National Semiconductor

May 1989

DP5380 Asynchronous SCSI Interface (ASI)

General Description

The DP5380 ASI 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 ASI 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 DP5380 is pin and program compatible with the NMOS NCR5380 device. NCR5380 or AM5380 applications can use it with no changes to hardware or software. The DP5380 is available in a 40-pin DIP or a 44-pin PCC.

The ASI 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 ASI easily interfaces to a DMA controller using normal or Block Mode. The ASI can be used in either a polled or interrupt-driven environment.

Features

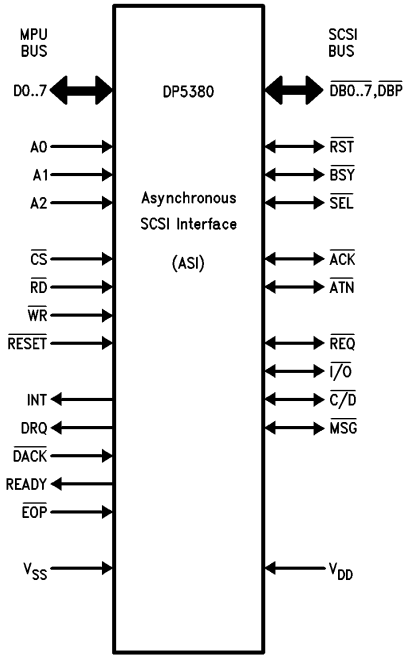
SCSI Interface

- Supports TARGET and INITIATOR roles
- Parity generation with optional checking
- Arbitration support
- 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

Connection Diagram



TL/F/9756–1

Table of Contents

1.0 Functional Description

1.1 OVERVIEW

The ASI 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.

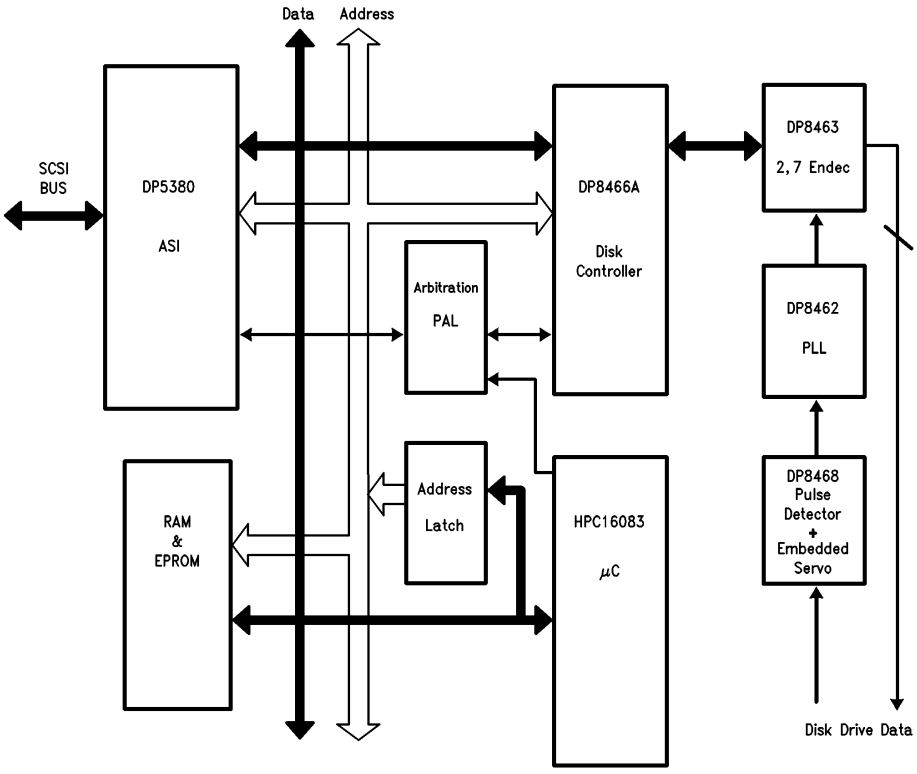
For best performance a DMA controller can be easily interfaced directly to the ASI. The ASI provides request/acknowledge and wait-state signals for the DMA interface.

The SCSI bus is easily controlled via the ASI 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 ASI provides hardware support for much of the protocol.

The ASI 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—the μP has only to check for highest priority.
- Selection or Reselection of any bus device. The ASI 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.

Figure 1 shows an ASI 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.



TL/F/9756–2

FIGURE 1. ASI Application

1.0 Functional Description (Continued)

1.2 μP INTERFACE

Figure 2 shows a block diagram of the ASI. Key blocks within the ASI 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 ASI has three interfaces, one to SCSI, one to a DMA controller and the third to a μP. The internal registers control all operations of the ASI.

The μP interface consists of non-multiplexed address and data busses with associated control signals. Address decode logic selects a register for reading or writing. The address lines A0–2 select the register for μP accesses while for DMA accesses the address lines are ignored.

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).

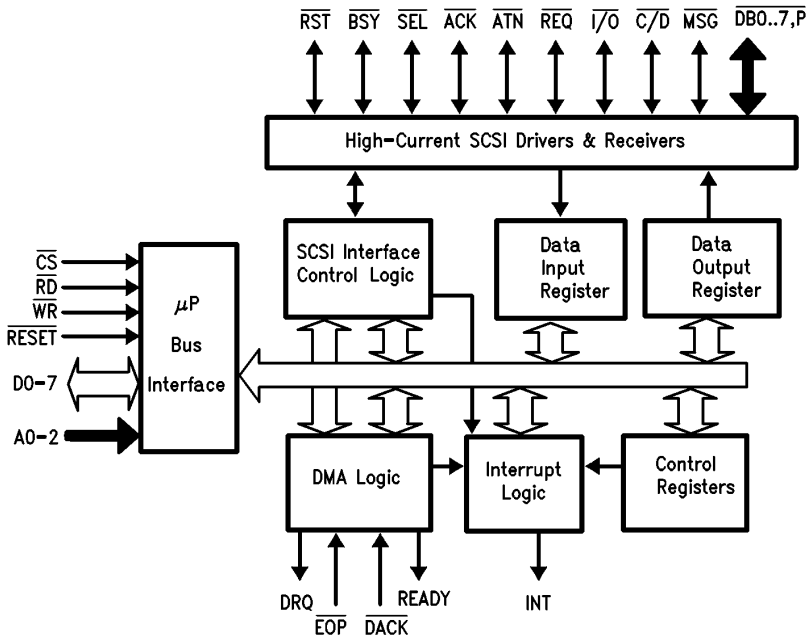
1.3 DMA INTERFACE

The DMA logic interfaces to single-cycle, block mode, flow-through or fly-by controllers. Single byte transfers are accomplished via the DRQ/$\overline{DACK}$ handshake signals. Block mode transfers use the READY output to control the speed (insert wait-states). An End Of Process ($\overline{EOP}$) input from the DMA controller signals the ASI to halt DMA transfers. An interrupt can be generated for DMA completion or an error (see Section 5.0). All DMA data passes through the SCSI data input and output registers, automatically selected during DMA cycles.

1.4 SCSI INTERFACE

The ASI 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 ASI 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, $\overline{BSY}$ monitoring for bus disconnection, bus reset and parity generation and checking.



TL/F/9756–3

FIGURE 2. ASI Block Diagram

1.0 Functional Description (Continued)

The ASI arbitration logic controls arbitration for use of the SCSI bus. The µP programs the SCSI device ID into the ASI, then sets the ARBITRATE bit. The INITIATOR COMMAND REGISTER (ICR) is read to determine when arbitration has started and whether it is won or lost.

The $\overline{BSY}$ signal is continously monitored to detect bus disconnection and bus free phases. The ASI 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 ASI 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.

1.5 PARITY

The ASI provides for parity protection on the SCSI interface. The data bus has eight data bits and one parity bit. The parity may be enabled via a register bit. A parity error can be programmed to cause an interrupt.

