{DBP}}$

CSB

5.6 DMA PHASE MISMATCH

MODE N: When the SCSI $\overline{\text{REQ}}$ goes active during a DMA operation the contents of the TCR are compared with the SCSI phase lines $\overline{\text{C/D}}$, $\overline{\text{MSG}}$ and $\overline{\text{I/O}}$. If the two do not match an interrupt is generated. This interrupt will occur as long as the MR2 DMA bit is set (bit 1), i.e., it cannot be masked in MODE N. The mismatch removes the EASI from driving the SCSI data bus. The interrupt may reset by reading HA 7. Following an interrupt the BSR and CSB should contain the values shown below.

Bit 7							Bit 0
X	0	X	1	0	X	X	X
EDMA	DRQ	SPER	INT	PHSM	BSY	$\overline{\text{ATN}}$	$\overline{\text{ACK}}$

BSR

Bit 7							Bit 0
0	X	X	X	X	X	0	X
$\overline{\text{RST}}$	$\overline{\text{BSY}}$	$\overline{\text{REQ}}$	$\overline{\text{MSG}}$	$\overline{\text{C/D}}$	$\overline{\text{I/O}}$	$\overline{\text{SEL}}$	$\overline{\text{DBP}}$

CSB

MODE E: Bit 4 of the ISR will be set by a DMA phase mismatch—the same conditions as MODE N. The interrupt and setting if ISR bit 4 may be masked by setting bit 4 of the IMR. The interrupt may be reset by writing 01 to bits 2 and 1 of the EMR followed by any other pattern.

5.7 ANY PHASE MISMATCH

MODE N: This feature is not available under this mode.

MODE E: This condition is similar to DMA Phase Mismatch except that it applies to all operations—not just DMA. If the TCR contents do not match the SCSI phase lines when $\overline{\text{REQ}}$ goes active an interrupt is generated and bit 3 set in the ISR. The ISR bit and the interrupt may be masked by setting bit 3 in the IMR. The interrupt may be reset by writing 01 to bits 2 and 1 of the EMR followed by any other pattern.

5.8 BUSY LOSS

MODE N: If bit 2 in MR2 is set the SCSI $\overline{\text{BSY}}$ signal is monitored and an interrupt is generated if $\overline{\text{BSY}}$ is continuously inactive for at least a BUS SETTLE DELAY (400 ns). This interrupt may be reset by reading HA 7. Following an interrupt the BSR and CSB should contain the values shown below, where usually CSB = 00.

Bit 7							Bit 0
X	X	X	1	X	1	X	X
EDMA	DRQ	SPER	INT	PHSM	BSY	$\overline{\text{ATN}}$	$\overline{\text{ACK}}$

BSR

Bit 7							Bit 0
0	0	X	X	X	X	X	X
$\overline{\text{RST}}$	$\overline{\text{BSY}}$	$\overline{\text{REQ}}$	$\overline{\text{MSG}}$	$\overline{\text{C/D}}$	$\overline{\text{I/O}}$	$\overline{\text{SEL}}$	$\overline{\text{DBP}}$

CSB

MODE E: Bit 2 in MR2 performs the same function as in MODE N. When an interrupt is generated bit 2 in the ISR will be set. The ISR bit and the interrupt may be masked by setting bit 2 in the IMR. The interrupt may be reset by writing 01 to bits 2 and 1 of the EMR followed by any other pattern.

5.9 (RE)SELECTION

MODE N: An interrupt will be generated when: $\overline{\text{SEL}}$ is active, $\overline{\text{BSY}}$ is inactive, and the device ID is true. The device ID is determined by the value in the SER. If ANY non-zero bit in the SER has its corresponding SCSI data bit active during selection, the device ID is true. If $\overline{\text{I/O}}$ is active this is a reselection. The interrupt is disabled by writing all zeroes to the SER, and reset by reading HA 7.

If SCSI parity checking is enabled it will be checked and should be valid. Following an interrupt the BSR and CSB should contain the values shown below.

Bit 7							Bit 0
0	0	0	1	X	0	X	0
EDMA	DRQ	SPER	INT	PHSM	BSY	$\overline{\text{ATN}}$	$\overline{\text{ACK}}$

BSR

Bit 7							Bit 0
0	0	0	0	0	0	1	X
$\overline{\text{RST}}$	$\overline{\text{BSY}}$	$\overline{\text{REQ}}$	$\overline{\text{MSG}}$	$\overline{\text{C/D}}$	$\overline{\text{I/O}}$	$\overline{\text{SEL}}$	$\overline{\text{DBP}}$

CSB

MODE E: The functioning of the (re)selection logic is the same as for MODE N. Additionally, bit 1 in the ISR will be set upon a (re)selection interrupt. The interrupt and setting of bit 1 in the ISR may be masked by setting bit 1 in the IMR. The interrupt may be reset by writing all zeroes to the SER then writing 01 to bits 2 and 1 of the EMR followed by any other pattern.

5.10 ARBITRATION COMPLETE

MODE N: No interrupt is generated in MODE N.

MODE E: When bit 0 of the EMR is set to 1 the EASI monitors $\overline{\text{BSY}}$ and $\overline{\text{SEL}}$ to determine a BUS FREE Phase. The EASI then carries out all steps required for bus arbitration. When either arbitration is lost or the arbitration process has completed, bit 0 in the ISR is set and an interrupt generated. The interrupt and bit 0 in the ISR can be masked by setting bit 0 in the IMR. Note that arbitration is not affected by the IMR, just the interrupt.

6.0 Reset Conditions

6.1 GENERAL

There are three ways to reset the EASI; μP chip $\overline{\text{RESET}}$, SCSI bus reset applied externally, SCSI bus reset issued by the EASI.

6.2 CHIP RESET

When the $\overline{\text{RESET}}$ signal is asserted for the required duration the EASI clears ALL internal registers and therefore resets all logic. This action does not create an interrupt or generate a SCSI reset. Since all registers contain zeroes,

6.0 Reset Conditions (Continued)

the EASI is in MODE N under any of the three reset conditions.

6.3 EXTERNAL SCSI RESET

When an SCSI $\overline{\text{RST}}$ is applied externally the EASI resets all registers and logic and issues an interrupt. The only register bits not affected are the Assert RST bit (bit 7) in the ICR and the TARGET Mode bit (bit 6) in MR2. Note that the ISR will contain all zeroes.

6.4 SCSI RESET ISSUED

When the μP sets the Assert RST bit in the ICR, the $\overline{\text{RST}}$ signal goes active. Since the EASI monitors $\overline{\text{RST}}$ also, the same reset actions as in 6.2 apply. The SCSI $\overline{\text{RST}}$ signal will remain active as long as bit 7 in the ICR is set—i.e., until programmed 0 or a chip $\overline{\text{RESET}}$ occurs.

7.0 Loopback Testing

7.1 GENERAL

The DP8490 EASI features loopback testing, enabled by bit 3 of the EMR. When the LOOP bit is set in the EMR all SCSI drivers are disabled at the pads and the signals looped back internally. Additionally, both TARGET and INITIATOR signals may be simultaneously asserted—this is not possible in normal operation. All this enables testing of EASI operation without affecting the SCSI bus.

7.2 SCSI SIGNAL DRIVING/MONITORING

Since each SCSI signal is looped back, testing is accomplished by asserting a signal via the ICR or TCR, then reading back via CSB and BSR to check its state. The code below provides examples. The first example tests SCSI control signals and the second the SCSI data bus. Note that loopback mode must be enabled prior to asserting any signals if they are not to drive the SCSI bus.

```
SCSI_signal_test
{
  ICR = 40h;    /* MODE E */
  EMR = 08h;    /* LOOPback */
  ICR = 5Eh;    /* drive INIT controls */
  TCR = 0Fh;    /* drive TARG controls */
  if((CSB==7Eh) AND (BSR==0Fh))
        ok;
  else error;
}

SCSI_data_test
{
  ICR = 40h;    /* MODE E */
  EMR = 08h;    /* LOOPback */
  ICR = 41h;    /* assert data bus */
  MR2 = 30h;    /* parity check & interrupt */
  ODR = 0AAh;
  if(CSD==0AAh)  ok;
  else  error;
  ODR = 55h;
  if(CSD==55h)  ok;
  else  error;
  if(interrupt)error;  /* no parity errors */
  else ok;
}
```

7.3 (RE)SELECTION AND ARBITRATION

Both these features may be tested in loopback. Note that when checking BSY via the CSB register the "debounced" version of BSY is presented and will be active for 400 ns–800 ns after BSY goes inactive. Logic within the EASI continuously monitors BSY when it becomes inactive to detect a valid Bus Free Phase. One of the outputs of this logic is a clean version of BSY which is accessed through the CSB.

