CALMOS™

CA53C80

SCSI INTERFACE CONTROLLER

- **Pin and functional compatibility with the industry standard 53C80**
- **TTL input/output compatibility**
- **Low power CMOS implementation**

SCSI Interface

- **Asynchronous operation with data transfer rates of 1.5 MBytes/sec and 3.0 MBytes/sec**
- **Supports Initiator and Target roles**
- **Parity generation with optional checking**
- **Supports arbitration, selection/reselection**
- **Direct control of all bus signals**
- **High current outputs drive SCSI Bus directly**

CPU Interface

- **Memory or I/O mapped interface**
- **DMA or programmed I/O**
- **Normal or Block mode DMA**
- **Optional CPU interrupts**

The CA53C80 is a high performance Small Computer System Interface (SCSI) Controller chip. Designed to meet the standards defined by ANSI X3T9.2, it operates in both the Initiator and Target roles and can be used in host adapter, host port and formatter designs. The CA53C80 also supports arbitration, selection/reselection, and optional parity checking.

The CA53C80 is designed to communicate with the system microprocessor as a peripheral device. It is controlled by reading and writing several internal registers which may be addressed by standard or memory-mapped I/O. Data transfers can also be handled by normal DMA mode or block DMA mode for DMA controllers such as the CA82C37A which offer that feature.

On-chip, open-collector output drivers which can sink 48mA at 0.5V allow the CA53C80 to directly drive the SCSI bus. Additional ground pins reduce ground bounce.

The low power consumption of the CA53C80 makes it useful in portable and low power standby systems. Its high performance and ground bounce resilience are particularly desirable for defence or industrial applications.

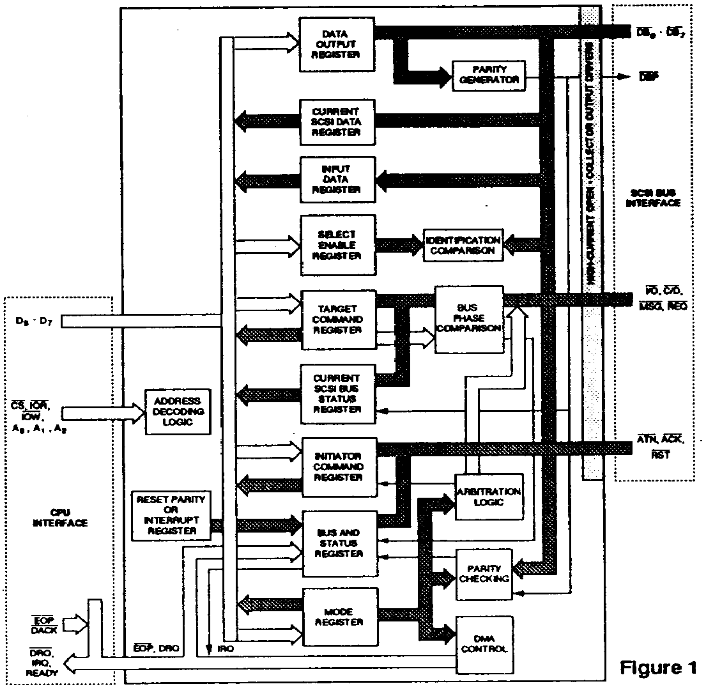
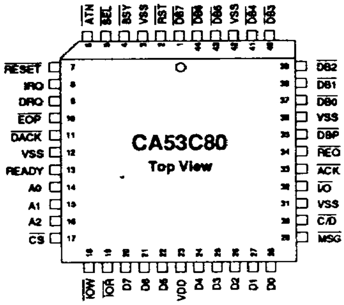

DATA OUTPUT REGISTER
PARITY GENERATOR
CURRENT SCSI DATA REGISTER
INPUT DATA REGISTER
SELECT ENABLE REGISTER
IDENTIFICATION COMPARISON
TARGET COMMAND REGISTER
BUS PHASE COMPARISON
CURRENT SCSI BUS STATUS REGISTER
INITIATOR COMMAND REGISTER
ARBITRATION LOGIC
BUS AND STATUS REGISTER
PARITY CHECKING
MODE REGISTER
DMA CONTROL
ADDRESS DECODING LOGIC
RESET PARITY OR INTERRUPT REGISTER
HIGH-CURRENT OPEN - COLLECTOR OUTPUT DRIVERS
$\overline{DB_0}$ - $\overline{DB_7}$
$\overline{DBP}$
SCSI BUS INTERFACE
$\overline{I/O}$, $\overline{C/D}$, $\overline{MSG}$, $\overline{REQ}$
$\overline{ATN}$, $\overline{ACK}$, $\overline{RST}$
D_0 - D_7
$\overline{CS}$, $\overline{IOR}$, $\overline{IOW}$, A_0, A_1, A_2
CPU INTERFACE
$\overline{EOP}$ $\overline{DACK}$
DRQ, IRQ, READY
$\overline{EOP}$, DRQ
IRQ

Figure 1 : CA53C80 BLOCK DIAGRAM



44 - LEAD PLCC

Figure 2 : PIN CONFIGURATION

Table 1 : PIN DESCRIPTIONS

Symbol	PLCC	Type	Name and Function
A_0 - A_2	14-16	I	**Address Lines:** Used with $\overline{\text{CS}}$, $\overline{\text{IOR}}$ or $\overline{\text{IOW}}$ to address all internal registers.
$\overline{\text{ACK}}$	33	I/O	**Acknowledge:** Driven by an Initiator, $\overline{\text{ACK}}$ indicates an acknowledgement for a $\overline{\text{REQ}}$/ $\overline{\text{ACK}}$ data-transfer handshake. In the Target role, $\overline{\text{ACK}}$ is received as a response to the $\overline{\text{REQ}}$ signal.
$\overline{\text{ATN}}$	6	I/O	**Attention:** Driven by an Initiator, $\overline{\text{ATN}}$ indicates an Attention condition. This signal is received in the Target role.
$\overline{\text{BSY}}$	4	I/O	**Busy:** This signal indicates that the SCSI Bus is being used. It is driven by both Initiator and Target devices.
$\overline{\text{C/D}}$	30	I/O	**Control/Data:** A signal driven by the Target device, $\overline{\text{C/D}}$ indicates that Control or Data information is on the Data Bus. This signal is received by the Initiator.
$\overline{\text{CS}}$	17	I	**Chip Select:** Enables a read or write of the internal register selected by A_0 - A_2
D_0 - D_7	28-24, 22-20	I/O	**Data Lines:** Bidirectional microprocessor data bus lines.
$\overline{\text{DACK}}$	11	I	**DMA Acknowledge:** $\overline{\text{DACK}}$ resets DRQ and selects the data register for input or output data transfers.
$\overline{\text{DB}_0}$ - $\overline{\text{DB}_7}$, $\overline{\text{DBP}}$	37-41, 43, 44, 1, 35	I/O	**Data Bits, Parity Bit:** These eight data bits ($\overline{\text{DB}_0}$ - $\overline{\text{DB}_7}$), plus a parity bit (DBP) form form the Data Bus. $\overline{\text{DB}_7}$ is the most significant bit (MSB) and has the highest priority during the Arbitration phase. Data parity is odd. Parity is always generated and optionally checked. Parity is not valid during Arbitration.
DRQ	9	O	**DMA Request:** DRQ indicates that the data register may be read or written. DRQ occurs only if DMA MODE bit is set in Command register. DRQ is cleared by $\overline{\text{DACK}}$.
$\overline{\text{EOP}}$	10	I	**End of Process:** $\overline{\text{EOP}}$ is used to terminate a DMA transfer. If asserted during a DMA cycle, the current byte is transferred, but no additional bytes are requested.
$\overline{\text{I/O}}$	32	I/O	**Input/Output:** $\overline{\text{I/O}}$ is a Target driven signal which controls the direction of data movement on the SCSI Bus. TRUE indicates input to the Initiator. This signal is also used to distinguish between Selection and Reselection phases.
$\overline{\text{IOR}}$	19	I	**I/O Read:** $\overline{\text{IOR}}$ is used to read an internal register selected by $\overline{\text{CS}}$ and A_0 - A_2. It also selects the Input Data Register when used with $\overline{\text{DACK}}$.
$\overline{\text{IOW}}$	18	I	**I/O Write:** $\overline{\text{IOW}}$ is used to write an internal register selected by $\overline{\text{CS}}$ and A_0 - A_2. It also selects the Output Data Register when used with $\overline{\text{DACK}}$.
IRQ	8	O	**Interrupt Request:** IRQ alerts the microprocessor to an error condition or the completion of an event.
$\overline{\text{MSG}}$	29	I/O	**Message:** $\overline{\text{MSG}}$ is a signal driven by the Target during the Message phase. This signal is received by the Initiator.
READY	13	O	**Ready:** READY is used to control the speed of Block mode DMA transfers. This signal goes *high* to indicate the chip is ready to send/receive data and remains *low* after a transfer until the last byte is sent or until the DMA MODE bit is reset.
$\overline{\text{REQ}}$	34	I/O	**Request:** Driven by a Target, $\overline{\text{REQ}}$ indicates a request for a $\overline{\text{REQ/ACK}}$ data-transfer handshake. This signal is received by the Initiator.
$\overline{\text{RESET}}$	7	I	**Reset:** $\overline{\text{RESET}}$ clears all registers. It does not force SCSI RST to the active state.
$\overline{\text{RST}}$	2	I/O	**SCSI Bus RESET:** The $\overline{\text{RST}}$ signal indicates an SCSI Bus $\overline{\text{RESET}}$ condition.
$\overline{\text{SEL}}$	5	I/O	**Select:** Used by an Initiator to select a Target, or by a Target to reselect an Initiator.
V_{DD}	23		**Power:** 5V ±10% DC Supply
V_{SS}	3, 12, 31, 36,42		**Ground:** 0V

FUNCTIONAL DESCRIPTION

The CA53C80 is easy to use because of its simple architecture. The chip allows direct control and monitoring of the SCSI Bus by providing a latch for each signal. However, portions of the protocol define timings which are much too quick for traditional microprocessors to control. Therefore, hardware support has been provided for DMA transfers, bus arbitration, phase change monitoring, bus disconnection, bus reset, parity generation, parity checking and device selection/ reselection.