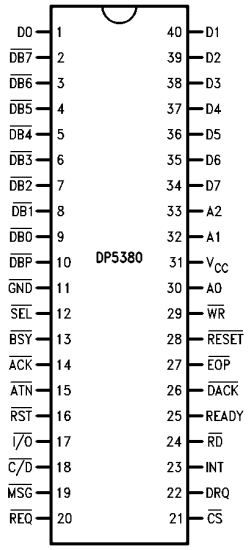
2.0 Pin Descriptions

Symbol	DIP	PCC	Type	Function
$\overline{CS}$	21	24	I	**Chip Select:** an active low enable for read or write operations, accessing the register selected by A0 . . . 2.
A0 . . . 2	30, 32, 33	33, 36, 37	I	**Address 0 . . . 2:** these three signals are used with $\overline{CS}$, $\overline{RD}$, and $\overline{WR}$ to address a register for read or write.
$\overline{RD}$	24	27	I	**Read:** an active low enable for reading an internal register selected by A0 . . . 2 and enabled by $\overline{CS}$. It also selects the Input Data Register when used with $\overline{DACK}$.
$\overline{WR}$	29	32	I	**Write:** an active low enable for writing an internal register selected by A0 . . . 2 and enabled by $\overline{CS}$. It also selects the Output Data Register when used with $\overline{DACK}$.
$\overline{RESET}$	28	31	I	**Reset:** an active low input with a Schmitt trigger. Clears all internal registers. (SCSI $\overline{RST}$ unaffected).
D0 . . . 7	1, 40–34	2, 44–38	I/O	**Data 0 . . . 7:** bidirectional TRI-STATE® signals connecting the active high µP data bus to the internal registers.
INT	23	26	O	**Interrupt:** an active high output to the µP when an error has occurred, an event requires service or has completed.
$\overline{DRQ}$	22	25	O	**DMA Request:** an active high output asserted when the data register is ready to read or written. DRQ occurs only if DMA mode is enabled. The signal is cleared by $\overline{DACK}$.
$\overline{DACK}$	26	29	I	**DMA Acknowledge:** an active low input that resets DRQ and addresses the data registers for input or output transfers. $\overline{DACK}$ is used instead of $\overline{CS}$ 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.
$\overline{EOP}$	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.
$\overline{DB0 . . . 7}$ $\overline{DBP}$	9 . . . 2, 10	10 . . . 3, 11	I/O	**DB0 . . . 7, DBP:** SCSI data bus with parity. $\overline{DB7}$ is the MSB and is the highest priority during arbitration. Parity is ODD. Parity is always generated and can be optionally checked. Parity is not valid during arbitration.
$\overline{RST}$	16	18	I/O	**Reset:** SCSI reset, monitored and can be set by ASI.
$\overline{BSY}$	13	15	I/O	**Busy:** indicates the SCSI bus is being used. Can be driven by TARGET or INITIATOR.
$\overline{SEL}$	12	14	I/O	**Select:** used by the INITIATOR to select a TARGET or by the TARGET to reselect an INITIATOR.
$\overline{ACK}$	14	16	I/O	**Acknowledge:** driven by the INITIATOR and received by the TARGET as part of the REQ/$\overline{ACK}$ handshake.
$\overline{ATN}$	15	17	I/O	**Attention:** driven by the INITIATOR to indicate an attention condition to the TARGET.

2.0 Pin Descriptions (Continued)

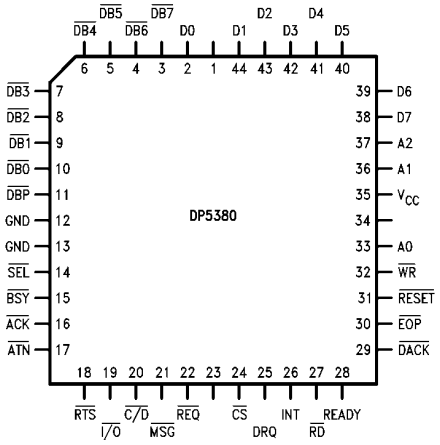
Symbol	DIP	PCC	Type	Function
$\overline{REQ}$	20	22	I/O	**Request:** driven by the TARGET and received by the INITIATOR as part of the $\overline{REQ}$/$\overline{ACK}$ handshake.
$\overline{I/O}$	17	19	I/O	**Input/Output:** driven by the TARGET to control the direction of transfers on the SCSI bus.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

2.1 Connection Diagrams



TL/F/9756–4

Order Number DP5380N
See NS Package Number N40A



TL/F/9756–5

Order Number DP5380V
See NS Package Number V44A

3.0 Register Description

3.1 GENERAL

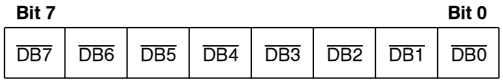
The DP5380 ASI is a register-based device with eight addressable locations. Some addresses have dual functions depending upon whether they are being read from or written to. Device operation is described in Section 4.

Figure 3.2 summarises the register map. 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.

3.2 REGISTERS

OUTPUT DATA REGISTER (ODR)
8 Bits HA 0 Write-Only

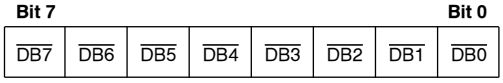
This is a transparent latch used to send data to the SCSI bus. The register can be written by µP cycles or via DMA. DMA writes automatically select the ODR at Hex Address 0 (HA 0). This register is also written with the ID bits required during arbitration and selection/reselection phases. Data is latched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



Output Data Register

CURRENT SCSI DATA (CSD)
8 Bits HA 0 Read-Only

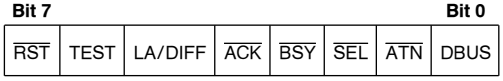
This register enables reading of the current SCSI data bus. If SCSI parity checking is enabled it will be checked at the beginning of the read cycle. The register is also used for µP accesses of SCSI data during programmed-I/O or ID checking during arbitration. Parity is not valid during arbitration.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



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 BSY is lost (see MR2 description).



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 ASI 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/O is false, and C/D, I/O, 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.

ATN: Assert Attention **Bit 1**

0 Deassert ATN.

1 Assert SCSI ATN signal. The MR2 TARGET MODE bit must also be false to assert the signal.

Reading the ICR reflects the state of this bit.

SEL: Assert Select **Bit 2**

0 Deassert SEL.

1 Assert SCSI SEL signal. Can be used in INITIATOR or TARGET mode.

Reading the ICR reflects the state of this bit.

BSY: Assert Busy **Bit 3**

0 Deassert BSY.

1 Assert SCSI BSY signal. Can be used in INITIATOR or TARGET mode.

Reading the ICR reflects the state of this bit.

Hex Adr	Register	Mnemonic	Bits	R/W
0	Output Data Register	ODR	8	WO
0	Current SCSI Data	CSD	8	RO
1	Initator 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 3.2. Registers

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.

1 Do not use. Reserved for future use on a differential pair device.

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 ASI loses arbitation, 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 ASI 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 must be set to enable arbitration.

TEST: Test Mode Enable **Bit 6 Write**

0 Output drivers are enabled.

1 Output drivers disabled.

AIP: Arbitration In Progress **Bit 6 Read**

0 Normally 0 to show no arbitration in progress.

1 Set when the ASI 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 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 HA2 Read-Write

This register is used to program basic operating conditions in the ASI. 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 ASI 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 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 ASI 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) enable 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

3.0 Register Description (Continued)

DMA transfers the SCSI phase lines ($\overline{C/D}$, $\overline{MSG}$, $\overline{I/O}$) 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	$\overline{REQ}$	$\overline{MSG}$	$\overline{C/D}$	$\overline{I/O}$

Target Command Register

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

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

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

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

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

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

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

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

0 Deassert $\overline{MSG}$.

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

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

0 Deassert $\overline{REQ}$.

1 Assert SCSI $\overline{REQ}$ 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 ASI to respond to during Selection or Reselection Phases. Only one bit in the register should be set. When $\overline{SEL}$ is true, $\overline{BSY}$ 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
$\overline{DB7}$	$\overline{DB6}$	$\overline{DB5}$	$\overline{DB4}$	$\overline{DB3}$	$\overline{DB2}$	$\overline{DB1}$	$\overline{DB0}$

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

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 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	$\overline{ATN}$	$\overline{ACK}$

Bus & Status Register

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

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

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

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

BSY: Busy Error **Bit 2**

0 No Error.

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

PHSM: Phase Match **Bit 3**

0 Phase Match. The SCSI $\overline{C/D}$, $\overline{I/O}$ and $\overline{MSG}$ 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 $\overline{DACK}$ 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 $\overline{DACK}$, $\overline{EOP}$ and either $\overline{RD}$ or $\overline{WR}$ 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 $\overline{REQ}$ and $\overline{ACK}$ should be monitored to verify when the last SCSI transfer is complete. This bit is reset when the MR2 DMA bit is reset.

3.0 Register Description (Continued)

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

START DMA TARGET RECEIVE (SDT)
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 ASI DMA logic. A DMA read ($\overline{DACK}$ and $\overline{RD}$) automatically selects this register. Programmed-I/O SCSI data reads should use the CSD (HA8)

START DMA INITIATOR RECEIVE (SDI)
0 Bits HA 7 Write-Only

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

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, Busy Loss and Interrupt Request latches.

4.0 Device Operation

4.1 GENERAL

This section describes overall operation of the ASI. 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 μP's and DMA controllers refer to Section 7.0.

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 A.1.

4.2 μP ACCESSES

The μP accesses the EASI via the $\overline{CS}$, $\overline{RD}$, $\overline{WR}$ and address and data lines in order to read/write the registers. *Figure 4.2* 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 ASI and can be read via the CSB, BSR and CSD registers. An active SCSI reads a 1 in the status registers.

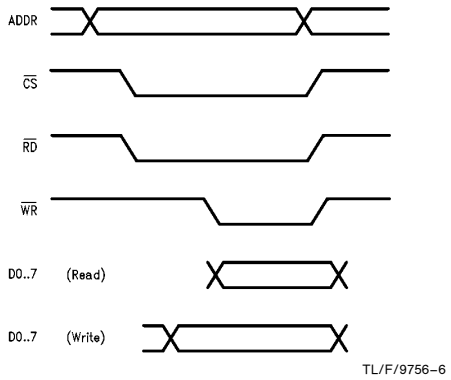


FIGURE 4.2. μP Cycles

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 ASI and how to program it.