7.4 DATA TRANSFERS

Both programmed-I/O and DMA transfers may be performed. When doing DMA transfers the MR2 TARGET MODE bit (6) must be programmed according to the type of DMA—i.e., set to 1 for TARGET Send or Receive, reset to 0 for INITIATOR Send or Receive. Additionally, the actions of the other SCSI device must be programmed. For example, when testing INITIATOR operations the BSY signal must be set via ICR to simulate a TARGET connected, and the REQ signal must be programmed active and inactive to perform the handshake. The code below shows a single byte transfer as INITIATOR Send.

```
DMA_test
{
  program DMA controller;
  ICR = 40h;     /* MODE E */
  EMR = 08h;     /* LOOPback */
  ICR = 49h;     /* BSY & data bus on */
  MR2 = 0Ah;     /* DMA & EOP interrupt */
  TCR = 08h;     /* assert REQ */
  SDS = 0;       /* Start DMA Send */
  /* DMA cycle with EOP is done here */
  TCR = 0;       /* deassert REQ */
  if(IDR==data byte)  ok;
  else  error;
  if(BSR==90h)  ok;
  else  error;
  /* also should be an EOP interrupt */
}
```

8.0 Extra Features/Compatibility

8.1 OPERATING MODES

This section is intended to clearly indentify the differences between the DP8490 and DP5380. The description covers registers, signals, timing, "bugs" and enhancements. For a more detailed description of register programming and bit functions refer to Sections 3–7.

Before discussing differences a review of DP8490 operation is required. The EASI can be operated in two modes—Normal Mode (MODE N) and Enhanced Mode (MODE E). The EASI is in one or the other mode at any time. MODE E is selected when bit 6 in the INITIATOR COMMAND REGISTER (ICR) is set to 1. A SCSI or external chip reset clears all registers (except MODE REGISTER 2 bit 6) so MODE N is selected as the default. MODE N may also be selected at any time by clearing bit 6 of the ICR to zero. If MODE E

8.0 Extra Features/Compatibility (Continued)

functions have been invoked, selecting MODE N does not in general affect their operation. This means MODE E is used to enable selection of enhanced features which can be used in either mode. Normally an application will choose to remain in MODE E always since all functions are accessible in this mode. MODE N is only required for downward compatibility with a standard 5380 device. The DP5380 operates ONLY in MODE N.

The MODE selection via bit 6 of the ICR enables existing 5380 firmware to run unaltered with the EASI. On the 5380 this is a TEST MODE bit which disables ALL output drivers—NOTHING ELSE! No logic is "tested" and no changes are made to internal logic. Since the μP data bus drivers are also disabled the internal state of the 5380 is inaccessible. This means the TEST MODE bit has very limited application for the end user. By comparison the LOOPBACK bit in the EASI enables thorough testing of device operation.

In summary, current operating firmware should not be using bit 6 of the ICR so the DP8490 uses this bit to enable enhanced operation. *As long as this bit is not set in the EASI the DP8490 appears the same as the NCR, AMD or DP5380. Programming, device operation, and timing sequences are all the same as a 5380.*

8.2 INTERNAL REGISTERS

Figure 8.1 shows the register map of the EASI. Note that in MODE N hex address 7 (HA 7) causes a Parity/Interrupt Reset when read and a Start DMA Initiator Receive when written. In MODE E the read/write EXTRA MODE REGISTER (EMR) is mapped into HA 7. Since the two original functions at HA 7 would conflict, their functions are reproduced by writing to bits 1 and 2 of the EMR. This removes any need to switch between modes. The INTERRUPT MASK REGISTER (IMR) and the INTERRUPT STATUS REGISTER (ISR) are accessed indirectly via the EMR. Setting both bits 1 and 2 of the EMR to 1 enables the next read of address 7 to access the ISR or the next write to access the IMR. Once the access is made subsequent uses of address 7 use the EMR.

Hex Addr	Read Register	Write Register
00	Current SCSI Data	Output Data Register
01	Initiator Command	Initiator Command
02	Mode Register 2	Mode Register 2
03	Target Command	Target Command
04	Current SCSI Bus Status	Select Enable Register
05	Bus and Status	Start DMA Send
06	Input Data Register	Start DMA Targ Rx
07	Reset Parity/Interrupt (MODE N)	Start DMA Init Rx (MODE N)
07	Extra Mode Register (MODE E)	Extra Mode Register (MODE E)
07	Interrupt Status Reg (MODE E and bits 1 and 2 of EMR = 1)	Interrupt Mask Reg (MODE E and bits 1 and 2 of EMR = 1)

FIGURE 8.1. EASI Register Map

8.3 ENHANCEMENT MODE REGISTERS

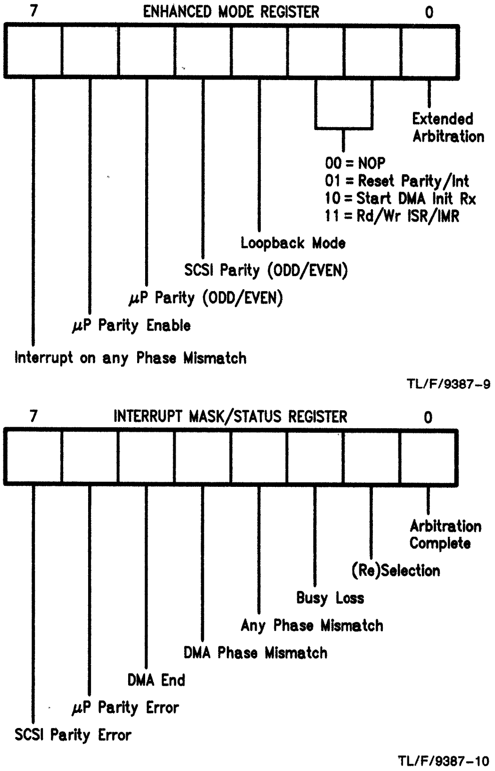
MODE E provides three new registers to provide control for the extra features of the EASI. The EXTRA MODE REGISTER (EMR) is a read/write register which enables or disables the extra functions. Setting an EMR bit to 1 enables the corresponding function while programming a 0 disables it. Note on power-up or reset the EMR is zeroed so all extra functions are disabled. The INTERRUPT MASK REGISTER (IMR) provides the ability to individually mask out interrupts. The INTERRUPT STATUS REGISTER (ISR) shows the status of the interrupt system at any time. When an interrupt occurs the ISR will contain 1's for interrupts that are active, enabled and not masked. *Figure 8.2* shows the format of the three registers.

8.4 EMR FEATURES

8.4.1 Arbitration

In MODE N, arbitration is enabled via bit 0 of MR2. However this requires polling the device for potentially many milliseconds. The μP polls the ICR for the ARBITRATION IN PROGRESS bit (6) which is set once the ASI/EASI detects a bus free phase and enters arbitration. The μP must then wait for the 2.2 μs ARBITRATION DELAY before checking IDs on the bus. No interrupt is given for any of these events.

In MODE E, EASI arbitration is enabled by bit 0 of either MR2 or EMR. The EMR bit enables extended arbitration. The EASI will wait for bus free, arbitrate, wait the 2.2 μs ARBITRATION DELAY and the INTERRUPT the μP if not masked in the IMR. An interrupt occurs if arbitration is complete or has been lost. This feature removes the need to poll the EASI.



TL/F/9387-9

TL/F/9387-10

FIGURE 8.2. MODE E Registers

8.0 Extra Features/Compatibility (Continued)

8.4.2 Loopback Mode

When bit 3 is set in the EMR the EASI disables all SCSI drivers and loops back the signals internally. SCSI Data Out is linked to SCSI Data In. $\overline{BSY}$, $\overline{SEL}$, $\overline{ATN}$, $\overline{RST}$, $\overline{I/O}$, $\overline{C/D}$, $\overline{MSG}$, $\overline{REQ}$ and $\overline{ACK}$ outputs are fed back to their own inputs. This enables testing of EASI operations, including DMA.

8.4.3 SCSI Parity

SCSI parity may be enabled and also cause an interrupt (or be masked). Bit 4 of the EMR allows the polarity of SCSI parity to be set to either ODD or EVEN, with a default of ODD. This feature allows checking of SCSI devices and cable. It also allows an INITIATOR to automatically detect whether a device supports parity. (Send a parity error and check it is reported).

8.4.4 μP Parity

The EASI includes parity for the μP data bus. This enables controllers to validate data while it passes through their data buffers. In common with SCSI parity the μP parity may be: enabled/disabled, interrupt or be masked, be EVEN or ODD polarity.

8.4.5 Phase Mismatch

In MODE N the EASI will only give a Phase Mismatch interrupt during DMA (normally data) transfers. In MODE E an interrupt can also be programmed for a phase mismatch during any phase. The mismatch is detected if $\overline{REQ}$ is active and the phase lines do not match the phase expected as specified in the TCR bits 0 to 2. This feature allows completely interrupt-driven operation of the EASI.

8.5 INTERRUPTS

The EASI has an improved interrupt structure to ease programming. In MODE N the structure is the same as a standard 5380. In this mode interrupts can occur which may only be determined by the lack of other interrupts (e.g., Selection and Phase Mismatch). Interrupts can also be missed if they occur while servicing others. To determine the cause of an interrupt requires reading the BSR and CSB registers then interpreting the results, knowing what the 5380 was currently doing.

In MODE E interrupts can be individually masked via the IMR and active ones determined by reading the ISR. Any unmasked interrupts sets an ISR bit to 1. After the interrupt service routine the μP should perform a Reset Parity/Interrupt function via the EMR—this will ONLY RESET interrupts which had a 1 in the ISR when it was last read. This feature means interrupts will not be lost.

8.6 TIMING

The NCR5380 timing has some aspects which have been improved for MODE E of the EASI. In MODE N timing details are the same as the 5380. Of course the DP5380 ASI remains the same as the NCR5380. The timing improvements are listed below.