The CA53C80 is a register-driven, clockwise peripheral device. Delays such as bus-free delay, bus-set delay, and bus-settle delay are implemented using gate delays. Inherent process variations may cause these delays to differ between devices, but the delays will be well within the proposed ANSI X359.2 specification. The structure is shown in Figure 1, and described briefly below:

The Mode register controls the operation of the chip. It determines the *target* or *initiator mode* and the desired mode of data transfer, initiates arbitration and enables or disables parity and interrupt generation. Reads and writes to the various registers are coordinated and enabled by the Address Decoding Logic.

The Current SCSI Data register contains current or non-latched data on the SCSI bus when it is read. Latched data appears in the Input Data register. Output data for the SCSI bus is held in the Data Output register.

The Select Enable register contains the device ID used in the selection or reselection phases. It is checked against the contents of the SCSI data bus during the (re)selection phase in Identification Comparison to determine if the chip's device has been selected.

The Current SCSI Bus Status register is used to check the unlatched status of 7 SCSI bus control signals and SCSI data parity bit. The remaining SCSI bus signals are controlled by the Initiator Command register, which also contains the arbitration status flags.

The Bus and Status register checks the status of SCSI control signals not monitored by the Current SCSI Bus Status register and monitors DMA transfer and interrupt condition status.

The Parity Generator produces an odd parity of the data in the Data Output register. Parity is output on the SCSI data parity bit when a send operation is in progress. Parity checking also checks the SCSI data parity bit for odd parity with incoming data. In each case, parity checking must be enabled. Parity, parity error and interrupt request bits are reset when the Reset Parity or Interrupt register is read.

Bus Phase Comparison compares the Current SCSI phase control signals with their assigned values in the Target Command register (used to set or reset target-operated SCSI bus control signals). An interrupt is generated when there is a phase mismatch.

DMA Control generates DMA control and handshaking signals and SCSI bus handshaking signals when in *DMA mode.* It also initiates Start DMA Send, Start DMA Target Receive and Start Initiator Receive events.

CA53C80 ENHANCEMENTS

The CA53C80 incorporates a number of improvements over the original 5380 SCSI Controller. These have been implemented while maintaining pin and architectural compatibility, and are listed below:

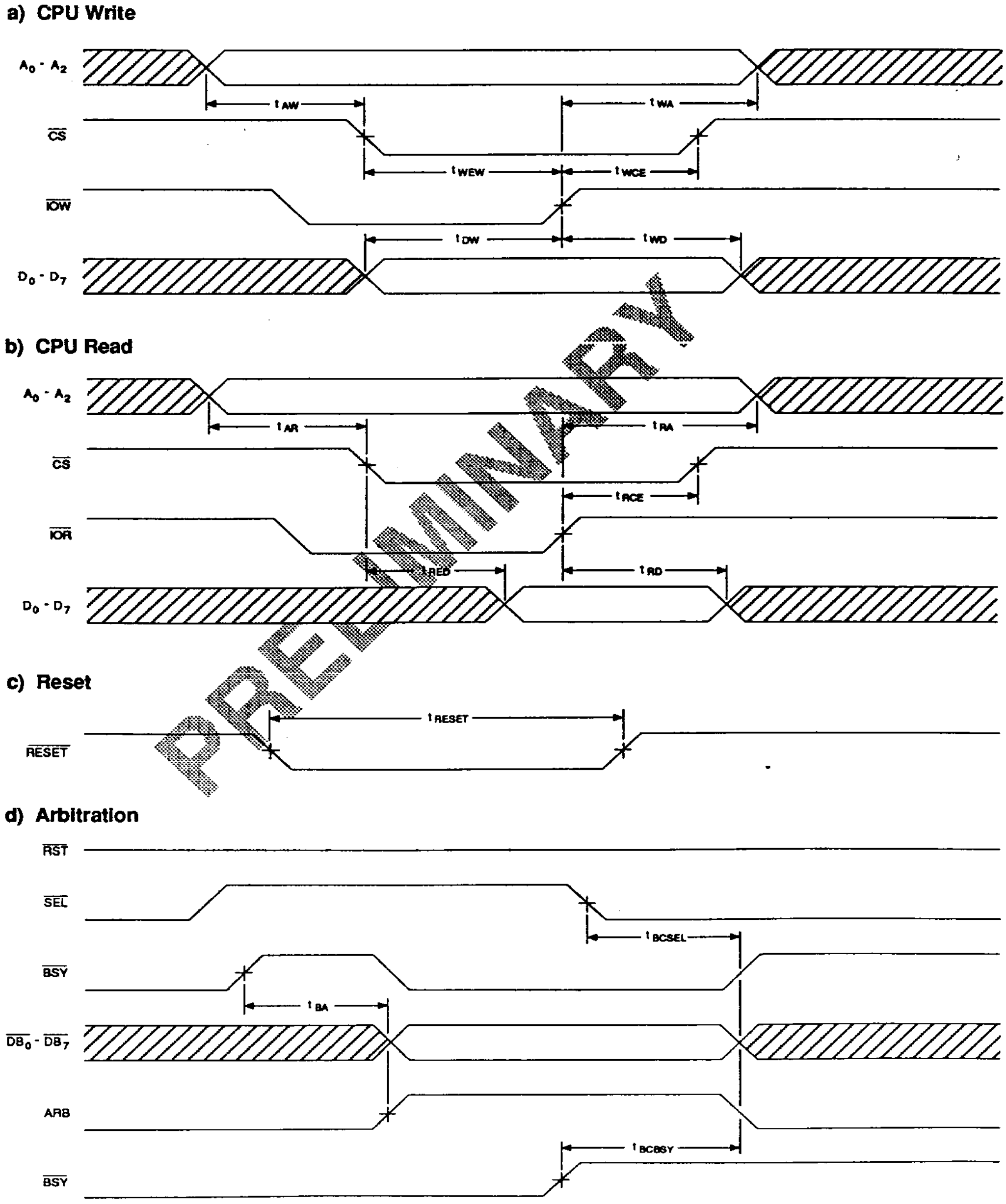
1. The CA53C80 will remain in a DMA mode after an EOP while the internal state machine returns to an idle condition. *Thus, the device does not respond to additional SCSI handshake attempts until another data transfer is explicity initiated.*
2. When operating as an *initiator* in DMA mode, $\overline{ACK}$ is properly deasserted after the receipt of a valid EOP.
3. High value internal pullups in the CA53C80 set under terminated SCSI pins to the stable inactive state.
4. A LASTBYTE status bit has been mapped to bit 7 of the Target Command register. This bit is set after a valid EOP has occurred, and the final byte has been successfully transmitted.

Table 2 : AC CHARACTERISTICS, ENVIRONMENTAL
(T_A = 0° to +70°C, V_{DD} = 5V ± 10%, V_{SS} = 0V)

Symbol	Parameter	1.5 Mbits/sec		3.0 Mbits/sec		Units
		Min	Max	Min	Max	
t_{AR}	Address setup to read enable (Note 2)	20		10		ns
t_{AW}	Address setup to write enable (Note 1)	20		10		ns
t_{BA}	ARBITRATE start from $\overline{\text{BSY}}$ false	1200	2200	1200	2200	ns
t_{BCBSY}	Bus clear from $\overline{\text{BSY}}$ false	400	1100	400	1100	ns
t_{BCSEL}	Bus clear from $\overline{\text{SEL}}$ true		600		600	ns
t_{DW}	Data setup to end of write enable (Note 1)	50		20		ns
t_{RA}	Address hold from end read enable (Note 2)	20		10		ns
t_{RCE}	Chip select hold from end of $\overline{\text{IOR}}$	0		0		ns
t_{RD}	Data hold time from end of $\overline{\text{IOR}}$	20		10		ns
t_{RED}	Data access time from read enable (Note 2)		130		65	ns
t_{RESET}	Minimum width of reset	200		50		ns
t_{WA}	Address hold from end write enable (Note 1)	20		10		ns
t_{WCE}	Chip select hold from end of $\overline{\text{IOW}}$	0		0		ns
t_{WD}	Data hold time from end of $\overline{\text{IOW}}$	30		10		ns
t_{WEW}	Write enable width (Note 1)	70		35		ns

Notes:
1. Write enable is the occurrence of $\overline{\text{IOW}}$ and $\overline{\text{CS}}$.
2. Read enable is the occurrence of $\overline{\text{IOR}}$ and $\overline{\text{CS}}$.