Since the SCSI arbitration process requires signal sequencing too fast for μP's, hardware support is provided by the ASI. 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 ASI will monitor the bus for a BUS FREE phase. The $\overline{BSY}$ signal is continuously monitored. If continuously inactive for at least a SCSI Bus Settle Delay (400 ns) and $\overline{SEL}$ is inactive, a valid Bus Free Phase exists. After a period of SCSI Bus Free Delay (800 ns) the ASI asserts $\overline{BSY}$ 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

4.0 Device Operation (Continued)

device asserts $\overline{SEL}$ 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 $\overline{SEL}$ 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.

4.5 SELECTION/RESELECTION

The ASI can be used to select or reselect a device. The ASI will also respond to selection or reselection.

4.5.1 Selecting/Reselecting

Selection requires programming the ODR with the desired and own device ID's; the data bus via ICR DBUS (bit 0); asserting $\overline{ATN}$ if required via ICR bit 1; asserting $\overline{SEL}$ via ICR bit 2; then resetting the MR2 ARB bit.

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

Resetting the ARB bit causes the ASI to remove $\overline{BSY}$ 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 ID's.

$\overline{BSY}$ 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 ASI off the bus. If the device responds the ICR DBUS and $\overline{SEL}$ bits should be reset to remove these signals. If this is a Reselection the ICR $\overline{BSY}$ bit (3) should be set before removing the other signals.

The bus is now ready to handle Information Transfer Phases.

4.5.2 (Re)Selection Response

The ASI responds to Selection or Reselection when the SER is non-zero. A (re)selected interrupt is generated when $\overline{BSY}$ is false for at least a Bus Settle Delay (400 ns); and $\overline{SEL}$ 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 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 so it must be determined by the absence of other interrupts, and the active state of the $\overline{SEL}$ signal. Reselection occurs when $\overline{I/O}$ is also active. See Section 6.0.

4.6 MONITORING BSY

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

This interrupt sets bit 2 in the BSR.

4.7 COMMAND/MESSAGE/STATUS TRANSFERS

Command message and status bytes are transferred using programmed-I/O. The SCSI $\overline{REQ}/\overline{ACK}$ 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 A.1 for flowcharts.

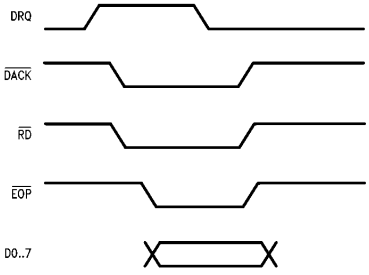
```
Initiator Command Send
{
   MR2 = monitor BSY
   TCR = Command Phase  /*02h*/
   while (bytes) to do) {
      while (REQ) inactive)
        idle; /*CSB bit 5 = 0*/
      if (BSR: phase match == 0)
        phase error;
      else {
        ODR = date byte;
        ICR = Assert ACK;
        while (REQ 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.

4.0 Device Operation (Continued)

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



TL/F/9756–7

FIGURE 4.8.1. Non-Block DMA Timing

4.8.1. NON-BLOCK DMA

DMA operation involves programming the ASI 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 ASI requires enabling or disabling the following: Data bus driving, DMA mode enable, $\overline{BSY}$ monitoring, $\overline{EOP}$ 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 $\overline{EOP}$ 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 7.0).

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

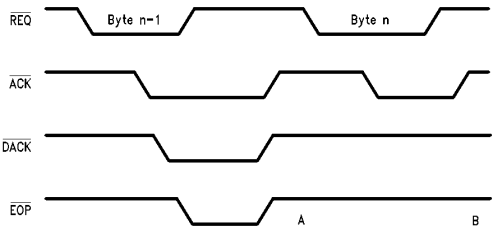
The code below shows programming of the ASI in each of the four DMA cases. One of these cases is shown in a flow diagram in Appendix A.

```
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, so wait until SCSI
     bus inactive before changing phase*/
    MR2 = 40h;
    change phase as required;
    }
}
```

```
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 simply detecting $\overline{REQ}$ and $\overline{ACK}$ simultaneously inactive is not enough.

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



TL/F/9756–8

FIGURE 4.8.2. Target Send DMA

As shown in *Figure 4.8.2* $\overline{ACK}$ going active causes the DRQ for the next byte and also $\overline{REQ}$ to go inactive. $\overline{ACK}$ going inactive allows $\overline{REQ}$ to go active for the next byte. If the INITIATOR is slow removing $\overline{ACK}$ the μP may sample the SCSI bus after the $\overline{EOP}$ interrupt at point A. Here both $\overline{REQ}$ and $\overline{ACK}$ 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 $\overline{REQ}$ and $\overline{ACK}$ and ONLY when both are inactive in three successive samples will the μP be at point B in the figure.

4.9 BLOCK MODE DMA TRANSFERS

In Block Mode the DMA interface uses the DRQ, $\overline{DACK}$, $\overline{EOP}$ and READY lines, DRQ is asserted once at the beginning of transfers and deasserted once $\overline{DACK}$ is received. $\overline{DACK}$ should be asserted continuously for the duration of all the transfer. $\overline{EOP}$ should be asserted during the last DMA byte signal when the next DMA byte transfers. The ASI asserts the READY signal when the next DMA byte should be transferred.

As for non-block mode the End of DMA interrupt is just $\overline{EOP}$, also in block mode receive the ASI does not return READY to an active signal after $\overline{EOP}$. This means external logic must gate off READY if the μP is not to be locked up. For more details see Section 7.0.

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.

When performing DMA as an INITIATOR the $\overline{EOP}$ signal does not deassert $\overline{ACK}$ on the SCSI bus. Firmware must determine when $\overline{REQ}$ is inactive after the last SCSI transfer then reset the MR2 DMA bit to deassert $\overline{ACK}$.

Programming the ASI 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 ASI DMA logic for non-data transfers. This removes the need to poll $\overline{REQ}$/$\overline{ACK}$ and program the assertion/deassertion of the handshake signal. The μP can emulate a DMA controller by asserting $\overline{DACK}$ and $\overline{EOP}$ 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 7.0 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: $\overline{EOP}$, phase mismatch and resetting the DMA MODE bit in MR2.

4.11.1 End Of Process

$\overline{EOP}$ is asserted for a minimum period during the last DMA cycle. The $\overline{EOP}$ signal generates the End of DMA interrupt. $\overline{EOP}$ does not cause the MR2 DMA mode bit to be reset.

4.11.2 DMA Phase Mismatch

If a $\overline{REQ}$ goes active while there is a phase mismatch the DMA will be halted and an interrupt generated. The ASI 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 $\overline{EOP}$ 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.

4.0 Device Operation (Continued)

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

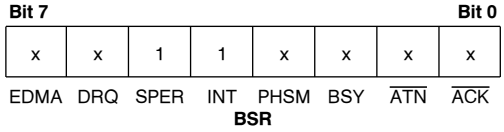
Before individually describing each interrupt an explanation of the use of interrupts is required.

5.2 USING INTERRUPTS

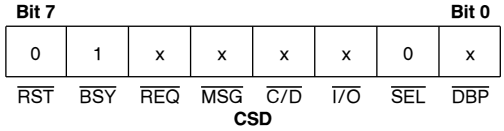
Interrupts are controlled by bits in MR2 if control is provided. Not all interrupts can be disabled under software control. When an interrupt occurs both the BSR and CSD register must be read and analysed to determine the source of interrupt. Since status is NOT provided for each interrupt great care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

5.3 SCSI PARITY ERROR

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 HA7. Following an interrupt the BSR and CSD should contain the values shown below.



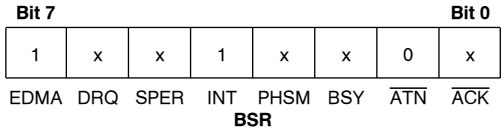
BSR



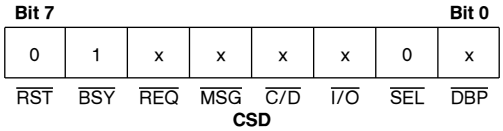
CSD

5.4 END OF DMA

If $\overline{EOP}$ 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. $\overline{EOP}$ is recognized when $\overline{EOP}$, $\overline{DACK}$ and either $\overline{IOR}$ or $\overline{IOW}$ are all simultaneously active for a minimum period. The interrupt may be reset by reading HA 7. Following an interrupt the BSR and CSD should contain the values shown below.



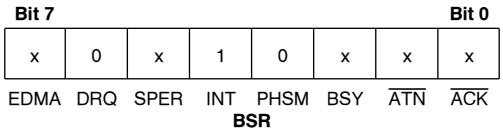
BSR



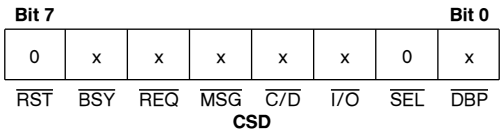
CSD

5.5 DMA PHASE MISMATCH

When the SCSI $\overline{REQ}$ goes active during a DMA operation the contents of the TCR are compared with the SCSI phase lines $\overline{C/D}$, $\overline{MSG}$ and $\overline{I/O}$. 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. The mismatch removes the ASI from driving the SCSI data bus. The interrupt may reset by reading HA 7. Following an interrupt the BSR and CSD should contain the values shown below.