1. **True End of DMA:** In MODE N the end of DMA is when the last byte is transferred by the DMA controller, not when the final SCSI transfer is done. In MODE E bit 7 of the TCR shows a true end of DMA status—i.e., the last byte transferred by the later of the two events. This bit is compatible with the NCR CMOS 53C80.
2. **SCSI Handshake after EOP:** in MODE N, INITIATOR receive when a $\overline{REQ}$ is received after $\overline{EOP}$ has been given an $\overline{ACK}$ will be generated although no valid data exists since no DRQ was issued. The EASI will NOT generate this invalid $\overline{ACK}$ while in MODE E.

μP and DMA accesses have relaxed timing on both the EASI and ASI. Data setup/hold and read access times are reduced. Faster handshaking on the SCSI bus, along with faster response to the DMA signals means that higher transfer rates are possible with both devices—typically over 3 MBytes/s.

9.0 Application Guide

This section is intended to show the interface between the μP, EASI and DMA controller (DMAC). *Figure 9.1* shows a general interface when the EASI and DMAC are I/O-mapped devices. This configuration will implement a 2 to 2.5 MBytes/s SCSI port using 2 cycle compressed timing from the 5 MHz DMAC.

Using a faster DMAC and memory may allow the EASI to operate at a higher rate—but of course any system will be limited by the available DMA rate from the SCSI device currently connected to. The interface shown has several features that are examined more closely in the following text.

All the interface signal requirements are satisfied by a PAL device. The memory interface is not shown, only the relevant DMAC and μP lines are included.

The EASI data and address lines connect directly to the μP/DMAC busses. The DRQ output from the EASI goes direct to the DMAC. The $\overline{EOP}$ output from the DMAC goes to the EASI input, via the PAL, but can also be asserted via the PAL since the DMAC output is open-drain.

The PAL is programmed so that the μP can access the EASI in three ways. The three access types are: Register R/W, DMA R/W, DMA with $\overline{EOP}$. Examination of the PAL equations below shows how the μP may perform any of the three basic access types simply by accessing the EASI at different I/O address slots. This enables the μP to simulate a DMAC (pseudo-DMA). DMA mode may then be used for all information transfer phases.

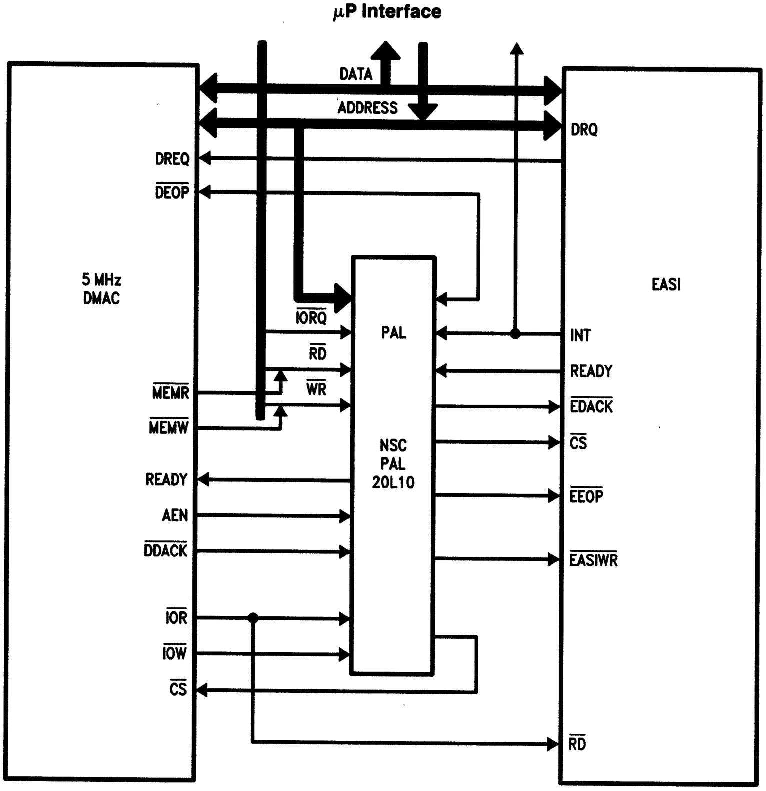
In DMA mode the EASI generates all SCSI handshakes. At all other times the μP is responsible for $\overline{REQ}/\overline{ACK}$ handshakes. Using pseudo-DMA may reduce μP overhead.

When doing DMA transfers via BLOCK MODE and an error occurs, the EASI may not deassert the READY signal. For some DMA controllers this may lock the bus, so the PAL asserts READY and $\overline{EOP}$ to the DMA if an interrupt occurs while READY is false. This completes the current DMA cycle and prevents further DMA for the rest of the block thus allowing the bus to be handed back to the μP for servicing.

The PAL generates $\overline{IOR}$ and $\overline{IOW}$ strobes while the μP is bus master, but the DMAC provides the strobes while it is bus master so the PAL outputs are TRI-STATE.

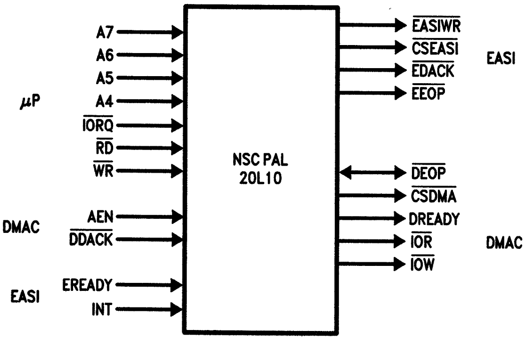
The PAL details are shown in *Figure 9.2* with the signal definitions and equations following.

9.0 Application Guide (Continued)



TL/F/9387-11

FIGURE 9.1. μP/EASI/DMA Interface



TL/F/9387-12

FIGURE 9.2. Interface PAL

9.0 Application Guide (Continued)

```
/CSEASI = /IORQ*/A7*/A6*/A5*/A4*/AEN          ; EASI reg R/W chip select
/EDACK =  /IORQ*/A7*/A6*/A5* A4*/RD           ; μP pseudo-DMA cycle
          /IORQ*/A7*/A6*/A5* A4*/WR
          +/IORQ*/A7*/A6* A5*/A4*/RD          ; μP pseudo-DMA with EOP
          +/IORQ*/A7*/A6* A5* A4*/WR
          +/DDACK                             ; DMAC DMA cycle
/EEOP = /IORQ*/A7*/A6* A5*/A4*/RD*/AEN        ; μP pseudo-DMA with EOP
        +/IORQ*/A7*/A6* A5*/A4*/WR*/AEN
        +/DEOP*EREADY                         ; Prevents EASI from seeing
                                              ; EOP until READY goes high.
IF(/DDACK*/EREADY*INT)/DEOP = /DDACK*/EREADY*INT; on error this will terminate the DMA
transfer.
/CSDMA = /IORQ*/A7*/A6*A5*A4                  ; DMAC register R/W
/DREADY = /EREADY*/INT                        ; EASI not READY and not INT
        +/EREADY*/DDACK                       ; EASI not READY and DMA cycle active
IF(/AEN) /IOR   = /IORQ*/RD                   ; μP I/O Read cycle
IF(/AEN) /IOW   = /IORQ*/WR                   ; μP I/O Write cycle
/EASIWR = /IORQ* WR*/AEN+/IOW*EREADY*AEN      ; Prevents SCSI data being
                                              ; changed before EASI is
                                              ; READY for next byte.
```

FIGURE 9.3. PAL Equations

The μP and DMA signals are defined below

A7–A4	Address bus
$\overline{\text{IORQ}}$	Memory I/O cycle select
$\overline{\text{RD}}$	Read Strobe
$\overline{\text{WR}}$	Write Strobe
AEN	High DMA address enable asserted by DMAC
$\overline{\text{DDACK}}$	DMAC DMA Acknowledge
$\overline{\text{CSDMA}}$	DMA Chip Select
DREADY	Ready signal to DMAC—inserts wait-states when low
$\overline{\text{IOR}}$, $\overline{\text{IOW}}$	I/O data strobes to/from DMAC
$\overline{\text{EASIWR}}$	EASI write strobe.

10.0 Absolute Maximum Ratings*

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Supply Voltage (V_{CC})	−0.5V to +7.0V
DC Input Voltage (V_{IN})	−0.5V to V_{CC} + 0.5V
DC Output Voltage (V_{OUT})	−0.5V to V_{CC} + 0.5V
Storage Temperature Range (T_{STG})	−65°C to +150°C
Power Dissipation (P_D)	500 mW
Lead Temperature (T_L) (Soldering, 10 seconds)	260°C
Electro-Static Discharge Rating	2 kV

*Absolute maximum ratings are those values beyond which damage to the device may occur.