Figure 3 : ENVIRONMENTAL TIMING DIAGRAMS



a) CPU Write

b) CPU Read

c) Reset

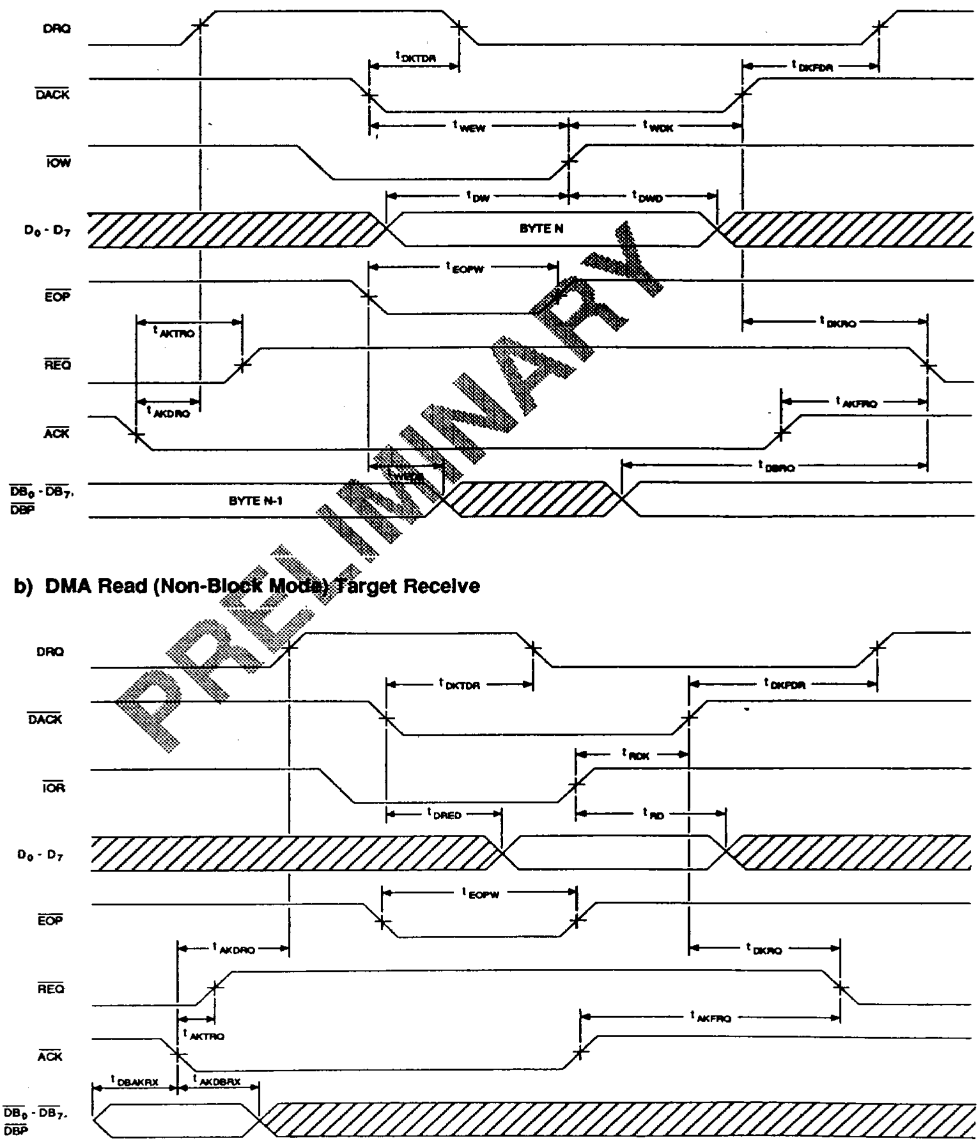
d) Arbitration

Table 3 : AC CHARACTERISTICS, TARGET, NON-BLOCK MODE
(T_A = 0° to +70°C, V_{DD} = 5V ± 10%, V_{SS} = 0V)

Symbol	Parameter	1.5 Mbits/sec		3.0 Mbits/sec		Units
		Min	Max	Min	Max	
t_{AKDBRX}	Data hold time from $\overline{\text{ACK}}$	50		20		ns
t_{AKDRQ}	$\overline{\text{ACK}}$ true to DRQ true (target)		110		60 Receive	ns
					50 Send	ns
t_{AKFRQ}	$\overline{\text{REQ}}$ from end of $\overline{\text{ACK}}$ ($\overline{\text{DACK}}$ false)		150		70	ns
t_{AKTRQ}	$\overline{\text{ACK}}$ true to $\overline{\text{REQ}}$ false		125		60	ns
t_{DBAKRX}	Data setup time to $\overline{\text{ACK}}$	20		10		ns
t_{DBRQ}	Data setup to $\overline{\text{REQ}}$ true (target)	60		30		ns
t_{DKFDR}	$\overline{\text{DACK}}$ false to DRQ true	30		20		ns
t_{DKRQ}	$\overline{\text{REQ}}$ from end of $\overline{\text{DACK}}$ ($\overline{\text{ACK}}$ false)		150		70	ns
t_{DKTDR}	DRQ false from $\overline{\text{DACK}}$ true		130		60	ns
t_{DRED}	Data access time from read enable (Note 3)		115		60	ns
t_{DW}	Data setup to end of write enable (Note 1)	50		20		ns
t_{DWD}	Data hold time from end of $\overline{\text{IOW}}$	40		20		ns
t_{EOPW}	Width of $\overline{\text{EOP}}$ pulse (Notes 2, 4)	100		50		ns
t_{RD}	Data hold time from end of $\overline{\text{IOR}}$	20		10		ns
t_{RDK}	$\overline{\text{DACK}}$ hold time from end of $\overline{\text{IOR}}$	0		0		ns
t_{WDK}	$\overline{\text{DACK}}$ hold from end of $\overline{\text{IOW}}$	0		0		ns
t_{WEDB}	Data hold from write enable	15		10		ns
t_{WEW}	Write enable width (Note 1)	100		50		ns

Notes:
1. Write enable is the occurrence of $\overline{\text{IOW}}$ and $\overline{\text{DACK}}$.
2. $\overline{\text{EOP}}$, $\overline{\text{IOW}}$, and $\overline{\text{DACK}}$ must be concurrently true for at least t_{EOPW} for proper recognition of the $\overline{\text{EOP}}$ pulse.
3. Read enable is the occurrence of $\overline{\text{IOR}}$ and $\overline{\text{DACK}}$.
4. $\overline{\text{EOP}}$, $\overline{\text{IOR}}$, and $\overline{\text{DACK}}$ must be concurrently true for at least t_{EOPW} for proper recognition of the $\overline{\text{EOP}}$ pulse.

Figure 4 : TARGET, NON-BLOCK MODE TIMING DIAGRAMS


a) DMA Write (Non-Block Mode) Target Send

b) DMA Read (Non-Block Mode) Target Receive

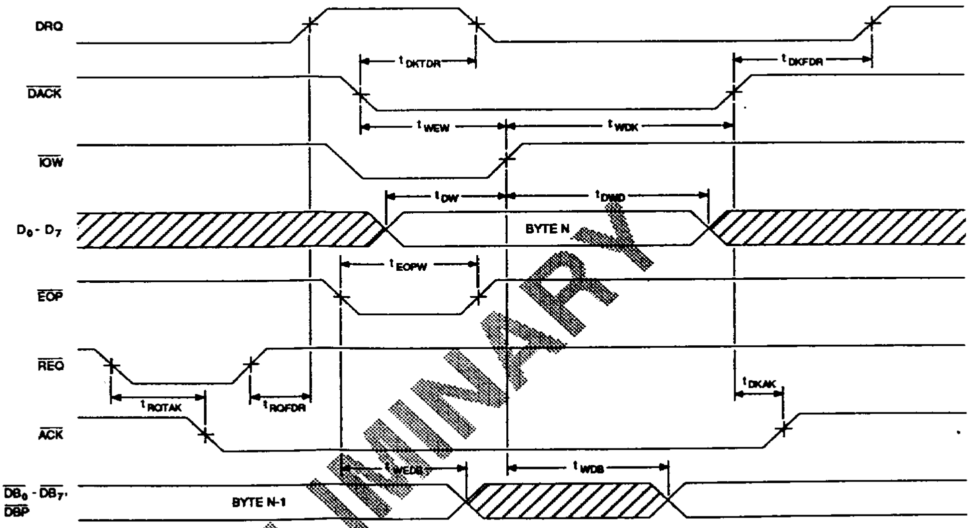
Table 4 : AC CHARACTERISTICS, INITIATOR, NON-BLOCK MODE
(T_A = 0° to +70°C, V_{DD} = 5V ± 10%, V_{SS} = 0V)