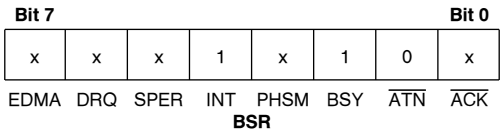
BSR



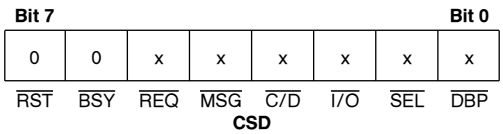
CSD

5.6 BUSY LOSS

If bit 2 MR2 is set the SCSI $\overline{BSY}$ signal is monitored and an interrupt is generated if $\overline{BSY}$ 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 CSD should contain the values shown below, where usually CSD = 00.



BSR



CSD

5.7 (RE)SELECTION

An interrupt will be generated when: $\overline{SEL}$ is active, $\overline{BSY}$ 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 $\overline{I/O}$ is active this is a reselection. The interrupt is disabled by writing all zeros to the SER, and reset by reading HA 7.

5.0 Interrupts (Continued)

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

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

BSR

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

CSD

6.0 Reset Conditions

6.1 GENERAL

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

6.2 CHIP RESET

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

6.3 EXTERNAL SCSI RESET

When a SCSI $\overline{RST}$ is applied externally the ASI 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.

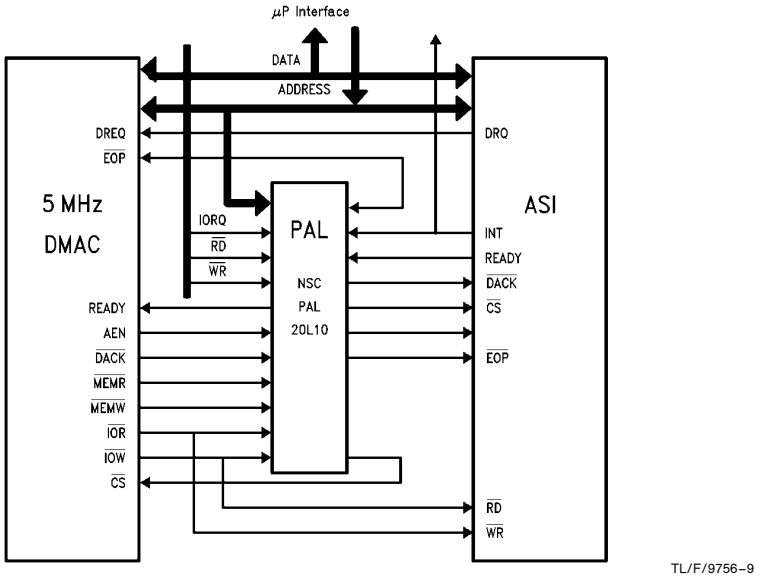
6.4 SCSI RESET ISSUED

When the μP sets the Assert RST bit in the ICR the $\overline{RST}$ signal goes active. Since the ASI monitors $\overline{RST}$ also the same reset actions as in 6.3 apply. The SCSI $\overline{RST}$ signal will remain active as long as bit 7 in the ICR is set—i.e. until programmed 0 or a chip $\overline{RESET}$ occurs.

7.0 Application Guide

This section is intended to show the interface between the μP, ASI and DMA controller (DMAC). *Figure 7.1* shows a general interface when the ASI and DMAC are I/O-mapped devices. This configuration will implement a 2 to 2.5M Bytes/sec SCSI port using 2 cycle compressed timing from the 5 MHz DMAC.

Using a faster DMAC and memory may allow the ASI 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.



TL/F/9756–9

FIGURE 7.1. μP/ASI/DMA Interface

7.0 Application Guide (Continued)

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

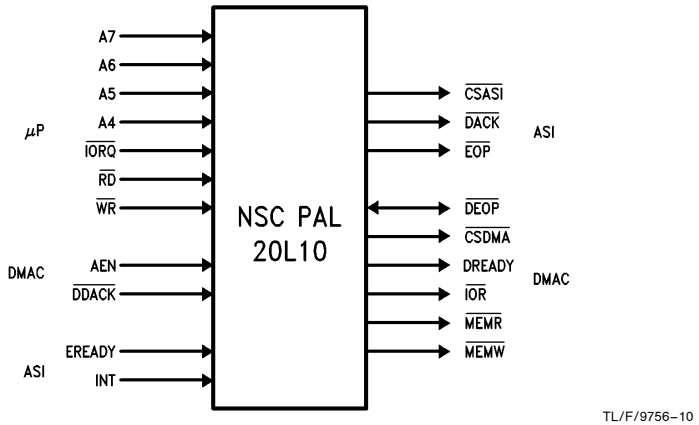
The PAL is programmed so that the µP can access the ASI in three ways. The three access types are: Register R/W, DMA R/W, DMA with $\overline{EOP}$. Examination of the PAL equations below shows how the µP may perform any of the three basic access types simply by accessing the ASI at different I/O address 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 ASI 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 ASI 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 $\overline{RD}$ and $\overline{WR}$ 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 7.2* with the signal definitions and equations following.



TL/F/9756–10

FIGURE 7.2. Interface PAL

7.0 Application Guide (Continued)

```
CSASI = IORQ*A7*A6*A5*A4*AEN; ASI reg R/W chip select
ADACK = 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
IF(AEN)AEOP = IORQ*A7*A6*A5*A4*RD; μP pseudo-DMA with EOP
              +IORQ*A7*A6*A5*A4*WR + DEOP* AREADY
IF(DDACK*AREADY*INT)DEOP = DDACK*AREADY*INT
                           ;DMA cycle with error
CSDMA = IORQ*A7*A6*A5*A4    ;DMAC register R/W
DREADY = AREADY*INT         ;ASI not READY and not INT
         +AREADY*DDACK      ;ASI 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
IF(AEN)MEMR = IORQ*RD       ;μP memory Read cycle
IF(AEN)MEMW = IORQ*WR       ;μP memory Write cycle
```

FIGURE 7.3. PAL Equations

The μP and DMA signals are defined below

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

8.0 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 sec)	260°C
Electro-Static Discharge Rating	2 kV

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

9.0 DC Electrical Characteristics (V_{CC} = 5.0V ± 10% unless otherwise specified) T_A = 0°C to +70°C

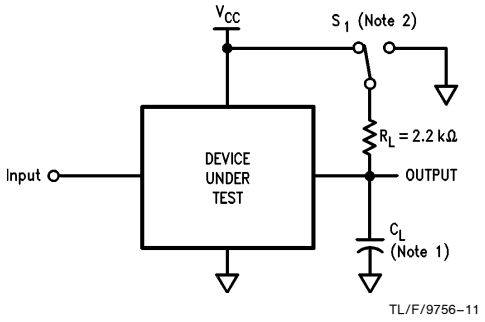
Symbol	Parameter	Conditions	Typ	Limit	Units
V_{IH}	Minimum High Level Input Voltage			2.0	V
V_{IL}	Maximum Low Level Input Voltage			0.8	V
V_{OH1} V_{OH2}	Minimum High Level Output Voltage	$\|I_{OUT}\|$ = 20 μA $\|I_{OUT}\|$ = 4.0 mA		V_{CC} − 0.1 2.4	V V
V_{OL1} V_{OL2} V_{OL3}	Maximum Low Level Output Voltage	SCSI Bus Pins: $\|I_{OL}\|$ = 48 mA Other Pins: $\|I_{OL}\|$ = 20 μA $\|I_{OL}\|$ = 8.0 mA		0.5 0.1 0.4	V V 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 Level	GND to 3.0V
Input Rise and Fall Times	6 ns
Input/Output Reference Levels	1.3V
TRI-STATE Reference Levels (Note 2)	Active Low + 0.5V Active High − 0.5V



TL/F/9756–11

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

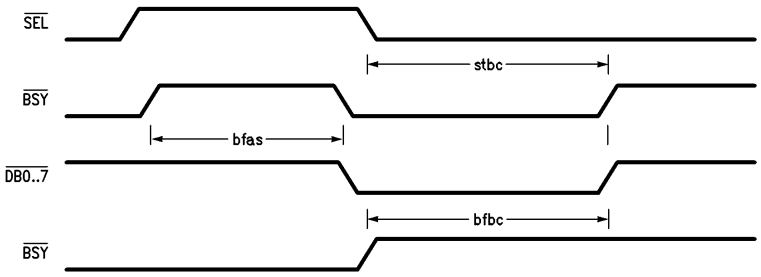
Note 2: S1 = Open for push-pull outputs.
S1 = V_{CC} for active low to TRI-STATE.
S1 = GND for active high to TRI-STATE.

Note 3: This parameter is not 100% tested.