11.0 DC Electrical Characteristics

T_A = 0°C to +70°C, V_{CC} = 5.0V ±5% unless otherwise specified

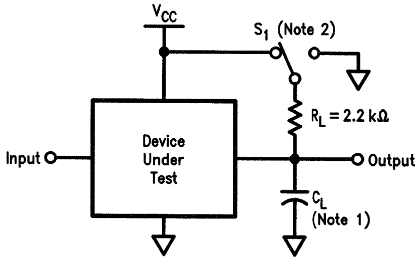
Symbol	Parameter	Conditions	Typ	Limit	Units
V_{IH}	Minimum High Level Input Voltage			2.0	V
V_{IL}	Maximum Low Level Input Voltage			0.8	V
V_{OH1}	Minimum High Level Output Voltage	$\|I_{OUT}\|$ = 20 μA		V_{CC} − 0.1	V
V_{OH2}		$\|I_{OUT}\|$ = 4.0 mA		2.4	V
V_{OL1}	Maximum Low Level Output Voltage	SCSI Bus Pins: $\|I_{OL}\|$ = 48 mA		0.5	V
V_{OL2}		Other Pins: $\|I_{OL}\|$ = 20 μA		0.1	V
V_{OL3}		$\|I_{OL}\|$ = 8.0 mA		0.4	V
I_{IN}	Maximum Input Current	V_{IN} = V_{CC} or GND		±1	μA
I_{OZ}	Maximum TRI-STATE Output Leakage Current	V_{OUT} = V_{CC} or GND		±10	μA
I_{CC}	Supply Current	V_{IN} = V_{CC} or GND SCSI Inputs = 3V	2.5	4	mA

Capacitance T_A = 25°C, f = 1 MHz

Symbol	Parameter (Note 3)	Typ	Units
C_{IN}	Input Capacitance	5	pF
C_{OUT}	Output Capacitance	7	pF

AC Test Conditions

Input Pulse Levels	GND to 3.0V
Input Rise and Fall Time	6 ns
Input/Output Reference Levels	1.3V
TRI-STATE Reference Levels (Note 2)	active low + 0.5V
	active high − 0.5V



TL/F/9387-13

Note 1: C_L = 50 pF including jig and scope capacitance.

Note 2: S1 = Open for push-pull outputs.
S1 = V_{CC} for active low to TRI-STATE.
S1 = GND for active high to TRI-STATE.

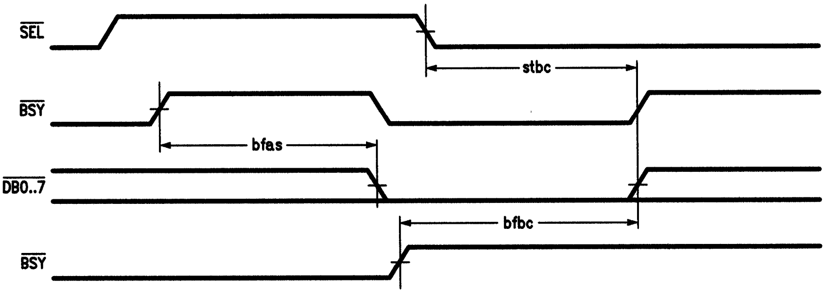
Note 3: This parameter is not 100% tested.

12.0 AC Electrical Characteristics

All parameters are preliminary and subject to change without notice

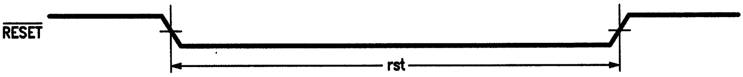
Symbol	Parameter	DP8490			Units
		Min	Typ	Max	
bfas	BSY false to arbitrate start	1200		2200	ns
bfbc	BSY false to bus clear			800	ns
stbc	SEL true to bus clear			500	ns
rst	RESET pulse width	100			ns

12.1 ARBITRATION



TL/F/9387-14

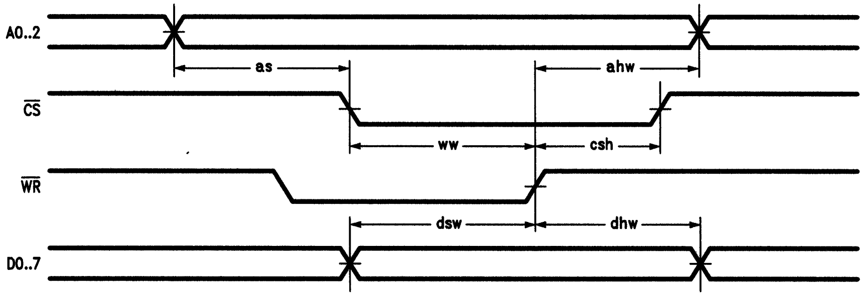
12.2 μP RESET



TL/F/9387-15

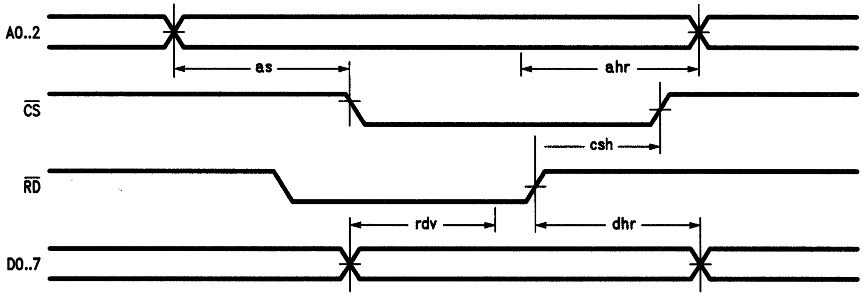
12.0 Electrical Characteristics

12.3 μP WRITE



TL/F/9387-16

12.4 μP READ



TL/F/9387-17

Symbol	Parameter	DP8490			Units
		Min	Typ	Max	
ahr	Address Hold from and of Read Enable (Note 1)	5			ns
ahw	Address Hold from End of Write Enable (Note 2)	5			ns
as	Address Setup to Read or Write Enable (Notes 1,2)	5			ns
csh	CS Hold from End of RD or WR	0			ns
dhr	Data Hold from End of Read Enable (Notes 1,3)	20		60	ns
dhw	μP Data Hold Time from End of WR	10			ns
dsw	Data Setup to End of Write Enable (no μP Parity) (Note 2) (with μP Parity)	35 35			ns ns
rdv	Data Valid from Read Enable (Note 1)			50	ns
ww	Write Enable Width (Note 2)	40			ns

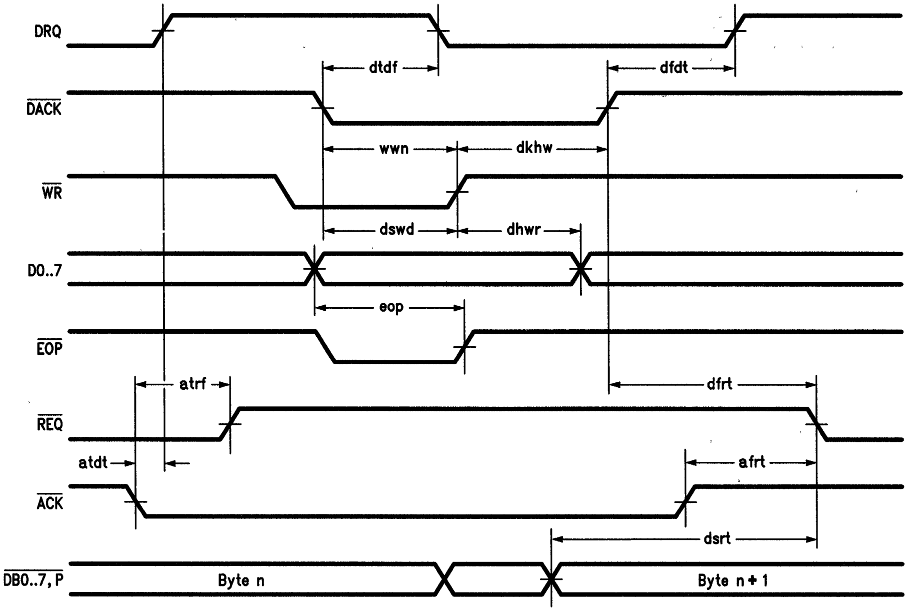
Note 1: Read enable (μP) is CS and RD active.

Note 2: Write enable (μP) is CS and WR active.

Note 3: This includes the RC delay inherent in the test's method. These signals typically turn off after 25 ns, enabling other devices to drive these lines with no contention.