Symbol	Parameter	1.5 Mbits/sec		3.0 Mbits/sec		Units
		Min	Max	Min	Max	
t_{DBRQRX}	Data setup time to $\overline{REQ}$	20		10		ns
t_{DKAK}	$\overline{DACK}$ false to $\overline{ACK}$ false		150		70	ns
t_{DKAKRX}	$\overline{DACK}$ false to $\overline{ACK}$ false ($\overline{REQ}$ false)		160		80	ns
t_{DKFDR}	$\overline{DACK}$ false to DRQ true	30		20		ns
t_{DKTDR}	DRQ false from $\overline{DACK}$ true		130		60	ns
t_{DRED}	Data access time from read enable (Note 3)		115		60	ns
t_{DW}	Data setup to end of write enable (Note 1)	50		20		ns
t_{DWD}	Data hold time from end of $\overline{IOW}$	40		20		ns
t_{EOPW}	Width of $\overline{EOP}$ pulse (Notes 2, 4)	100		50		ns
t_{RD}	Data hold time from end of $\overline{IOR}$	20		10		ns
t_{RDK}	$\overline{DACK}$ hold time from end of $\overline{IOR}$	0		0		ns
t_{RQDBRX}	Data hold time from $\overline{REQ}$	50		20		ns
$t_{RQFAKRX}$	$\overline{REQ}$ false to $\overline{ACK}$ false ($\overline{DACK}$ false)		140		70	ns
t_{RQFDR}	$\overline{REQ}$ false to $\overline{DRQ}$ true		110		50	ns
t_{RQTAK}	$\overline{REQ}$ true to $\overline{ACK}$ true		160		80	ns
t_{RQTDR}	$\overline{REQ}$ true to DRQ true		150		70	ns
t_{WDB}	$\overline{IOW}$ false to valid SCSI data		100		50	ns
t_{WDK}	$\overline{DACK}$ hold from end of $\overline{IOW}$	0		0		ns
t_{WEDB}	Data hold from write enable	15		10		ns
t_{WEW}	Write enable width (Note 1)	100		50		ns

Notes:
1. Write enable is the occurrence of $\overline{IOW}$ and $\overline{DACK}$.
2. $\overline{EOP}$, $\overline{IOW}$, and $\overline{DACK}$ must be concurrently true for at least t_{EOPW} for proper recognition of the $\overline{EOP}$ pulse.
3. Read enable is the occurrence of $\overline{IOR}$ and $\overline{DACK}$.
4. $\overline{EOP}$, $\overline{IOR}$, and $\overline{DACK}$ must be concurrently true for at least t_{EOPW} for proper recognition of the $\overline{EOP}$ pulse.

Figure 5 : INITIATOR, NON-BLOCK MODE TIMING DIAGRAMS

a) DMA Write (Non-Block Mode) Initiator Send



b) DMA Read (Non-Block Mode) Initiator Receive

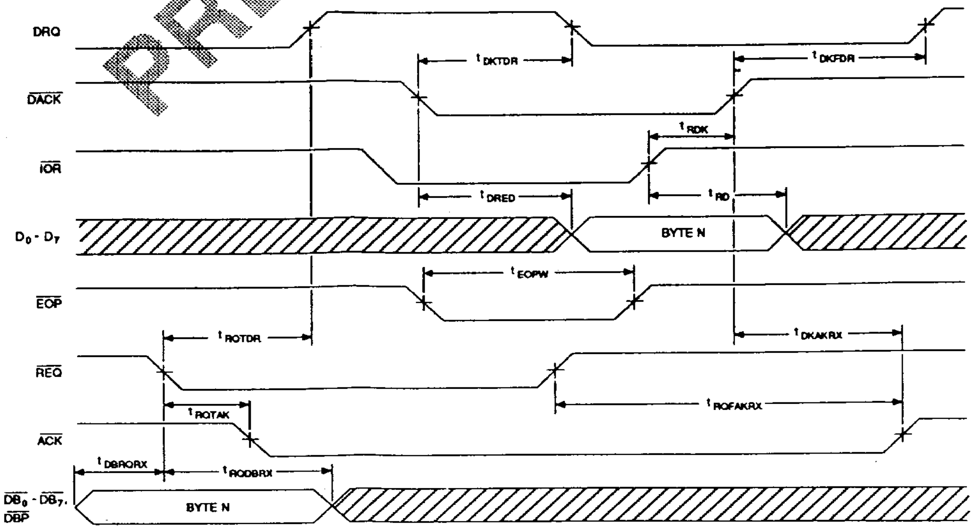
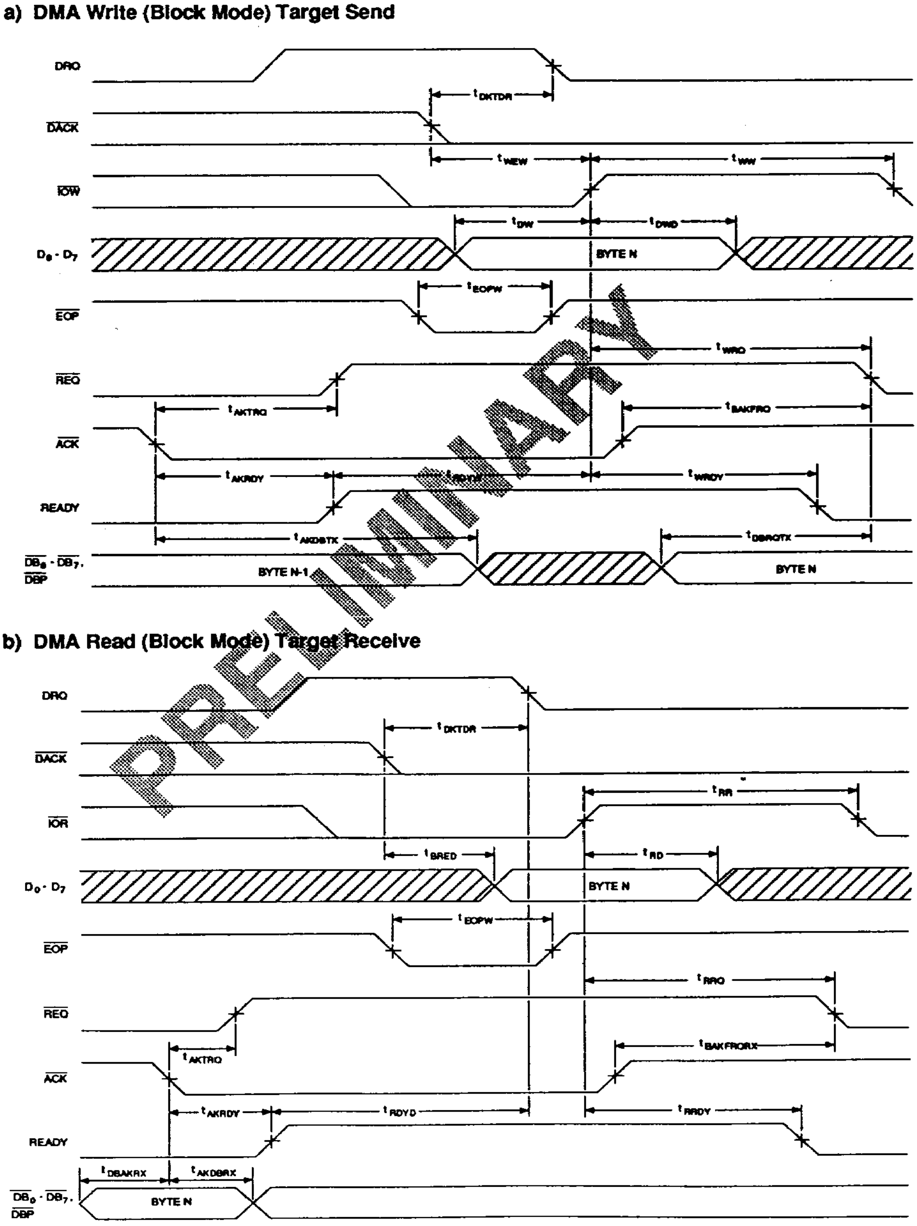


Table 5 : AC CHARACTERISTICS, TARGET, BLOCK-MODE
(T_A = 0° to +70°C, V_{DD} = 5V ± 10%, V_{SS} = 0V)

Symbol	Parameter	1.5 Mbits/sec		3.0 Mbits/sec		Units
		Min	Max	Min	Max	
t_{AKDBRX}	Data hold time from $\overline{\text{ACK}}$ true	50		20		ns
t_{AKDBTX}	Data hold time from $\overline{\text{ACK}}$ true	40		20		ns
t_{AKRDY}	$\overline{\text{ACK}}$ true from READY true		140		60 Receive	ns
					70 Send	ns
t_{AKTRQ}	$\overline{\text{ACK}}$ true to $\overline{\text{REQ}}$ false		125		50	ns
t_{BAKFRQ}	$\overline{\text{REQ}}$ from end of $\overline{\text{ACK}}$ ($\overline{\text{IOW}}$ false)		170		90	ns
$t_{BAKFRQRX}$	$\overline{\text{ACK}}$ false to $\overline{\text{REQ}}$ true ($\overline{\text{IOR}}$ false)		170		70	ns
t_{BRED}	Data access time from read enable (Note 3)		110		50	ns
t_{DBAKRX}	Data setup time to $\overline{\text{ACK}}$ true	20		10		ns
t_{DBRQTX}	Data setup to $\overline{\text{REQ}}$ true	60		30		ns
t_{DKTDR}	DRQ false from $\overline{\text{DACK}}$ true		130		60	ns
t_{DW}	Data setup to end of write enable (Note 1)	50		20		ns
t_{DWD}	Data hold time from end of $\overline{\text{IOW}}$	40		20		ns
t_{EOPW}	Width of $\overline{\text{EOP}}$ pulse (Notes 2, 4)	100		50		ns
t_{RD}	Data hold time from end of $\overline{\text{IOR}}$	20		10		ns
t_{RDYD}	READY true to valid data		50		20	ns
t_{RDYW}	READY true to $\overline{\text{IOW}}$ false	70		30		ns
t_{RR}	$\overline{\text{IOR}}$ recovery time	120		60		ns
t_{RRDY}	$\overline{\text{IOR}}$ false to READY false		140		70	ns
t_{RRQ}	$\overline{\text{IOR}}$ false to $\overline{\text{REQ}}$ true ($\overline{\text{ACK}}$ false)		190		70	ns
t_{WRDY}	$\overline{\text{IOW}}$ false to READY false		140		70	ns
t_{WRQ}	$\overline{\text{REQ}}$ from end of $\overline{\text{IOW}}$ ($\overline{\text{ACK}}$ false)		180		100	ns
t_{WEW}	Write enable width (Note 1)	100		50		ns
t_{WW}	Write recovery time	120		60		ns