10.0 AC Electrical Characteristics all parameters are preliminary and subject to change without notice

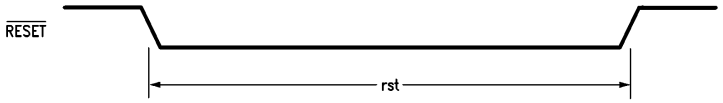
Symbol	Parameter	DP5380			Units
		Min	Typ	Max	
bfas	BSY False to Arbitrate Start	1200		2200	ns
bfbc	BSY False to Bus Clear			800	ns
rst	RESET Pulse Width	150			ns
stbc	SEL True to Bus Clear			500	ns

10.1 ARBITRATION



TL/F/9756–12

10.2 μP RESET



TL/F/9756–13

10.0 AC Electrical Characteristics

all parameters are preliminary and subject to change without notice (Continued)

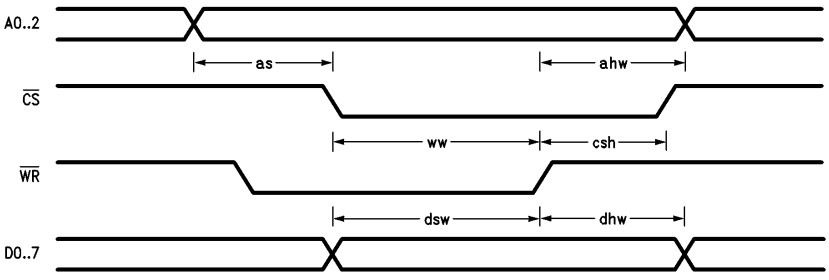
Symbol	Parameter	DP5380			Units
		Min	Typ	Max	
ahr	Address Hold from End of Read Enable (Note 1)	10			ns
ahw	Address Hold from End of Write Enable (Note 2)	10			ns
as	Address Setup to Read or Write Enable (Notes 1, 2)	10			ns
csh	CS Hold from End of RD or WR	0			ns
dhr	Data Hold from End of Read Enable (Notes 1, 3)	10		60	ns
dhw	μP Data Hold Time from End of WR	20			ns
dsw	Data Setup to End of Write Enable	50			ns
rdv	Data Valid from Read Enable (Note 1)			100	ns
ww	Write Enable Width (Note 2)	60			ns

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

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

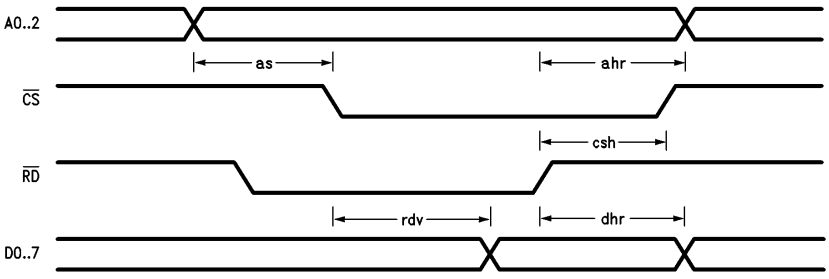
Note 3: This includes the RC delay inherent in the tests' method. These signals typically turn off after 25 ns enabling other devices to drive these lines with no contention.

10.3 μP WRITE



TL/F/9756–14

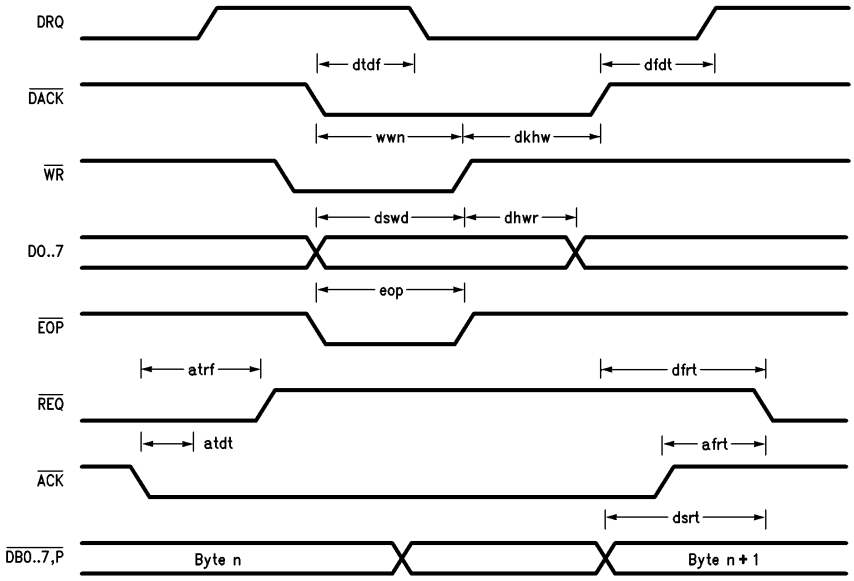
10.4 μP READ



TL/F/9756–15

10.0 AC Electrical Characteristics (Continued)

10.5 DMA WRITE (NON-BLOCK MODE) TARGET SEND



TL/F/9756–16

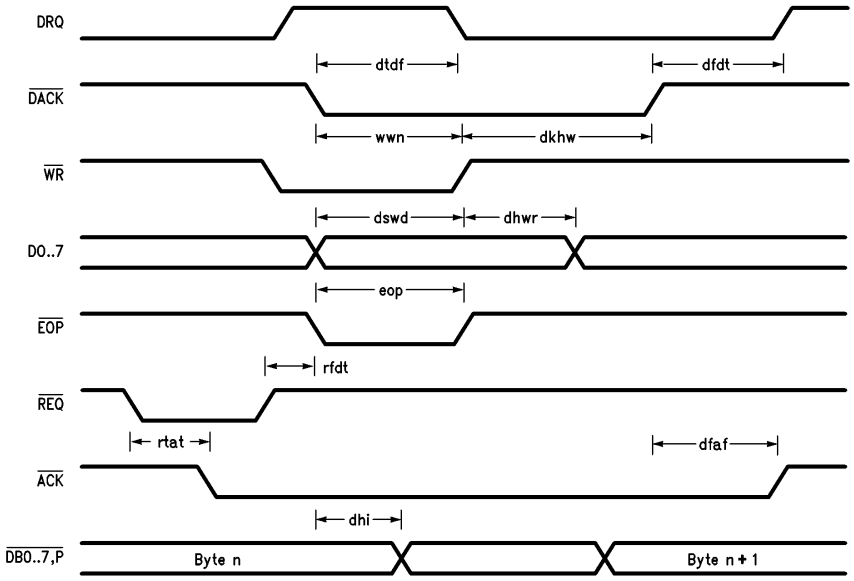
Symbol	Parameter	DP5380			Units
		Min	Typ	Max	
afrt	ACK False to REQ True (DACK or WR False)			120	ns
atdt	ACK True to DRQ True (Target)			90	ns
atrf	ACK True to REQ False (Target)			115	ns
dfdt	DACK False to DRQ True	30	90		ns
dfrt	DACK False to REQ True (ACK False)			110	ns
dhwr	DMA Data Hold Time from End of WR	30			ns
dkhw	DACK Hold from End of WR	0			ns
dsrt	SCSI Data Setup to REQ True (Target Send) (Note 1)	40			ns
dswd	Data Setup to End of DMA Write Enable	50			ns
dtdf	DACK True to DRQ False			100	ns
eop	Width of EOP Pulse (Note 2)	40			ns
wwn	DMA Non-Block Mode Write Enable Width (Note 3)	60			ns

Note 1: EOP, DACK, RD/WR must all be true for recognition of EOP.

Note 2: Write enable (DMA) is DACK and WR active.

10.0 AC Electrical Characteristics (Continued)

10.6 DMA WRITE (NON-BLOCK MODE) INITIATOR SEND



TL/F/9756–17

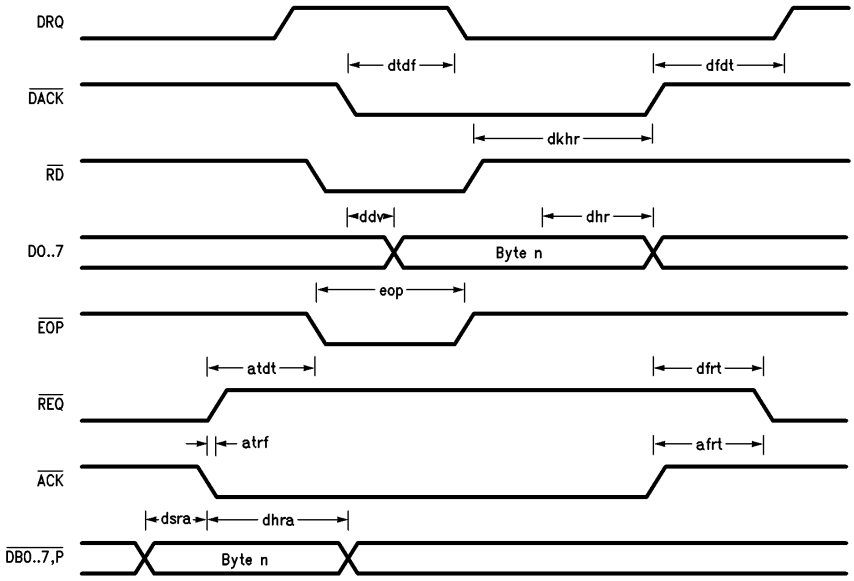
Symbol	Parameter	DP5380			Units
		Min	Typ	Max	
dfaf	DACK False to ACK False (Non-Block Initiator Send)			120	ns
dfdt	DACK False to DRQ True	30	90		ns
dhi	SCSI Data Hold from Write Enable—Initiator	15			ns
dhwr	DMA Data Hold Time from End of WR	30			ns
dkhw	DACK Hold from End of WR	0			ns
dswd	Data Setup to End of DMA Write Enable	50			ns
dtdf	DACK True to DRQ False			100	ns
eop	Width of EOP Pulse (Note 1)	40			ns
rfdt	REQ False to DRQ True			100	ns
rtat	REQ True to ACK True (Initiator Send)			100	ns
wwn	DMA Non-Block Mode Write Enable Width (Note 2)	60			ns

Note 1: EOP, DACK, RD/WR must all be true for recognition of EOP.