12.0 AC Electrical Characteristics (Continued)

12.5 DMA WRITE (NON-BLOCK MODE) TARGET SEND



TL/F/9387-18

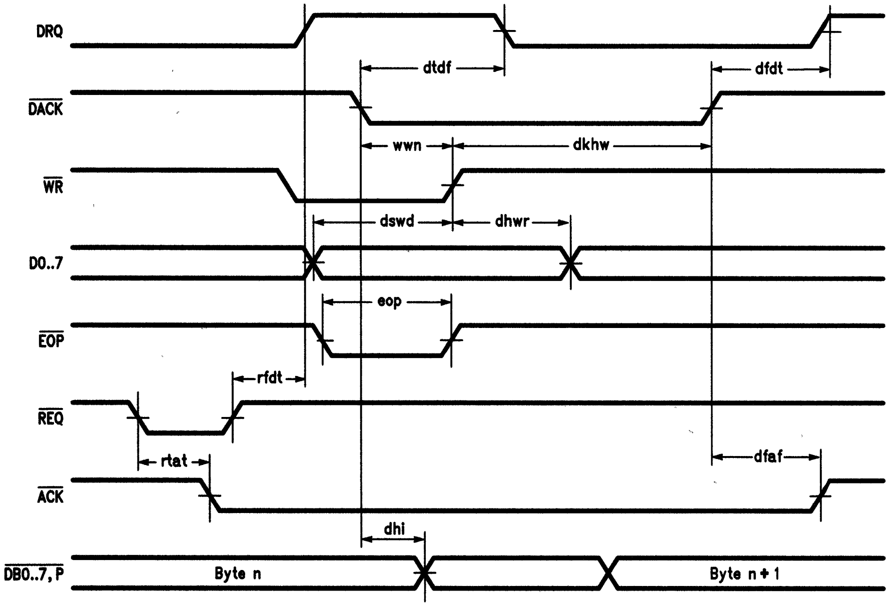
Symbol	Parameter	DP8490			Units
		Min	Typ	Max	
afrt	ACK False to REQ True (DACK or WR False)			75	ns
atdt	ACK True to DRQ True			55	ns
atrf	ACK True to REQ False			100	ns
dfdt	DACK False to DRQ True	30	90		ns
dfrt	DACK False to REQ True (ACK False)			75	ns
dhwr	DMA Data Hold Time from End of WR	10			ns
dkhw	DACK Hold from End of WR	0			ns
dsrt	SCSI Data Setup to REQ True	25			ns
dswd	Data Setup to End of DMA Write Enable (no μP Parity) (Note 1) (with μP Parity)	35 35			ns ns
dtdf	DACK True to DRQ False			45	ns
eop	Width of EOP Pulse (Note 2)	25			ns
wwn	DMA Non-block Mode Write Enable Width (Note 2)	40			ns

Note 1: Write enable (DMA) is DACK and WR active.

Note 2: EOP, DACK, RD/WR must all be true for recognition of EOP.

12.0 AC Electrical Characteristics (Continued)

12.6 DMA WRITE (NON-BLOCK MODE) INITIATOR SEND



TL/F/9387-19

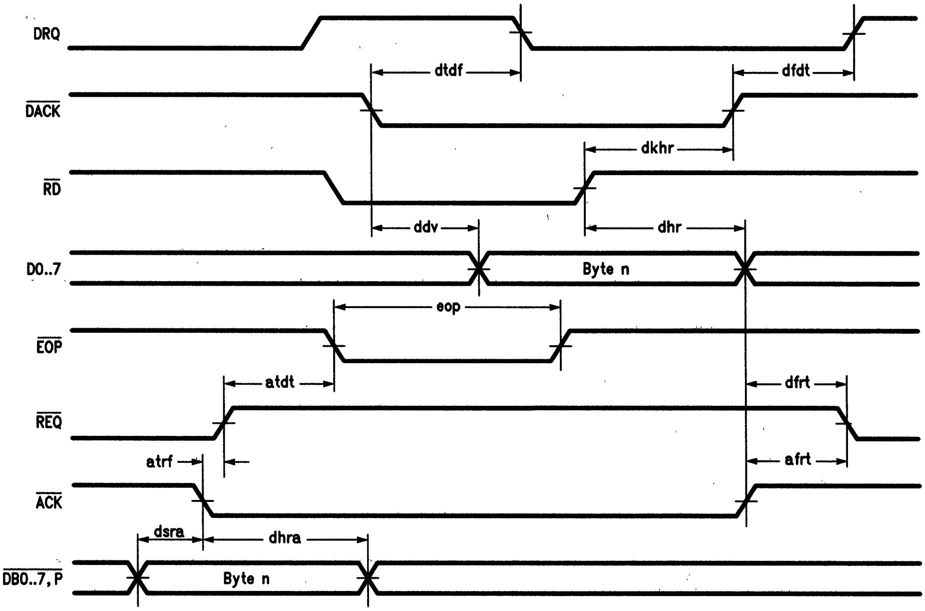
Symbol	Parameter	DP8490			Units
		Min	Typ	Max	
dfaf	DACK False to ACK False			90	ns
dfdt	DACK False to DRQ True	30	90		ns
dhi	SCSI Data Hold from Write Enable-Initiator	15			ns
dhwr	DMA Data Hold Time from End of WR	10			ns
dkhw	DACK Hold from End of WR	0			ns
dswd	Data Setup to End of DMA Write Enable (no μP Parity) (Note 1) (with μP Parity)	35 35			ns ns
dtdf	DACK True to DRQ False			45	ns
eop	Width of EOP Pulse (Note 2)	25			ns
rfdt	REQ False to DRQ True			60	ns
rtat	REQ True to ACK True			80	ns
wwn	DMA Write Enable Width (Note 1)	40			ns

Note 1: Write enable (DMA) is DACK and WR active.

Note 2: EOP, DACK, RD/WR must all be true for recognition of EOP.

12.0 AC Electrical Characteristics (Continued)

12.7 DMA READ (NON-BLOCK MODE) TARGET RECEIVE



TL/F/9387-20

Symbol	Parameter	DP8490			Units
		Min	Typ	Max	
afrt	ACK False to REQ True (DACK or WR False)			75	ns
atdt	ACK True to DRQ True			55	ns
atrf	ACK True to REQ False			100	ns
ddv	DMA Data Valid from Read Enable (Note 1)			40	ns
dfdt	DACK False to DRQ True	30	90		ns
dfrt	DACK False to REQ True (ACK False)			75	ns
dhr	Data Hold from End of Read Enable (Notes 1, 2)	20		60	ns
dhra	SCSI Data Hold from ACK True	15			ns
dkhr	DACK Hold from End of RD	0			ns
dsra	SCSI Data Setup Time to ACK True	10			ns
dtdf	DACK True to DRQ False			45	ns
eop	Width of EOP Pulse (Note 3)	25			ns

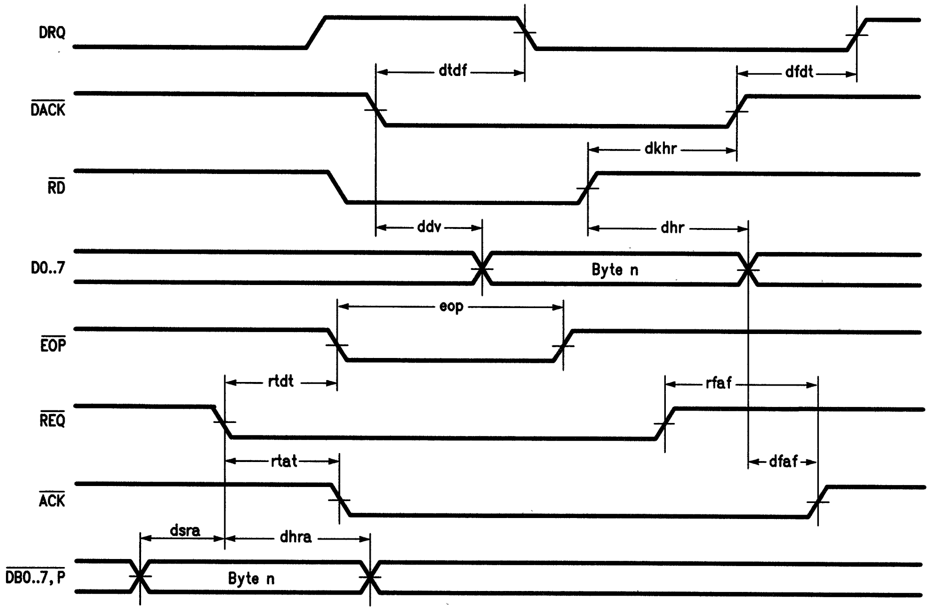
Note 1: Read enable (DMA) is DACK and RD active.

Note 2: This includes the RC delay inherent in the test's method. These signals typically turn off after 25 ns enabling other devices to drive these lines with no contention.

Note 3: EOP, DACK, RD/WR must all be true for recognition of EOP.

12.0 AC Electrical Characteristics (Continued)

12.8 DMA READ (NON-BLOCK MODE) INITIATOR RECEIVE



TL/F/9387-21

Symbol	Parameter	DP8490			Units
		Min	Typ	Max	
ddv	DMA Data Valid from Read Enable (Note 1)			40	ns
dfaf	DACK False to ACK False (REQ False)			90	ns
dfdt	DACK False to DRQ True	30	90		ns
dhr	Data Hold from End of Read Enable (Notes 1, 2)	20		60	ns
dhra	SCSI Data Hold from REQ True	15			ns
dkhr	DACK Hold from End of RD	0			ns
dsra	SCSI Data Setup Time to REQ True	10			ns
dtdf	DACK True to DRQ False			45	ns
eop	Width of EOP Pulse (Note 3)	25			ns
rfaf	REQ False to ACK False (DACK False)			90	ns
rtat	REQ True to ACK True			80	ns
rtdt	REQ True to DRQ True			70	ns