Notes:
1. Write enable is the occurrence of $\overline{\text{IOW}}$ and $\overline{\text{DACK}}$.
2. $\overline{\text{EOP}}$, $\overline{\text{IOW}}$, and $\overline{\text{DACK}}$ must be concurrently true for at least t_{EOPW} for proper recognition of the $\overline{\text{EOP}}$ pulse.
3. Read enable is the occurrence of $\overline{\text{IOR}}$ and $\overline{\text{DACK}}$.
4. $\overline{\text{EOP}}$, $\overline{\text{IOR}}$, and $\overline{\text{DACK}}$ must be concurrently true for at least t_{EOPW} for proper recognition of the $\overline{\text{EOP}}$ pulse.

Figure 6 : TARGET, BLOCK-MODE TIMING DIAGRAMS



a) DMA Write (Block Mode) Target Send

b) DMA Read (Block Mode) Target Receive

Table 6 : DC CHARACTERISTICS (T_A = 0° to +70°C, V_{DD} = 5V ±10%, V_{SS} = 0V)

Symbol	Parameter	Test Conditions	Limits		Units
			Min	Max	
I_{DD}	Supply Current	All input pins not floating		20	mA
I_{IH}	HIGH-level Input (All pins except SCSI Bus)	V_{IH} = 5.5 V V_{IL} = 0		10	µA
I_{IHS}	HIGH-level Input (SCSI Bus pins)			50	µA
I_{IL}	LOW-level Input (All pins except SCSI Bus)	V_{IH} = 5.5 V V_{IL} = 0		–10	µA
I_{ILS}	LOW-level Input (SCSI Bus pins)			–50	µA
V_{IH}	Input High Voltage		2.0	5.25	V
V_{IL}	Input Low Voltage		–0.3	0.8	V
V_{OH}	Output HighVoltage	V_{DD} = 4.5 V I_{OH} = –3.0 mA	2.4		V
V_{OL}	Output Low Voltage (All pins except SCSI Bus)	V_{DD} = 4.5 V I_{OL} = 7.0 mA		0.5	V
V_{OLS}	Output Low Voltage (SCSI Bus pins)	V_{DD} = 4.5 V I_{OL} = 48.0 mA		0.5	V

Table 7 : RECOMMENDED OPERATING CONDITIONS

DC Supply Voltage		+4.5 V to +5.5 V
Operating Temperature Range	Commercial	0° to +70°C

Table 8 : ABSOLUTE MAXIMUM RATINGS

DC Supply Voltage	+8.0 V
Input, Output or I/O Voltage applied	V_{SS} – 0.5 V to V_{DD} + 0.5 V
Storage temperature range	–65° to +150°C
Maximum Package Power Dissipation	2 W

Stresses beyond those listed above may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of this specification is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

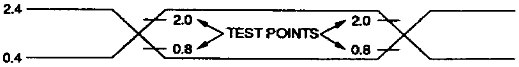


Figure 7 : AC TESTING OUTPUT WAVEFORMS

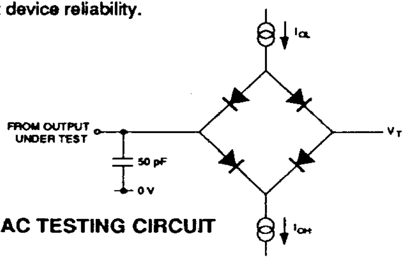


Figure 8 : AC TESTING CIRCUIT

INTERNAL REGISTER DESCRIPTION

The CA53C80 SCSI (Small Computer Systems Interface) device appears to the controlling microprocessor as a set of 13 registers accessed via 8 address slots. Since some registers are either read only or write only, a given address can be used to select more than one register according to whether a read or a write operation is being performed.

By reading and writing the appropriate registers, the microprocessor may initiate any SCSI Bus activity or may sample and assert any signal on the SCSI Bus. This allows all, or portions, of the SCSI protocol to be implemented in software. The CA53C80 registers are read (written) by activating $\overline{CS}$ with an address on $A_0 - A_2$ and then issuing an $\overline{IOR}$ ($\overline{IOW}$) pulse.

The CA53C80 registers fall into three functional groups:

Control Registers: Seven control registers are used by the CPU to set the various operational modes of the CA53C80, and to obtain status information. The control registers consist of the Bus and Status register, the Current SCSI Bus Status register, the Initiator Command register, the Mode register, the Reset Parity/Interrupts register, the Select Enable register and the Target Command register.

Data Registers: Three data registers are used to transfer SCSI commands, data, status, and message bytes between the microprocessor Data Bus and the SCSI Bus. The CA53C80 does not interpret any information that passes through the data registers. The data registers consist of the transparent Current SCSI Data register, the Output Data register and the Input Data register.

DMA Registers: Three write only registers initiate all DMA activity. These consist of the Start DMA Send register (Port 5), the Start DMA Target Receive register (Port 6) and the Strart DMA Initiator Receive register (Port 7). Simply writing to these registers starts the DMA transfers.

Data presented to the CA53C80 on signals D_0 - D_7 during the register write is meaningless and has no effect on the operation. Prior to writing these registers, the Block Mode DMA bit (Bit 7), the DMA Mode bit (Bit 1) and the TARGETMODE bit (Bit 6) in the Mode register (Port 2) must be appropriately set.

The CA53C80 registers are listed in Table 9 and described individually in the sub-sections which follow.

Table 9 : REGISTER SUMMARY

Register Name	Type		Address		
			A_2	A_1	A_0
Bus and Status	Control	R	1	0	1
Current SCSI Bus Status	Control	R	1	0	0
Current SCSI Data	Data	R	0	0	0
Initiator Command	Control	R/W	0	0	1
Input Data	Data	R	1	1	0
Mode	Control	R/W	0	1	0
Output Data	Data	W	0	0	0
Reset Parity/Interrupts	Control	R	1	1	1
Select Enable	Control	W	1	0	0
Start DMA Initiator Receive	DMA	W	1	1	1
Start DMA Send	DMA	W	1	0	1
Start DMA Target Receive	DMA	W	1	1	0
Target Command	Control	R/W	0	1	1

Current SCSI Data Register: Addr 0 (Read Only)

The Current SCSI Data register is a read only register that allows the microprocessor to read the active SCSI Data Bus. This is accomplished by activating $\overline{CS}$ with an address on A_2 - A_0 of 000 and issuing an $\overline{IOR}$ pulse. If parity checking is enabled, the SCSI Bus parity is checked at the beginning of the read cycle. This register is used during a programmed I/O data read or during Arbitration to check for higher priority arbitrating devices. Parity is not guaranteed valid during Arbitration. (See Figure 9)

Output Data Register: Addr 0 (Write Only)

The Output Data register is a write only register used to send data to the SCSI Bus. This is accomplished by either using a normal CPU write, or under DMA control, by using $\overline{IOW}$ and $\overline{DACK}$. This register is also used to assert the proper ID bits on the SCSI bus during the Arbitration and Selection phases. (See Figure 9)

Input Data Register: Addr 6 (Read Only)

The Input Data register is a read only register used to read latched data from the SCSI Bus. Data is latched either during a DMA Target receive operation when $\overline{ACK}$ goes active or during a DMA Initiator receive when $\overline{REQ}$ goes active. The DMA Mode bit (Port 2, Bit 1) must be set before data can be latched in the Input Data register. This register may be read under DMA control using $\overline{IOR}$ and $\overline{DACK}$. Parity can be optionally checked when the Input Data register is loaded. (See Figure 9)

7	6	5	4	3	2	1	0
$\overline{DB_7}$	$\overline{DB_6}$	$\overline{DB_5}$	$\overline{DB_4}$	$\overline{DB_3}$	$\overline{DB_2}$	$\overline{DB_1}$	$\overline{DB_0}$

Figure 9 : INPUT DATA, OUTPUT DATA and CURRENT SCSI DATA REGISTERS

Initiator Command Register: Addr 1 (Read/Write)

The Initiator Command register is a read/write register used to assert certain SCSI Bus signals, monitor those signals, and monitor the progress of bus arbitration. Many of these bits are significant only when being used as an Initiator, though most can also be used during Target role operation.