Note 2: Write enable (DMA) is DACK and WR active.

10.0 AC Electrical Characteristics (Continued)

10.7 DMA READ (NON-BLOCK MODE) TARGET RECEIVE



TL/F/9756–18

Symbol	Parameter	DP5380			Units
		Min	Typ	Max	
afrt	ACK False to REQ True (DACK or WR False)			120	ns
atdt	ACK True to DRQ True (Target)			90	ns
atrf	ACK True to REQ False (Target)			115	ns
ddv	DMA Data Valid from Read Enable (Note 1)			90	ns
dfdt	DACK False to DRQ True	30	90		ns
dfrt	DACK False to REQ True (ACK False)			110	ns
dhr	Data Hold from End of Read Enable (Notes 1, 2)	10		60	ns
dhra	SCSI Data Hold from REQ or ACK True (Receive)	30			ns
dkhr	DACK Hold from End of RD	0			ns
dsra	SCSI Data Setup Time to REQ or ACK True (Receive)	20			ns
dtdf	DACK True to DRQ False			100	ns
eop	Width of EOP Pulse (Note 4)	40			ns

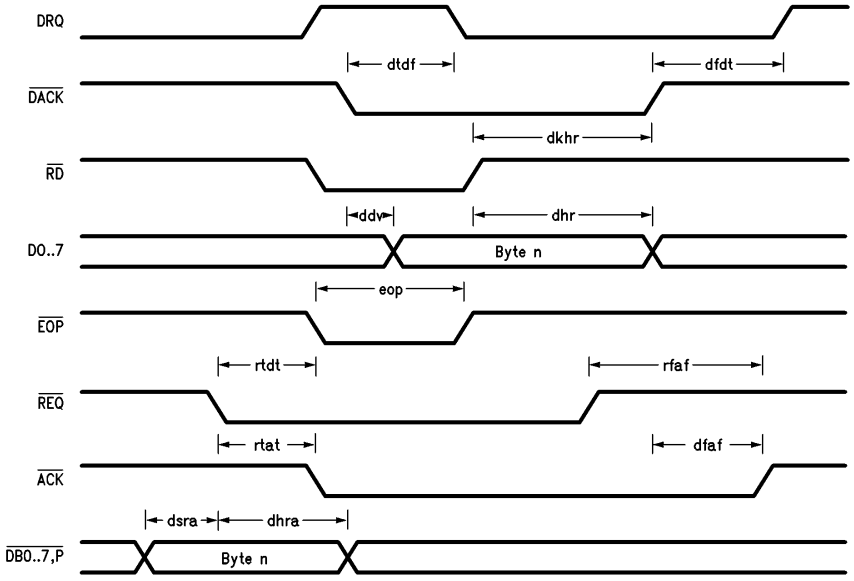
Note 1: Read enable (DMA) is DACK and RD active.

Note 2: This includes the RC delay inherent in the tests' 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.

10.0 AC Electrical Characteristics (Continued)

10.8 DMA READ (NON-BLOCK MODE) INITIATOR RECEIVE



TL/F/9756–19

Symbol	Parameter	DP5380			Units
		Min	Typ	Max	
ddv	DMA Data Valid from Read Enable (Note 1)			90	ns
dfaf	DACK False to ACK False (REQ False, Non-block, In rx)			120	ns
dfdt	DACK False to DRQ True	30	90		ns
dhr	Data Hold from End of Read Enable (Notes 1, 2)	10		60	ns
dhra	SCSI Data Hold from REQ or ACK True (Receive)	30			ns
dkhr	DACK Hold from End of RD	0			ns
dsra	SCSI Data Setup Time to REQ or ACK True (Receive)	20			ns
dtdf	DACK True to DRQ False			100	ns
eop	Width of EOP Pulse (Note 3)	40			ns
rfaf	REQ False to ACK False (DACK False)			100	ns
rtat	REQ True to ACK True (Initiator Receive)			100	ns
rtdt	REQ True to DRQ True			120	ns

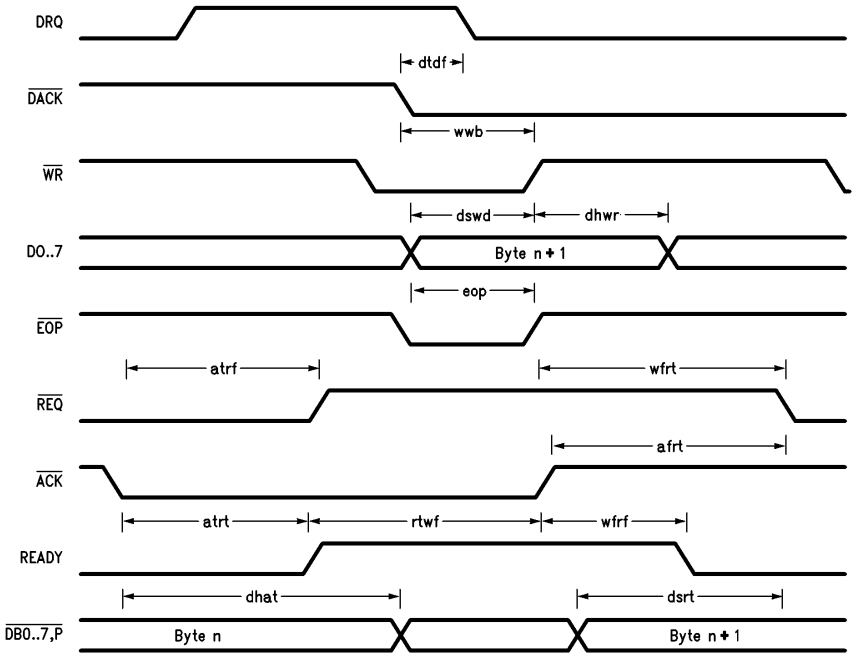
Note 1: Read enable (DMA) is DACK and RD active.

Note 2: This includes the RC delay inherent in the tests' 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.

10.0 AC Electrical Characteristics (Continued)

10.9 DMA WRITE (BLOCK MODE) TARGET SEND



TL/F/9756–20

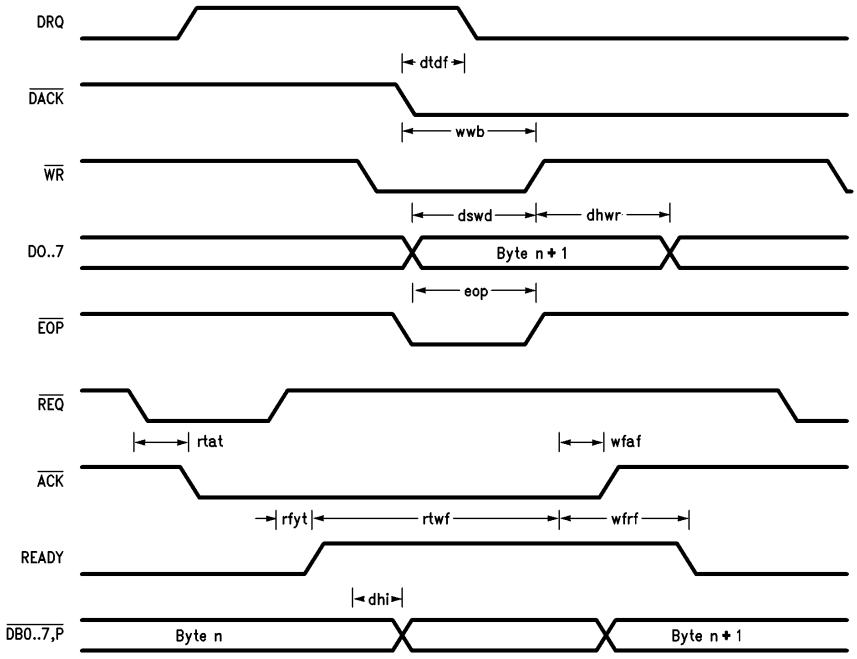
Symbol	Parameter	DP5380			Units
		Min	Typ	Max	
afrt	ACK False to REQ True (DACK or WR False)			120	ns
atrf	ACK True to REQ False (Target)			115	ns
atrt	ACK True to READY True (Block Mode Target Send)			110	ns
dhat	SCSI Data Hold from ACK True	40			ns
dhwr	DMA Data Hold Time from End of WR	30			ns
dsrt	SCSI Data Setup to REQ True	50			ns
dswd	Data Setup to End of DMA Write Enable	50			ns
dtdf	DACK True to DRQ False			100	ns
eop	Width of EOP Pulse (Note 1)	40			ns
rtwf	READY true to WR False	60			ns
wfrf	WR False to READY False			100	ns
wfrt	WR False to REQ True (ACK False)			120	ns
wwb	DMA Write Enable Width (Note 2)	60			ns

Note 1: EOP, DACK, RD, WR must all be true for recognition of EOP.

Note 2: Write enable (DMA) is DACK and WR active.