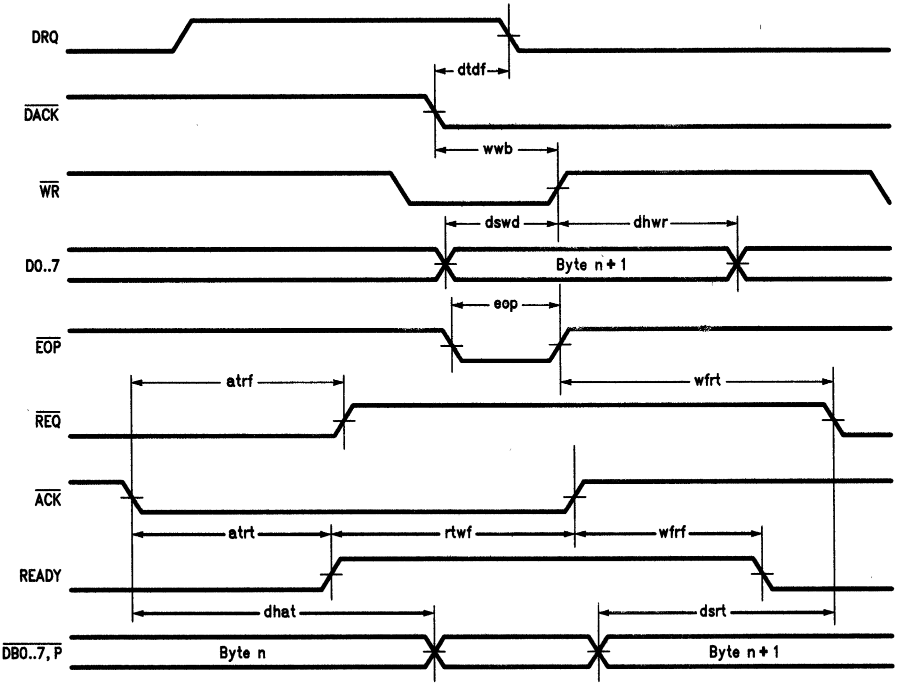
Note 1: Read enable (DMA) is DACK and RD active.

Note 2: This includes the RC delay inherent in the test's method. These signals typically turn off after 25 ns enabling other devices to drive these lines with no contention.

Note 3: EOP, DACK, RD/WR must all be true for recognition of EOP.

12.0 AC Electrical Characteristics (Continued)

12.9 DMA WRITE (BLOCK MODE) TARGET SEND



TL/F/9387-22

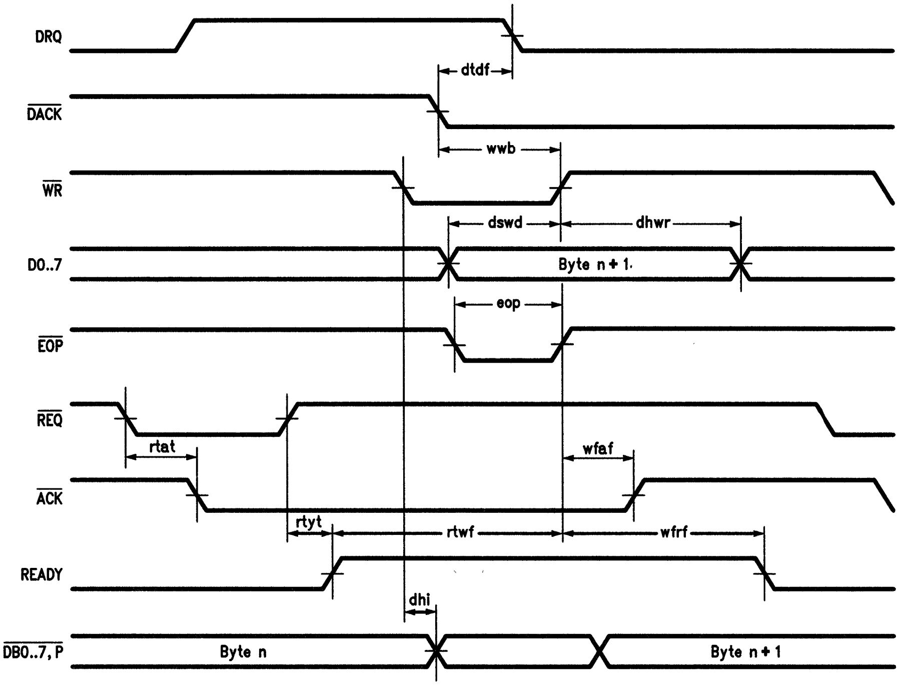
Symbol	Parameter	DP8490			Units
		Min	Typ	Max	
afrt	ACK False to REQ True (DACK or WR False)			75	ns
atrf	ACK True to REQ False			100	ns
atrt	ACK True to READY True			50	ns
dhat	SCSI Data Hold from ACK True	40			ns
dhwr	DMA Data Hold Time from End of WR	10			ns
dsrt	SCSI Data Setup to REQ True	35			ns
dswd	Data Setup to End of DMA Write Enable (no µP Parity) (Note 1) (with µP Parity)	35 35			ns ns
dtdf	DACK True to DRQ False			45	ns
eop	Width of EOP Pulse (Note 2)	25			ns
rtwf	READY True to WR False	40			ns
wfrf	WR False to READY False			50	ns
wfrt	WR False to REQ True (ACK False)			80	ns
wwb	DMA Write Enable Width (Note 1)	40			ns

Note 1: Write enable (DMA) is DACK and WR active.

Note 2: EOP, DACK, RD/WR must all be true for recognition of EOP.

12.0 AC Electrical Characteristics (Continued)

12.10 DMA WRITE (BLOCK MODE) INITIATOR SEND



TL/F/9387-23

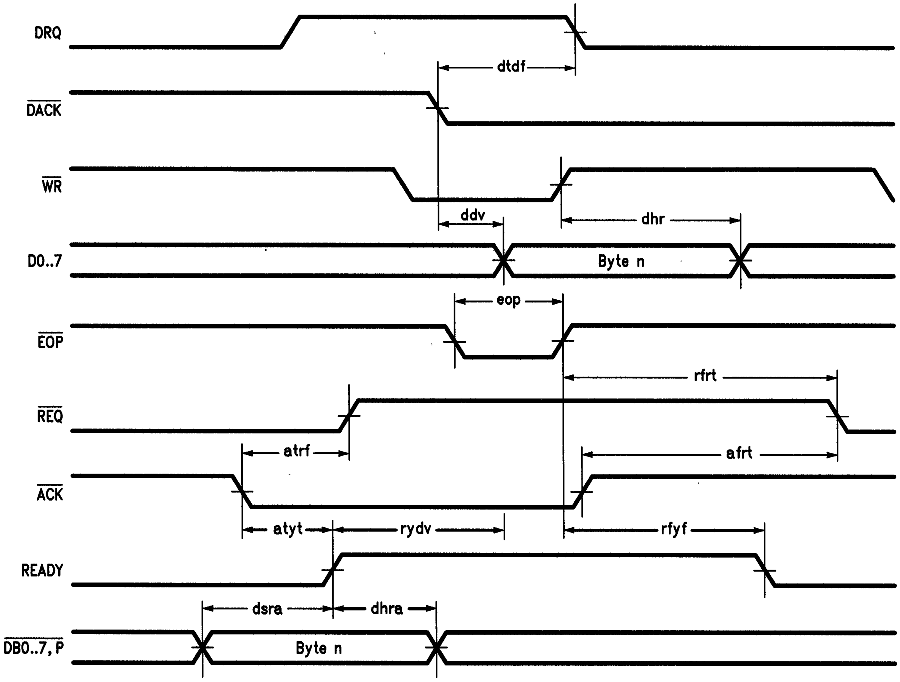
Symbol	Parameter	DP8490			Units
		Min	Typ	Max	
dhi	SCSI Data Hold from Write Enable	15			ns
dhwr	DMA Data Hold Time from End of WR	10			ns
dswd	Data Setup to End of DMA Write Enable (no μP Parity) (Note 1) (with μP Parity)	35 35			ns ns
dtdf	DACK True to DRQ False			45	ns
eop	Width of EOP Pulse (Note 2)	25			ns
rfyt	REQ False to READY True			60	ns
rtat	REQ True to ACK True			80	ns
rtwf	READY True to WR False	40			ns
wfaf	WR False to ACK False (REQ False)			95	ns
wfrf	WR False to READY False			50	ns
wwb	DMA Write Enable Width (Note 1)	40			ns

Note 1: Write enable (DMA) is DACK and WR active.

Note 2: EOP, DACK, RD/WR must all be true for recognition of EOP.