7	6	5	4	3	2	1	0
ASSERT $\overline{RST}$	AIP	LA	ASSERT $\overline{ACK}$	ASSERT $\overline{BSY}$	ASSERT $\overline{SEL}$	ASSERT $\overline{ATN}$	ASSERT DATA BUS

a) Register Read

7	6	5	4	3	2	1	0
		0					
ASSERT $\overline{RST}$	TEST MODE	DIFF ENBL	ASSERT $\overline{ACK}$	ASSERT $\overline{BSY}$	ASSERT $\overline{SEL}$	ASSERT $\overline{ATN}$	ASSERT DATA BUS

b) Register Write

Figure 10 : INITIATOR COMMAND REGISTERS

The following describes the operation of all bits in the Initiator Command register.

Bit 7 - ASSERT RST

Whenever a *one* is written to bit 7 of the Initiator Command register, the $\overline{RST}$ signal is asserted on the SCSI Bus. The $\overline{RST}$ signal remains asserted until this bit is reset or until an external $\overline{RESET}$ occurs. After this bit is set (1), IRQ goes active and all internal logic and control registers are reset (except for the interrupt latch and the ASSERT $\overline{RST}$ bit). Writing a *zero* to bit 7 of the Initiator Command register de-asserts the $\overline{RST}$ signal. Reading this register simply reflects the status of this bit.

Bit 6 - AIP (Arbitration in Progress) (Read Bit)

This bit is used to determine if Arbitration is in progress. For this bit to be active, the ARBITRATE bit (port 2, bit 0) must have been set previously. It indicates that a Bus-Free condition has been detected and that the chip has asserted $\overline{BSY}$ and the contents of the Output Data register (port 0) onto the SCSI Bus. AIP will remain active until the ARBITRATE bit is reset.

Bit 6 - TEST MODE (Write Bit)

This bit may be written during a test environment to disable all output drivers, effectively removing the CA53C80 from the circuit. Resetting this bit returns the part to normal operation.

Bit 5 - LA (Lost Arbitration) (Read Bit)

This bit, when active, indicates that the CA53C80 detected a Bus-Free condition, arbitrated for use of the bus by asserting $\overline{BSY}$ and its ID on the Data Bus, and lost Arbitration due to $\overline{SEL}$ being asserted by another bus device. For this bit to be active, the ARBITRATE bit (port 2, bit 0) must be active.

Bit 5 - DIFF ENBL (Differential Enable) (Write Bit)

This bit should be written with a *zero* for proper operation.

Bit 4 - ASSERT ACK

This bit is used by the bus initiator to assert $\overline{ACK}$ on the SCSI Bus. In order to assert $\overline{ACK}$, the TARGETMODE bit (port 2, bit 6) must be FALSE. Writing a *zero* to this bit resets $\overline{ACK}$ on the SCSI Bus. Reading this register simply reflects the status of this bit.

Bit 3 - ASSERT BSY

Writing a *one* into this bit position asserts $\overline{BSY}$ onto the SCSI Bus. Conversely, a *zero* resets the $\overline{BSY}$ signal. Asserting $\overline{BSY}$ indicates a successful selection or reselection and resetting this bit creates a Bus-Disconnect condition. Reading this register simply reflects the status of this bit.

Bit 2 - ASSERT SEL

Writing a *one* into this bit position asserts $\overline{SEL}$ onto the SCSI Bus. $\overline{SEL}$ is normally asserted after Arbitration has been successfully completed. $\overline{SEL}$ may be de-asserted by resetting this bit to a *zero*. A read of this register simply reflects the status of this bit.

Bit 1 - ASSERT ATN

$\overline{ATN}$ may be asserted on the SCSI Bus by setting this bit to a *one* if the TARGETMODE bit (port 2, bit 6) is FALSE. $\overline{ATN}$ is normally asserted by the initiator to request a Message Out bus phase. Note that since ASSERT $\overline{SEL}$ and ASSERT $\overline{ATN}$ are in the same register, a select with $\overline{ATN}$ may be implemented with one CPU write. $\overline{ATN}$ may be de-asserted by resetting this bit to *zero*. A read of this register simply reflects the status of this bit.

Bit 0 - ASSERT DATA BUS

The ASSERT DATA BUS bit, when set allows the contents of the Output Data register to be enabled as chip outputs on the signals $\overline{DB_0}$ - $\overline{DB_7}$. Parity is also generated and asserted on $\overline{DBP}$.

When connected as an Initiator, the outputs are only enabled if the TARGETMODE bit (port 2, bit 6) is FALSE, the received signal I/O is FALSE, and the phase signals (C/D, I/O, and MSG) match the contents of ASSERT C/D, ASSERT I/O, and ASSERT MSG in the Target Command register.

This bit should also be set during DMA send operations.

Mode Register: Addr 2 (Read/Write)

The Mode register is used to control the operation of the chip. This register determines whether the CA53C80 operates as an Initiator or a Target, whether DMA transfers are being used, whether parity is checked, and whether interrupts are generated on various external conditions. This register may be read to check the value of these internal control bits. Figure 11 describes the operation of these control bits.

7	6	5	4	3	2	1	0
BLOCK MODE DMA	TARGET MODE	ENABLE PARITY CHECKING	ENABLE PARITY INTERRUPT	ENABLE EOP INTERRUPT	MONITOR BUSY	DMA MODE	ARBITRATE

Figure 11 : MODE REGISTER

Bit 7 - BLOCK MODE DMA

The BLOCK MODE DMA bit controls the characteristics of the DMA DRQ-DACK handshake. When this bit is reset (0) and the DMA MODE bit is active (1), the DMA handshake uses the normal interlocked handshake, and the rising edge of DACK indicates the end of each byte being transferred.

In block mode operations, BLOCK MODE DMA bit set (1) and DMA MODE bit set (1), the end of IOR or IOW signifies the end of each byte transferred and DACK is allowed to remain active throughout the DMA operation. READY can then be used to request the next transfer.

Bit 6 - TARGETMODE

The TARGETMODE bit allows the CA53C80 to operate as either an SCSI Bus Initiator, bit reset (0), or as an SCSI Bus Target device, bit set (1). In order for the signals ATN and ACK to be asserted on the SCSI Bus, the TARGETMODE bit must be reset (0). In order for the signals C/D, I/O, MSG and REQ to be asserted on the SCSI Bus, the TARGETMODE bit must be set (1).

Bit 5 - ENABLE PARITY CHECKING

The ENABLE PARITY CHECKING bit determines whether parity errors will be ignored or saved in the parity error latch. If this bit is reset (0), parity will be ignored. Conversely, if this bit is set (1), parity errors will be saved.

Bit 4 - ENABLE PARITY INTERRUPT

The ENABLE PARITY INTERRUPT bit, when set (1), will cause an interrupt (IRQ) to occur if a parity error is detected. A parity interrupt will only be generated if the ENABLE PARITY CHECKING bit (Bit 5) is also set (1).

Bit 3 - ENABLE EOP INTERRUPT

The ENABLE EOP INTERRUPT, when set (1), causes an interrupt to occur when an EOP (End of Process) signal is received from the DMA controller logic.

Bit 2 - MONITOR BUSY

The MONITOR BUSY bit, when set (1), causes an interrupt to be generated for an unexpected loss of BSY. When the interrupt is generated due to loss of BSY, the lower six bits of the Initiator Command register are reset (0) and all signals are removed from the SCSI Bus.

Bit 1 - DMA MODE

The DMA MODE bit is normally used to enable a DMA transfer and must be set (1) prior to writing Ports 5 through 7. Ports 5 through 7 are used to start DMA transfers. The TARGETMODE bit (Port 2, Bit 6) must be consistent with writes to port 6 and 7 (i.e. set (1) for a write to port 6 and reset (0) for a write to Port 7). The control bit ASSERT DATA BUS (port 1, bit 0) must be set (1) for all DMA send operations. In the DMA mode, REQ and ACK are automatically controlled.

The DMA MODE bit is not reset upon the receipt of an EOP signal. Any DMA transfer may be stopped by resetting (0) this bit location, though care must be taken not to cause CS and DACK to be active simultaneously.

Bit 0 - ARBITRATE

The ARBITRATE bit is set (1) to start the Arbitration process. Prior to setting this bit, the Output Data register should contain the proper SCSI device ID value. Only one data bit should be active for SCSI Bus Arbitration. The CA53C80 will wait for a Bus-Free condition before entering the Arbitration phase. The results of the Arbitration phase may be determined by reading the status bits LA and AIP (port 1, bits 5 and 6, respectively).

Target Command Register: Addr 3 (Read/Write)

When connected as a target device, the Target Command register allows the CPU to control the SCSI Bus Information Transfer phase and/or to assert REQ simply by writing this register. The TARGETMODE bit (Port 2, Bit 6) must be set (1) for bus assertion to occur. The SCSI Bus phases are given in Table 10.