10.0 AC Electrical Characteristics (Continued)

10.10 DMA WRITE (BLOCK MODE) INITIATOR SEND



TL/F/9756–21

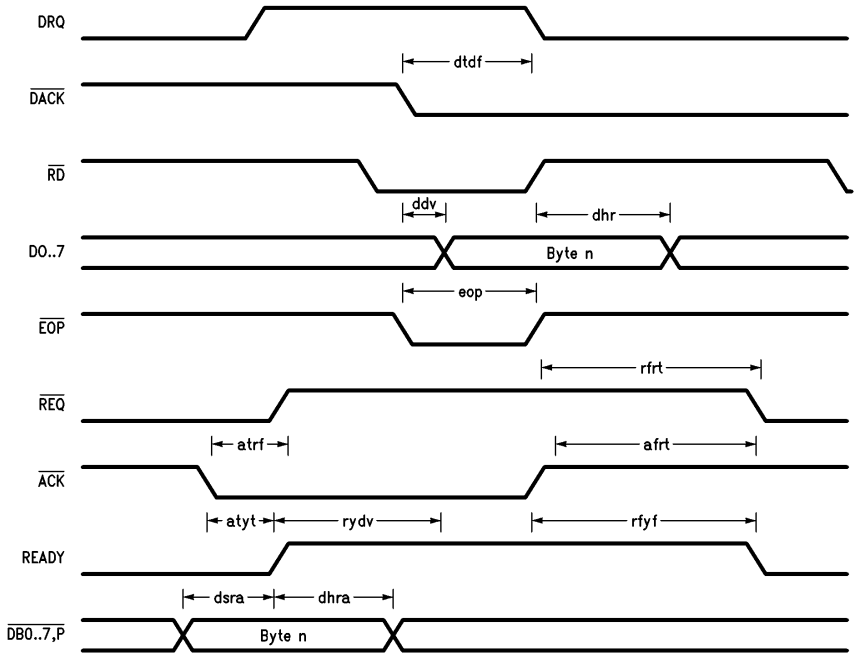
Symbol	Parameter	DP5380			Units
		Min	Typ	Max	
dhi	SCSI Data Hold from Write Enable—Initiator	15			ns
dhwr	DMA Data Hold Time from End of WR	30			ns
dswd	Data Setup to End of DMA Write Enable	50			ns
dtdf	DACK True to DRQ False			100	ns
eop	Width of EOP Pulse (Note 2)	40			ns
rfyt	REQ False to READY True			80	ns
rtat	REQ True to ACK True			100	ns
rtwf	READY True to WR False	60			ns
wfaf	WR False to ACK False (REQ False)			120	ns
wfrf	WR False to READY False			100	ns
wwb	DMA Write Enable Width (Note 1)	60			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.

10.0 AC Electrical Characteristics (Continued)

10.11 DMA READ (BLOCK MODE) TARGET RECEIVE



TL/F/9756–22

Symbol	Parameter	DP5380			Units
		Min	Typ	Max	
afrt	ACK False to REQ True (DACK or WR False)			120	ns
atrf	ACK True to REQ False			115	ns
atyt	ACK True to READY True			110	ns
ddv	DMA Data Valid from Read Enable (Note 1)			90	ns
dhr	Data Hold from End of Read Enable (Notes 1, 2)	10		60	ns
dhra	SCSI Data Hold from REQ or ACK True	30			ns
dsra	SCSI Data Setup Time to REQ or ACK True	20			ns
dtdf	DACK True to DRQ False			100	ns
eop	Width of EOP Pulse (Note 3)	40			ns
rfrt	RD False to REQ True (ACK False)			100	ns
rfyf	RD False to READY False			110	ns
rydv	READY True to Data Valid			35	ns

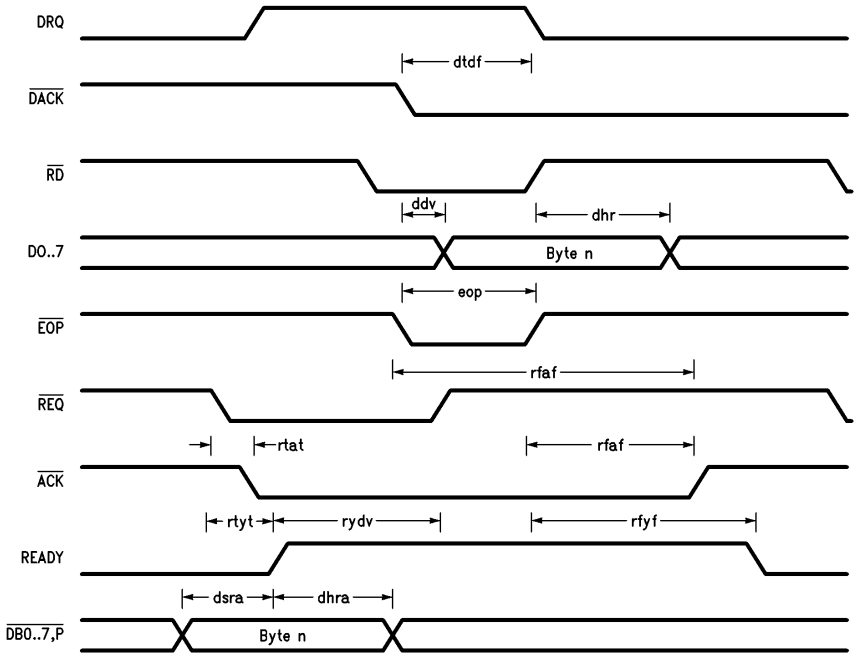
Note 1: Read enable (DMA) is DACK and RD active.

Note 2: This includes the RC delay inherent in the tests' 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.

10.0 AC Electrical Characteristics (Continued)

10.12 DMA READ (BLOCK MODE) INITIATOR RECEIVE



TL/F/9756–23

Symbol	Parameter	DP5380			Units
		Min	Typ	Max	
ddv	DMA Data Valid from Read Enable (Note 1)			90	ns
dhr	Data Hold from End of Read Enable (Notes 1, 2)	10		60	ns
dhra	SCSI Data Hold from REQ or ACK True	30			ns
dsra	SCSI Data Setup Time to REQ or ACK True	20			ns
dtdf	DACK True to DRQ False			100	ns
eop	Width of EOP Pulse (Note 3)	40			ns
rdaf	RD False to ACK False (REQ False)			125	ns
rfaf	REQ False to ACK False (DACK False)			100	ns
rfyf	RD False to READY True False			110	ns
rtat2	REQ True to ACK True			100	ns
rtyt	REQ True to READY True			75	ns
rydv	READY True to Data Valid			35	ns

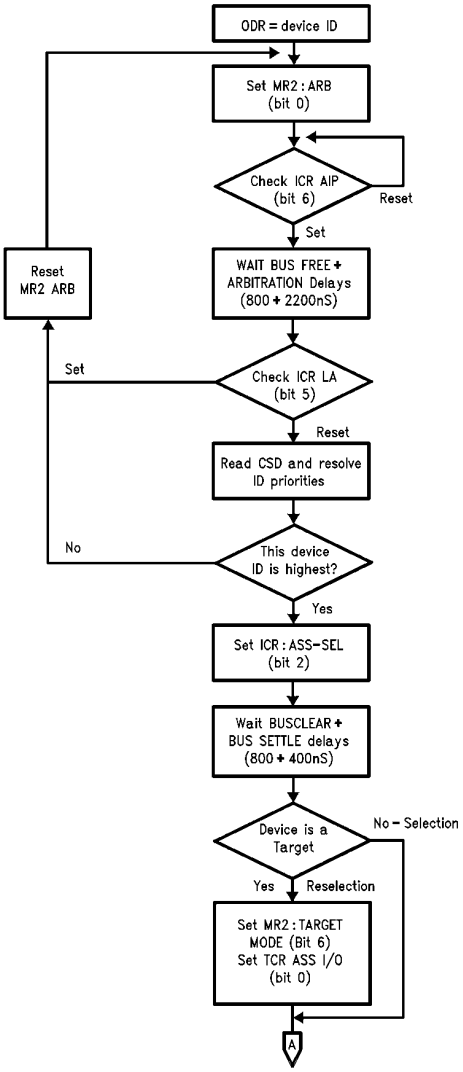
Note 1: Read enable (DMA) is DACK and RD active.

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

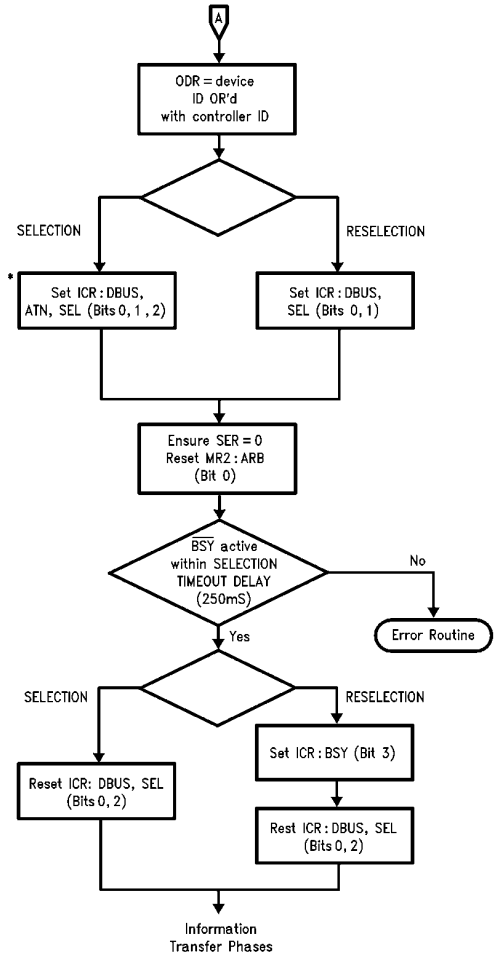
Note 3: EOP, DACK, RD/WR must all be active for recognition of EOP.