12.0 AC Electrical Characteristics (Continued)

12.11 DMA WRITE (BLOCK MODE) TARGET RECEIVE



TL/F/9387-24

Symbol	Parameter	DP8490			Units
		Min	Typ	Max	
afrt	ACK False to REQ True (DACK or WR False)			75	ns
atrf	ACK True to REQ False			100	ns
atyt	ACK True to READY True			50	ns
ddv	DMA Data Valid from Read Enable (Note 1)			40	ns
dhr	Data Hold from End of Read Enable (Notes 1,2)	20		60	ns
dhra	SCSI Data Hold from ACK True	15			ns
dsra	SCSI Data Setup Time to ACK True	10			ns
dtdf	DACK True to DRQ False			45	ns
eop	Width of EOP Pulse (Note 3)	25			ns
rfrt	RD False to REQ True (ACK False)			75	ns
rfyf	RD False to READY False			45	ns
rydv	READY True to Data Valid			20	ns

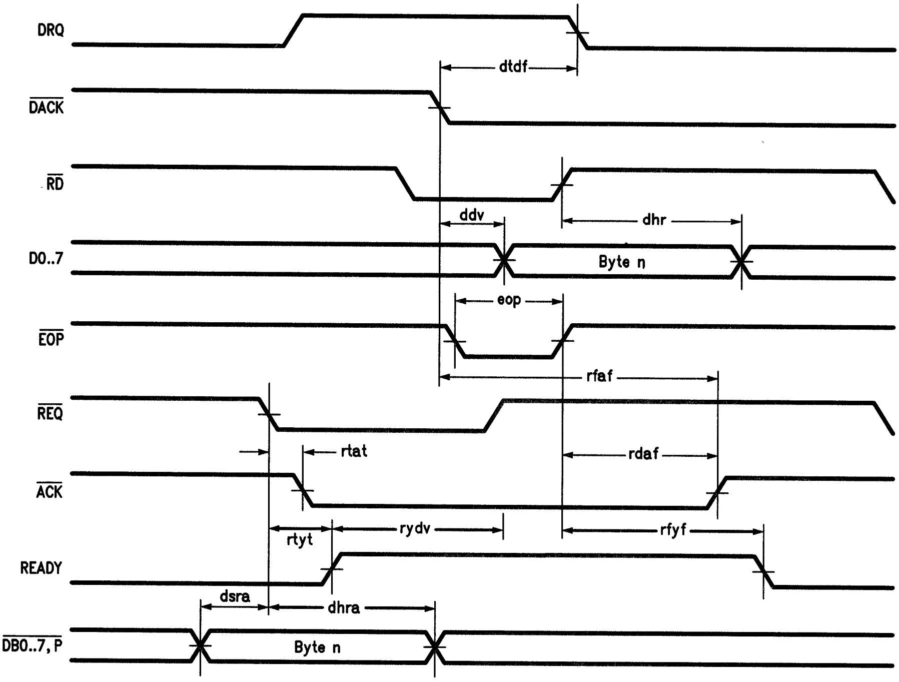
Note 1: Read enable (DMA) is DACK and RD active.

Note 2: This includes the RC delay inherent in the test's method. These signals typically turn off after 25 ns enabling other devices to drive these lines with no contention.

Note 3: EOP, DACK, RD/WR must all be active for recognition of EOP.

12.0 AC Electrical Characteristics (Continued)

12.12 DMA READ (BLOCK MODE) INITIATOR RECEIVE



TL/F/9387-25

Symbol	Parameter	DP8490			Units
		Min	Typ	Max	
ddv	DMA Data Valid from Read Enable (Note 1)			40	ns
dhr	Data Hold from End of Read Enable (Notes 1,2)	20		60	ns
dhra	SCSI Data Hold from REQ True	15			ns
dsra	SCSI Data Setup Time to REQ True	10			ns
dtdf	DACK True to DRQ False			45	ns
eop	Width of EOP Pulse (Note 3)	25			ns
rdaf	RD False to ACK False (REQ False)			120	ns
rfaf	REQ False to ACK False (DACK False)			90	ns
rfyf	RD False to READY False			45	ns
rtat	REQ True to ACK True			80	ns
rtyt	REQ True to READY True			65	ns
rydv	READY True to Data Valid			20	ns

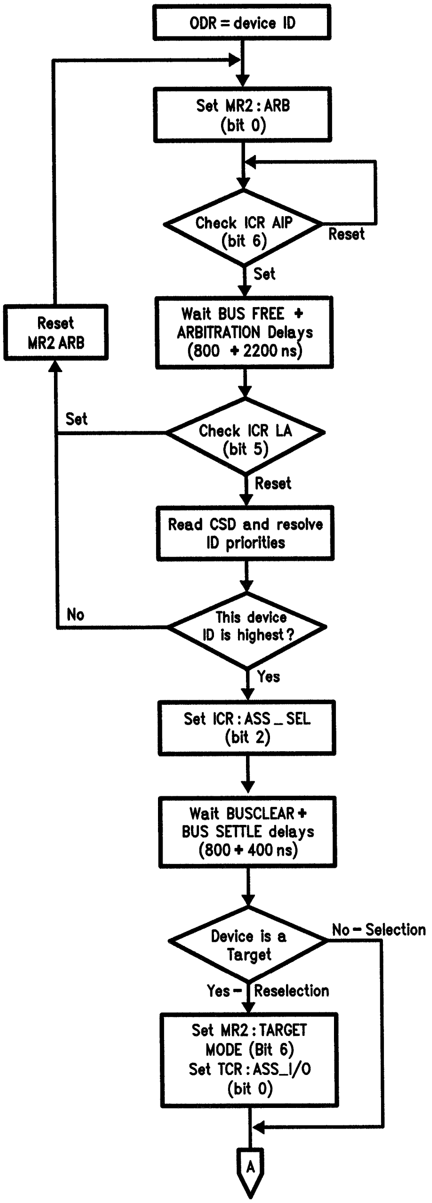
Note 1: Read enable (DMA) is DACK and RD active.

Note 2: This includes the RC delay inherent in the test's method. These signals typically turn off after 25 ns enabling other devices to drive these lines with no contention.

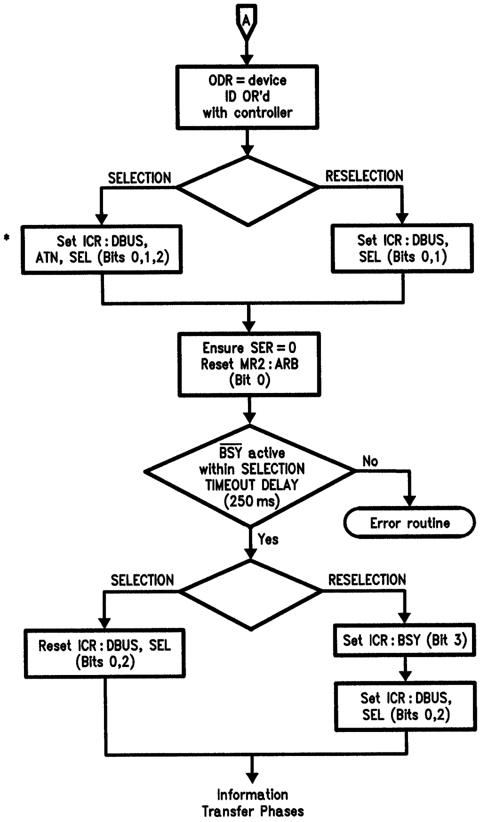
Note 3: EOP, DACK, RD/WR must all be active for recognition of EOP.

Appendix A1

Arbitration & (Re)Selection (Normal Mode)



TL/F/9387-26

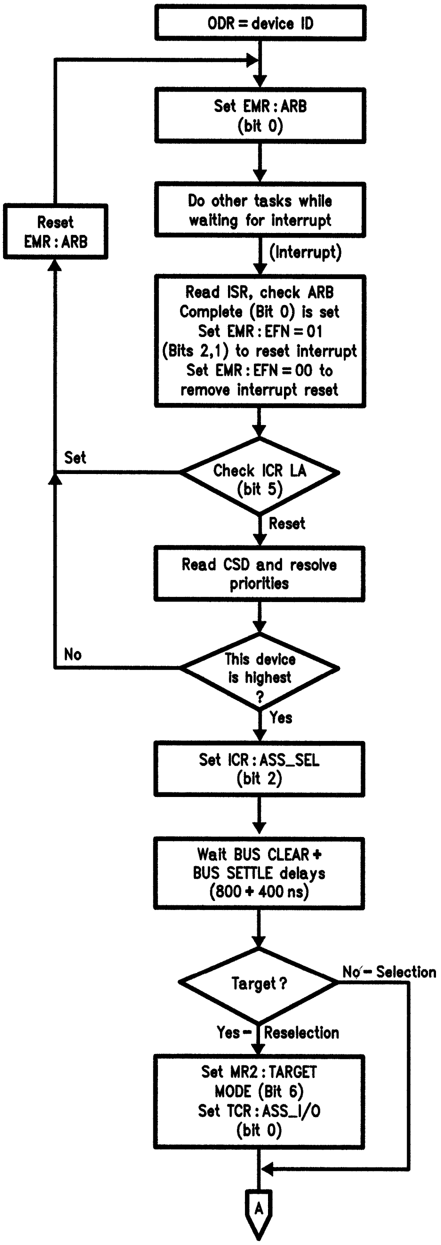


TL/F/9387-27

*Only set ATN if Select with ATN is desired.