Table 10 : SCSI INFORMATION TRANSFER PHASES

Bus Phase	ASSERT $\overline{\mathrm{I/O}}$	ASSERT $\overline{\mathrm{C/D}}$	ASSERT $\overline{\mathrm{MSG}}$
Data Out	0	0	0
Unspecified	0	0	1
Command	0	1	0
Message Out	0	1	1
Data In	1	0	0
Unspecified	1	0	1
Status	1	1	0
Message In	1	1	1

If, when connected as an Initiator with DMA Mode set (1), the phase lines ($\overline{\mathrm{I/O}}$, $\overline{\mathrm{C/D}}$, and $\overline{\mathrm{MSG}}$) do not match the phase bits in the Target Command register, a phase-mismatch interrupt is generated when $\overline{\mathrm{REQ}}$ goes active. In order to send data as an Initiator, the ASSERT $\overline{\mathrm{I/O}}$, ASSERT $\overline{\mathrm{C/D}}$, and ASSERT $\overline{\mathrm{MSG}}$ bits must match the corresponding bits in the Current SCSI Bus Status register (Port 4). The ASSERT $\overline{\mathrm{REQ}}$ bit (Bit 3) has no meaning when operating as an Initiator.

7	6	5	4	3	2	1	0
X	X	X	X	ASSERT $\overline{\mathrm{REQ}}$	ASSERT $\overline{\mathrm{MSG}}$	ASSERT $\overline{\mathrm{C/D}}$	ASSERT $\overline{\mathrm{I/O}}$

Figure 12 : TARGET COMMAND REGISTER

Current SCSI Bus Status Register: Addr 4 (Read Only)

The Current SCSI Bus Status register is a read only register which is used to monitor seven SCSI Bus control signals plus the Data Bus parity bit. For example, an Initiator device can use this register to determine the current bus phase and to poll $\overline{\mathrm{REQ}}$ for pending data transfers. This register may also be used to determine the current bus phase and to poll $\overline{\mathrm{REQ}}$ for pending data transfers. This register may also be used to determine why a particular interrupt occurred. Figure 13 describes the Current SCSI Bus Status register.

7	6	5	4	3	2	1	0
$\overline{\mathrm{RST}}$	$\overline{\mathrm{BSY}}$	$\overline{\mathrm{REQ}}$	$\overline{\mathrm{MSG}}$	$\overline{\mathrm{C/D}}$	$\overline{\mathrm{I/O}}$	$\overline{\mathrm{SEL}}$	$\overline{\mathrm{DBP}}$

Figure 13 : CURRENT SCSI BUS STATUS REGISTER

Select Enable Register: Addr 4 (Write Only)

The Select Enable register is a write only register which is used as a mask to monitor a signal ID during a selection attempt. The simultaneous occurrence of the correct ID bit, $\overline{\mathrm{BSY}}$ FALSE, and $\overline{\mathrm{SEL}}$ set (1) causes an interrupt. This interrupt can be disabled by resetting all bits in this register. If the ENABLE PARITY CHECKING bit (Port 2, Bit 5) is set (1), parity will be checked during selection.

7	6	5	4	3	2	1	0
$\overline{\mathrm{DB}_7}$	$\overline{\mathrm{DB}_6}$	$\overline{\mathrm{DB}_5}$	$\overline{\mathrm{DB}_4}$	$\overline{\mathrm{DB}_3}$	$\overline{\mathrm{DB}_2}$	$\overline{\mathrm{DB}_1}$	$\overline{\mathrm{DB}_0}$

Figure 14 : SELECT ENABLE REGISTER

Bus and Status Register: Addr 5 (Read Only)

The Bus and Status register is a read only register which can be used to monitor the remaining SCSI control signals not found in the Current SCSI Bus Status register ($\overline{\mathrm{ATN}}$ and $\overline{\mathrm{ACK}}$), as well as six other status bits. The following describes each bit of the Bus and Status register individually.

7	6	5	4	3	2	1	0
END OF DMA	DMA RE-QUEST	PARITY ERROR	INTER-RUPT RE-QUEST ACTIVE	PHASE MATCH	BUSY ERROR	$\overline{\mathrm{ATN}}$	$\overline{\mathrm{ACK}}$

Figure 15 : BUS AND STATUS REGISTER

Bit 7 - END OF DMA TRANSFER

The END OF DMA TRANSFER bit is set if $\overline{\mathrm{EOP}}$, $\overline{\mathrm{DACK}}$, and either $\overline{\mathrm{IOR}}$ or $\overline{\mathrm{IOW}}$ are simultaneously active for at least 100 ns. Since the $\overline{\mathrm{EOP}}$ signal can occur during the last byte sent to the Output Data register (Port 0), the $\overline{\mathrm{REQ}}$ and $\overline{\mathrm{ACK}}$ signals should be monitored to ensure that the last byte has been transferred. This bit is reset when the DMA MODE bit is reset (0) in the Mode register (Port 2).

Bit 6 - DMA REQUEST

The DMA REQUEST bit allows the CPU to sample the output pin DRQ. DRQ can be cleared by asserting $\overline{\mathrm{DACK}}$ or by resetting the DMA MODE bit (Bit 1) in the Mode register (Port 2). The DRQ signal does not reset when a phase-mismatch interrupt occurs.

Bit 5 - PARITY ERROR

This bit is set if a parity error occurs during a data receive or a device selection. The PARITY ERROR bit can only be set (1) if the ENABLE PARITY CHECK bit (Port 2, bit 5) is active (1). This bit may be cleared by reading the Reset Parity/Interrupt register (Port 7).

Bit 4 - INTERRUPT REQUEST ACTIVE

This bit is set if an enabled interrupt condition occurs. It reflects the current state of the IRQ output and can be cleared by reading the Reset Parity/Interrupt register (Port 7).

Bit 3 - PHASE MATCH

The SCSI signals, MSG, CD, and I/O, represent the current Information Transfer phase. The PHASE MATCH bit indicates whether the current SCSI Bus phase matches the lower 3 bits of the Target Command register. PHASE MATCH is continuously updated and is only significant when operating as a Bus Initiator. A phase match is required for data transfers to occur on the SCSI Bus.

Bit 2 - BUSY ERROR

The BUSY ERROR bit is set (1) if an unexpected loss of the BSY signal has occurred. This latch is set whenever the MONITOR BUSY bit (Port 2, Bit 2) is TRUE and BSY signal is *high*. An unexpected loss of BSY will disable any SCSI outputs and will reset the DMA MODE bit (Port 2, Bit 1).

Bit 1 - ATN

This bit reflects the condition of the SCSI Bus control signal ATN. This signal is normally monitored by the Target device.

BIT 0 - ACK

This bit reflects the condition of the SCSI Bus control signal ACK. This signal is normally monitored by the Target device.

Start DMA Send: Addr 5 (Write Only)

This register is written to initiate a DMA send, from the DMA to the SCSI Bus, for either Initiator or Target role operations. The DMA MODE bit (port 2, bit 1) must be set prior to writing this register.

Start DMA Target Receive: Addr 6 (Write Only)

This register is written to initiate a DMA receive - from the SCSI Bus to the DMA, for Initiator operation only. The DMA MODE bit (bit 6) must be FALSE (0) in the Mode register (port 2) prior to writing this register.

Start DMA Initiator Receive: Addr 7 (Write Only)

This register is written to initiate a DMA receive - from the SCSI Bus to the DMA, for Initiator operation only. The DMA MODE bit (bit 6) must be FALSE (0) in the Mode register (port 2) prior to writing this register.

Reset Parity/Interrupt: Addr 7 (Read Only)

Reading this register resets the PARITY ERROR bit (bit 5), the INTERRUPT REQUEST bit (bit 4), and the BUSY ERROR bit (bit 2) in the Bus and Status register (port 5).

ARBITRATION

Arbitration is accomplished using a Bus-Free filter to continuously monitor BSY. If BSY remains inactive for at least 400 ns then the SCSI Bus is considered free and Arbitration may begin. Arbitration will begin if the bus is free, SEL is inactive, and the ARBITRATION bit (Port 2, Bit 0) is set (1). Once arbitration has begun (BSY asserted), an arbitration delay of 2.2 µs must elapse before the Data Bus can be examined to determine if Arbitration has been won. This delay must be implemented in the controlling software driver.

INTERRUPTS

The CA53C80 provides an interrupt output (IRQ) to indicate task completion or abnormal bus occurrences. The use of interrupts is optional and may be disabled by resetting the appropriate bits in the Mode register (Port 2) or the Select Enable register (Port 4).

When an interrupt occurs, the Bus and Status register and the Current SCSI Bus Status register must be read to determine which condition created the interrupt. IRQ is reset by reading the Reset Parity/Interrupt register (Port 7) or by applying an external chip reset.

Assuming the CA53C80 has been properly initialized, five interrupts can be generated if the chip is selected or reselected: if an EOP signal occurs during a DMA transfer, if a SCSI Bus reset occurs, if a parity error occurs during a data transfer, if a bus phase mismatch occurs, or if an SCSI Bus disconnection occurs.

Selection/Reselection

The CA53C80 can generate a select interrupt if SEL is *low*, its device ID is set (1), and BSY is *high* for at least a bus-settle delay (400 ns). If I/O is active, this should be considered a reselect interrupt. The correct ID bit is determined by a match in the Select Enable register (Port 4). Only a single bit match is required to generate an interrupt. This interrupt is disabled by writing zeros into all bits of the Select Enable register.