Appendix A1

Arbitration and (Re)Selection



TL/F/9756–24

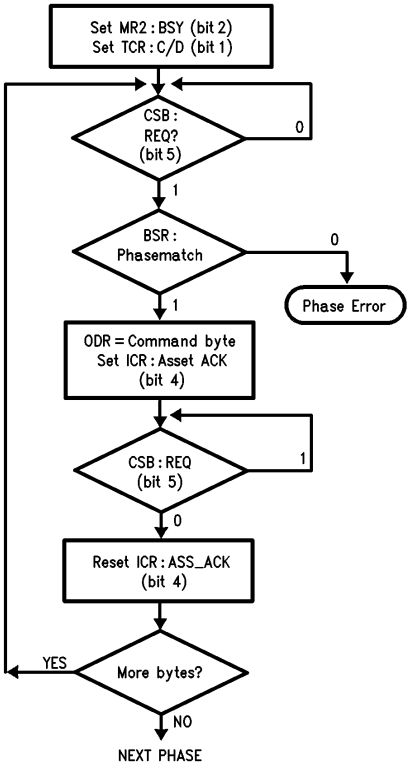


TL/F/9756–25

*Only set ATN if Select with ATN is desired.

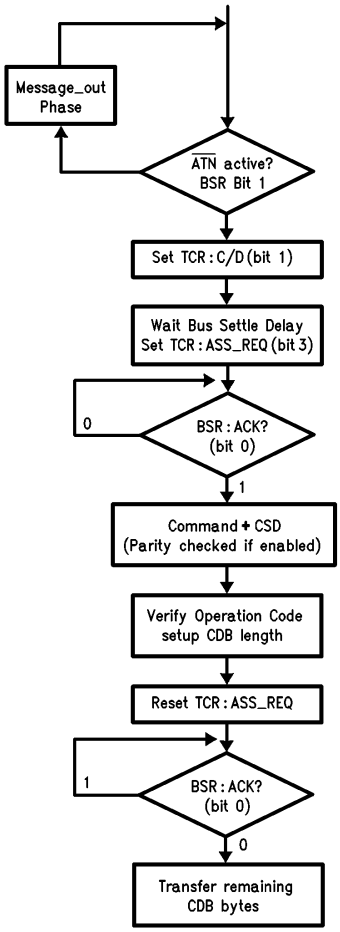
Appendix A1 (Continued)

Command Transfer (Initiator)



TL/F/9756–26

Command Transfer (Target)



TL/F/9756–27

Appendix A2

Register Chart

READ

Bit 7 — Current SCSI Data (CSD) — Bit 0

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

Bit 7 — Initiator Command Register (ICR) — Bit 0

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

Bit 7 — Mode Register 2 (MR2) — Bit 0

BLK	TARG	PCHK	PINT	EOP	BSY	DMA	ARB

Bit 7 — Target Command Register (TCR) — Bit 0

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

Bit 7 — Current SCSI Bus Status (CSB) — Bit 0

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

Bit 7 — Bus and Status Register (BSR) — Bit 0

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

Bit 7 — Input Data Register (IDR) — Bit 0

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

Bit 7 — Reset Parity/Interrupt (RPI)—Mode N — Bit 0

x	x	x	x	x	x	x	x

X = Unknown

WRITE

Bit 7 — Output Data Register (ODR) — Bit 0

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

Bit 7 — Initiator Command Register (ICR) — Bit 0

$\overline{RST}$	TEST	DIFF EN	$\overline{ACK}$	$\overline{BSY}$	$\overline{SEL}$	$\overline{ATN}$	$\overline{DBUS}$

Bit 7 — Mode Register 2 (MR2) — Bit 0

BLK	TARG	PCHK	PINT	EOP	BSY	DMA	ARB

Bit 7 — Target Command Register (TCR) — Bit 0

x	x	x	x	$\overline{REQ}$	$\overline{MSG}$	$\overline{C/D}$	$\overline{I/O}$

Bit 7 — Select Enable Register (SER) — Bit 0

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

Bit 7 — Start DMA Send (SDS) — Bit 0

x	x	x	x	x	x	x	x

Bit 7 — Start DMA Target Receive (SDT) — Bit 0

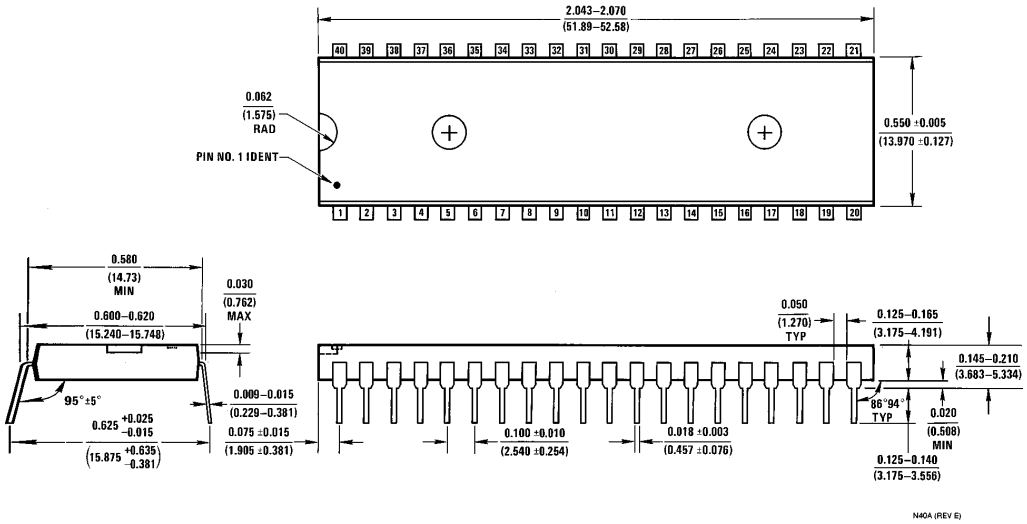
x	x	x	x	x	x	x	x

Bit 7 — Start DMA Initiator Receive (SDI)—Mode N — Bit 0

x	x	x	x	x	x	x	x

X = Don't Care

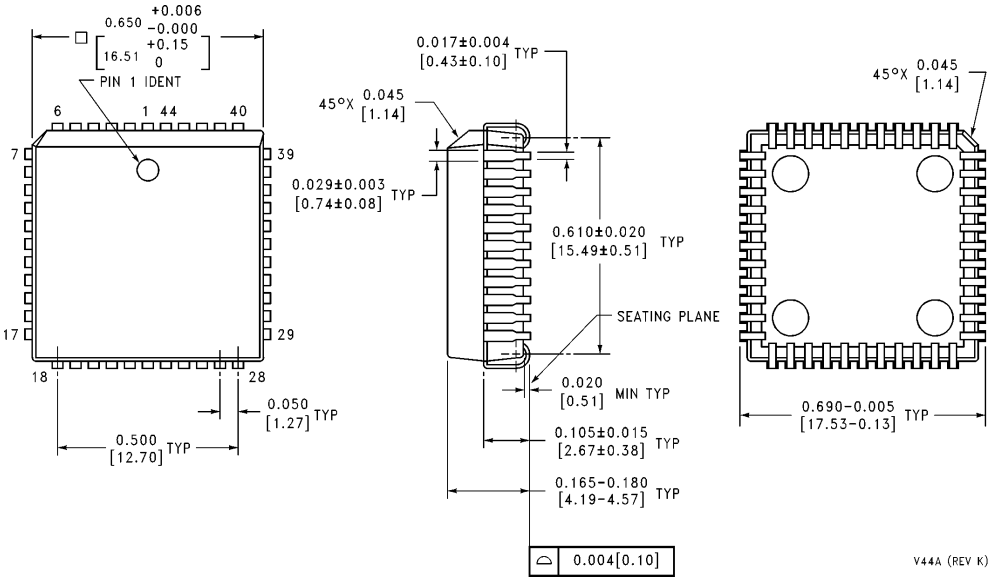
Physical Dimensions inches (millimeters)



Molded Dual-In-Line Package (N)
Order Number DP5380N
NS Package Number N40A

Physical Dimensions inches (millimeters) (Continued)

Lit. # 102926



Plastic Chip Carrier (V)
Order Number DP5380V
NS Package Number V44A

LIFE SUPPORT POLICY

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

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

National Semiconductor Corporation
1111 West Bardin Road
Arlington, TX 76017
Tel: 1(800) 272-9959
Fax: 1(800) 737-7018

National Semiconductor Europe
Fax: (+49) 0-180-530 85 86
Email: cnjwge@tevm2.nsc.com
Deutsch Tel: (+49) 0-180-530 85 85
English Tel: (+49) 0-180-532 78 32
Français Tel: (+49) 0-180-532 93 58
Italiano Tel: (+49) 0-180-534 16 80

National Semiconductor Hong Kong Ltd.
13th Floor, Straight Block,
Ocean Centre, 5 Canton Rd.
Tsimshatsui, Kowloon
Hong Kong
Tel: (852) 2737-1600
Fax: (852) 2736-9960

National Semiconductor Japan Ltd.
Tel: 81-043-299-2309
Fax: 81-043-299-2408