Appendix A1 (Continued)

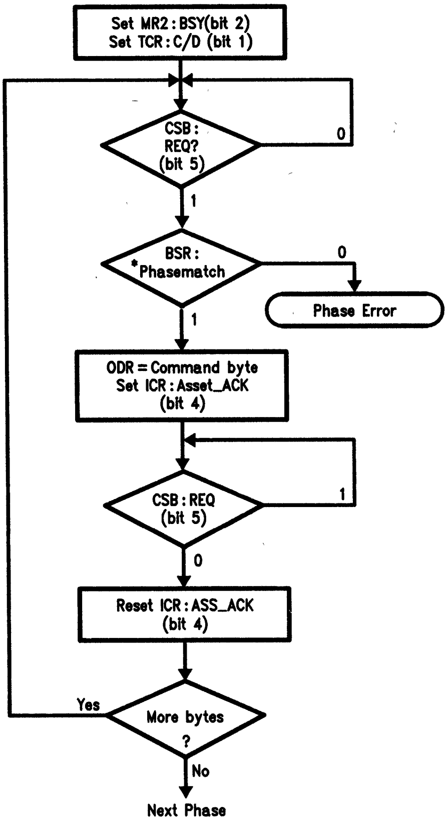
Arbitration & (Re)Selection (Enhanced Mode)



TL/F/9387–28

Appendix A1 (Continued)

Command Transfer (Initiator)

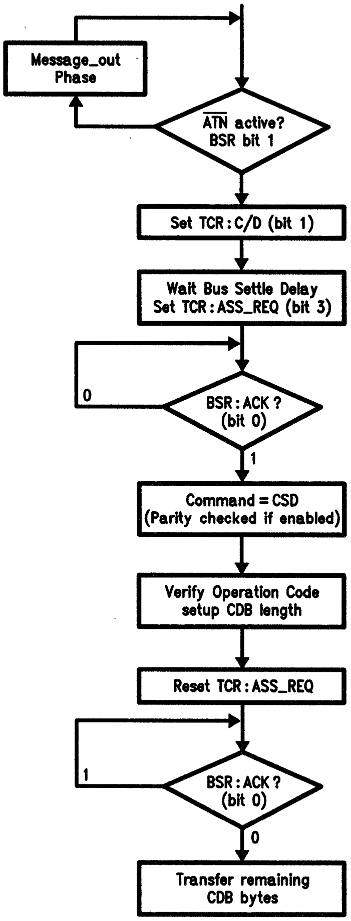


TL/F/9387-29

*This step unnecessary in MODE E if the EMR : APHS (bit 7) is enabled. Logic automatically checks the phase on any transfer and interrupts on an error.

Appendix A1 (Continued)

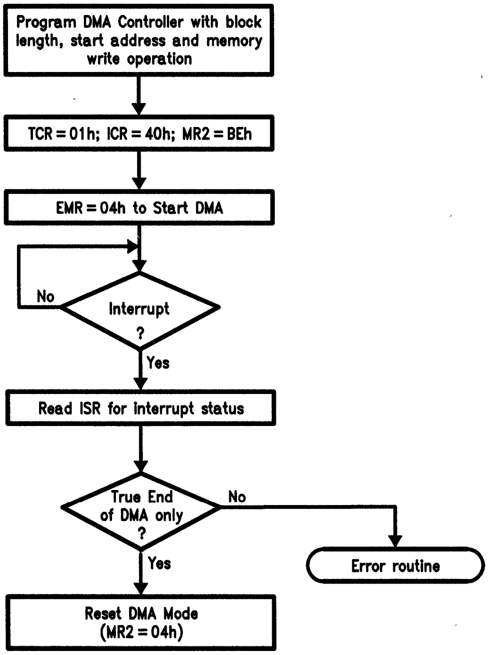
Command Transfer (Target)


Message_out Phase
$\overline{\text{ATN}}$ active? BSR bit 1
Set TCR: C/D (bit 1)
Wait Bus Settle Delay Set TCR: ASS_REQ (bit 3)
0
BSR: ACK? (bit 0)
1
Command = CSD (Parity checked if enabled)
Verify Operation Code setup CDB length
Reset TCR: ASS_REQ
1
BSR: ACK? (bit 0)
0
Transfer remaining CDB bytes

TL/F/9387-30

Appendix A1 (Continued)

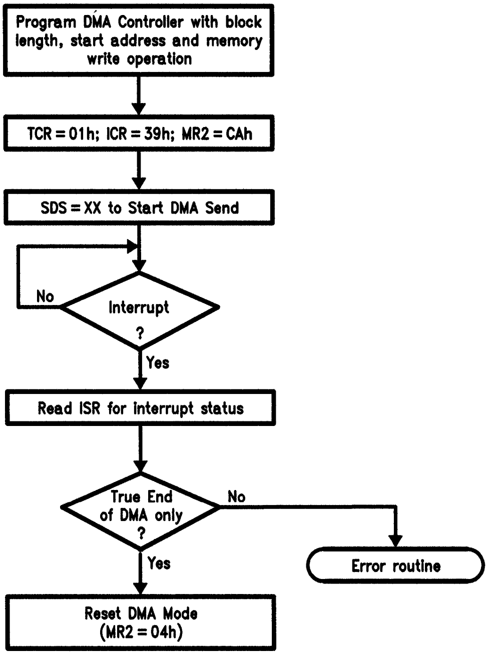
Block Mode DMA Transfer Initiator Receive (MODE E)



Program DMA Controller with block length, start address and memory write operation
TCR = 01h; ICR = 40h; MR2 = BEh
EMR = 04h to Start DMA
No
Interrupt ?
Yes
Read ISR for interrupt status
True End of DMA only ?
No
Yes
Error routine
Reset DMA Mode (MR2 = 04h)

TL/F/9387-31

Block Mode DMA Transfer Target Send (MODE E)



Program DMA Controller with block length, start address and memory write operation
TCR = 01h; ICR = 39h; MR2 = CAh
SDS = XX to Start DMA Send
No
Interrupt ?
Yes
Read ISR for interrupt status
True End of DMA only ?
No
Yes
Error routine
Reset DMA Mode (MR2 = 04h)

TL/F/9387-32

Appendix A2

Register Chart

Read

Current SCSI Data (CSD)

Bit 7							Bit 0
$\overline{DB7}$	$\overline{DB6}$	$\overline{DB5}$	$\overline{DB4}$	$\overline{DB3}$	$\overline{DB2}$	$\overline{DB1}$	$\overline{DB0}$

Initiator Command Register (ICR)

Bit 7							Bit 0
$\overline{RST}$	AIP	LA	$\overline{ACK}$	$\overline{BSY}$	$\overline{SEL}$	$\overline{ATN}$	DBUS

Mode Register 2 (MR2)

Bit 7							Bit 0
BLK	TARG	PCHK	PINT	EOP	BSY	DMA	ARB

Target Command Register (TCR)

Bit 7							Bit 0
0	0	0	0	$\overline{REQ}$	$\overline{MSG}$	$\overline{C/D}$	$\overline{I/O}$

Current SCSI Bus Status (CSB)

Bit 7							Bit 0
$\overline{RST}$	$\overline{BSY}$	$\overline{REQ}$	$\overline{MSG}$	$\overline{C/D}$	$\overline{I/O}$	$\overline{SEL}$	$\overline{DBP}$

Bus and Status Register (BSR)

Bit 7							Bit 0
EDMA	DRQ	SPER	INT	PHSM	BSY	$\overline{ATN}$	$\overline{ACK}$

Input Data Register (IDR)

Bit 7							Bit 0
$\overline{DB7}$	$\overline{DB6}$	$\overline{DB5}$	$\overline{DB4}$	$\overline{DB3}$	$\overline{DB2}$	$\overline{DB1}$	$\overline{DB0}$

Reset Parity/Interrupt (RPI)—MODE N

Bit 7							Bit 0
X	X	X	X	X	X	X	X

Enhanced Mode Register (EMR)

Bit 7							Bit 0
APHS	MPEN	MPOL	SPOL	LOOP	EFN1	EFN0	ARB

Interrupt Status Register (ISR)

Bit 7							Bit 0
SPE	MPE	EDMA	DPHS	APHS	BSY	SEL	ARB

Target Command Register (TCR)—MODE E

Bit 7							Bit 0
(true) EDMA	0	0	0	$\overline{REQ}$	$\overline{MSG}$	$\overline{C/D}$	$\overline{I/O}$

X = Unknown

Write

Output Data Register (ODR)

Bit 7							Bit 0
$\overline{DB7}$	$\overline{DB6}$	$\overline{DB5}$	$\overline{DB4}$	$\overline{DB3}$	$\overline{DB2}$	$\overline{DB1}$	$\overline{DB0}$

Initiator Command Register (ICR)

Bit 7							Bit 0
$\overline{RST}$	MODE E	DIFF EN	$\overline{ACK}$	$\overline{BSY}$	$\overline{SEL}$	$\overline{ATN}$	DBUS

Mode Register 2 (MR2)

Bit 7							Bit 0
BLK	TARG	PCHK	PINT	EOP	BSY	DMA	ARB

Target Command Register (TCR)

Bit 7							Bit 0
X	X	X	X	$\overline{REQ}$	$\overline{MSG}$	$\overline{C/D}$	$\overline{I/O}$

Select Enable Register (SER)

Bit 7							Bit 0
$\overline{DB7}$	$\overline{DB6}$	$\overline{DB5}$	$\overline{DB4}$	$\overline{DB3}$	$\overline{DB2}$	$\overline{DB1}$	$\overline{DB0}$

Start DMA Send (SDS)

Bit 7							Bit 0
X	X	X	X	X	X	X	X

Start DMA Target Receive (SDT)

Bit 7							Bit 0
X	X	X	X	X	X	X	X

Start DMA Initiator Receive (SDI)—MODE N

Bit 7							Bit 0
X	X	X	X	X	X	X	X

Enhanced Mode Register (EMR)

Bit 7							Bit 0
APHS	MPEN	MPOL	SPOL	LOOP	EFN1	EFN0	ARB

Interrupt Mask Register (IMR)

Bit 7							Bit 0
SPE	MPE	EDMA	DPHS	APHS	BSY	SEL	ARB

Target Command Register (TCR)—MODE E

Bit 7							Bit 0
X	X	X	X	$\overline{REQ}$	$\overline{MSG}$	$\overline{C/D}$	$\overline{I/O}$

X = Don't Care