If parity is supported, parity should also be good during the selection phase. Therefore, if the ENABLE PARITY bit (Port 2, Bit 5) is set (1), then the PARITY ERROR bit should be checked to ensure that a proper selection has occurred. The ENABLE PARITY INTERRUPT bit need not be set for this interrupt to be generated.

The proposed SCSI specification also requires that no more than two device IDs be active during the selection process. To ensure this, the Current SCSI Data register (Port 0) should be read.

Valid values for the Bus and Status register (Port 5) and the Current SCSI Bus Status register (Port 4) are given in Tables 11 and 12, respectively.

End of Process (EOP) Interrupt

An End of Process signal ($\overline{\text{EOP}}$) which occurs during a DMA transfer, DMA MODE set (1), sets the END OF DMA Status bit (Port 5, Bit 7) and (optionally) generates an interrupt if the ENABLE $\overline{\text{EOP}}$ INTERRUPT bit (Port 2, Bit 3) is set (1). The $\overline{\text{EOP}}$ pulse is not recognized (END OF DMA bit set) unless $\overline{\text{EOP}}$, $\overline{\text{DACK}}$, and either $\overline{\text{IOR}}$ or $\overline{\text{IOW}}$ are concurrently active for at least 100 ns. DMA transfers can still occur if $\overline{\text{EOP}}$ was not asserted at the correct time. This interrupt is disabled by resetting the ENABLE $\overline{\text{EOP}}$ INTERRUPT bit.

Valid values for the Bus and Status register (Port 5) and the Current SCSI Bus Status register (Port 4) for this interrupt are given in Tables 11 and 12, respectively.

The END OF DMA bit is used to determine when a block transfer is complete. Receive operations are complete when there is no data left in the chip and no additional handshakes are occurring. The only exception to this occurs when receiving data as an Initiator, and the Target opts to send additional data for the same phase. In this case, $\overline{\text{REQ}}$ goes active and the new data is present in the Input Data register. Since a phase-mismatch interrupt will not occur, $\overline{\text{REQ}}$ and $\overline{\text{ACK}}$ need to be sampled to determine that the Target is attempting to send more data.

For send operations, the END OF DMA bit is set when the DMA finishes its transfer, but the SCSI transfer may still be in progress. If connected as a Target, $\overline{\text{REQ}}$ and $\overline{\text{ACK}}$ should be sampled until both are *high*. If connected as an initiator, a phase change interrupt can be used to signal the completion of the previous phase.

The Target can also request additional data for the same phase. In this case, a phase change will not occur and both $\overline{\text{REQ}}$ and $\overline{\text{ACK}}$ must be sampled to determine when the last byte was transferred.

SCSI Bus Reset

The CA53C80 generates an interrupt when the $\overline{\text{RST}}$ signal transitions to *low*. The device releases all bus signals within a bus-clear delay (800 ns) of this transition. This interrupt also occurs after setting the ASSERT $\overline{\text{RST}}$ bit (Port 1, Bit 7). This interrupt cannot be disabled. (Note: $\overline{\text{RST}}$ is not latched in Bit 7 of the Current SCSI Bus Status register and may not be active when this port is read. For this case, the Bus Reset interrupt may be determined by default).

Valid values for the Bus and Status register (Port 5) and the Current SCSI Bus Status register (Port 4) are given in Tables 11 and 12, respectively.

Parity Error

An interrupt is generated for a received parity error if the ENABLE PARITY CHECK (Bit 5) and the ENABLE PARITY INTERRUPT (Bit 4) bits are set (1) in the Mode register (Port 2). Parity is checked during a read of the Current SCSI Data register (Port 0) and during a DMA receive operation. A parity error can be detected without generating an interrupt by disabling the ENABLE PARITY INTERRUPT bit and checking the PARITY ERROR flag (Port 5, Bit 5).

Valid values for the Bus and Status register (Port 5) and the Current SCSI Bus Status register (Port 4) are given in Tables 11 and 12, respectively.

Bus Phase Mismatch

The SCSI phase lines are comprised of the signals $\overline{\text{I/O}}$, $\overline{\text{C/D}}$, and $\overline{\text{MSG}}$. These signals are compared with the corresponding bits in the Target Command register. ASSERT $\overline{\text{I/O}}$ (Bit 0), ASSERT $\overline{\text{C/D}}$ (Bit 1), and ASSERT $\overline{\text{MSG}}$ (Bit 2). The comparison occurs continually and is reflected in the PHASE MATCH bit (Bit 3) of the Bus and Status register (Port 5). If the DMA MODE bit (Port 2, Bit 1) is set (1) and a phase mismatch occurs when $\overline{\text{REQ}}$ transitions from *high* to *low*, an interrupt (IRQ) is generated.

A phase mismatch prevents the recognition of $\overline{\text{REQ}}$ and removes the chip from the bus during an Initiator send operation. That is, $\overline{\text{DB}_0}$ - $\overline{\text{DB}_7}$ and $\overline{\text{DBP}}$ are not driven even though the ASSERT DATA BUS bit (Port 1, Bit 0) is active. This interrupt is only significant when connected as an Initiator though it is possible for it to occur when connected as a Target if another device is driving the phase lines to a different state. It is disabled by resetting the DMA MODE bit.

Valid values for the Bus and Status register (Port 5) and the Current SCSI Bus Status register (Port 4) are given in Tables 11 and 12, respectively.

Loss of BSY

If the MONITOR BUSY bit (Bit 2) in the Mode register (Port 2) is active, an interrupt is generated if the $\overline{\text{BSY}}$ signal goes *high* for at least a bus-settle delay (400 ns). This interrupt is disabled by resetting the MONITOR BUSY bit. Register values are given in Tables 11 and 12.

Table 11 : BUS AND STATUS REGISTER INTERRUPT SETTINGS

Interrupt Signals	Bit/Function							
	7	6	5	4	3	2	1	0
	End of DMA	DMA Request	Parity Error	Interrupt Request Active	Phase Match	Busy Error	ATN	ACK
Selection/Reselection	0	0	0	1	X	0	X	0
End of Process Interrupt	1	0	0	1	0	0	0	X
SCSI Bus Reset	0	0	0	X	X	X	1	X
Parity Error	0	X	1	1	1	0	X	X
Bus Phase Mismatch	0	0	0	1	0	0	X	0
Loss of BSY	0	0	0	1	X	1	0	0

Table 12 : CURRENT SCSI BUS STATUS REGISTER INTERRUPT SETTINGS

Interrupt Signals	Bit/Function							
	7	6	5	4	3	2	1	0
	$\overline{RST}$	$\overline{BSY}$	$\overline{REQ}$	$\overline{MSG}$	$\overline{C/D}$	$\overline{I/O}$	$\overline{SEL}$	$\overline{DBP}$
Selection/Reselection	0	0	0	X	X	X	1	X
End of Process Interrupt	0	1	X	X	X	X	0	X
SCSI Bus Reset	0	0	0	X	X	X	1	X
Parity Error	0	1	0	X	X	X	1	X
Bus Phase Mismatch	0	1	1	X	X	X	0	X
Loss of BSY	0	0	0	X	X	X	1	X

RESET CONDITIONS

Three possible reset situations exist with the CA53C80, as described individually as follows:

Hardware Chip Reset

When the $\overline{RESET}$ signal is active for at least 200 ns, the CA53C80 is reinitialized and all internal logic and control registers are cleared. This is a chip reset only and does not create an SCSI Bus-Reset condition.

SCSI Bus Reset (RST) Received

When a SCSI $\overline{RST}$ signal is received, an IRQ interrupt is generated and a chip reset is performed. All internal logic and registers are cleared, except for the IRQ interrupt latch and the ASSERT $\overline{RST}$ bit (Bit 7) in the Initiator Command register (Port 1). Note that the $\overline{RST}$ signal is sampled by reading the Current SCSI Bus Status register (Port 4). However, this signal is not latched and may not be present when this port is read.

SCSI Bus Reset (RST) Issued

If the CPU sets the ASSERT $\overline{RST}$ bit (Bit 7) in the Initiator Command register (Port 1) the $\overline{RST}$ signal goes active on the SCSI Bus and an internal reset is performed. Again, all internal logic and registers are cleared except for the IRQ interrupt latch and the ASSERT $\overline{RST}$ bit (Bit 7) in the Initiator Command register (Port 1). The $\overline{RST}$ signal will continue to be active until the ASSERT $\overline{RST}$ bit is reset or until a hardware reset occurs.

DATA TRANSFERS

Data may be transferred between SCSI Bus devices in one of four modes: Programmed I/O, Normal DMA, Block Mode DMA, and Pseudo DMA. Note that for all data transfer operations, $\overline{DACK}$ and $\overline{CS}$ should never be active simultaneously. The data transfer modes are described individually below:

Programmed I/O Transfers

Programmed I/O is the most primitive form of data transfer. The $\overline{REQ}$ and $\overline{ACK}$ handshake signals are individually monitored and asserted by reading and writing the appropriate register bits. This type of transfer is normally used when transferring small blocks of data such as command blocks or message and status bytes.

An Initiator send operation would begin by setting the $\overline{C/D}$, $\overline{I/O}$, and $\overline{MSG}$ bits in the Target Command register to the correct state so that a phase match exists. In addition to the phase match condition, it is necessary for the ASSERT DATA BUS bit (port 1, bit 0) to be TRUE and the received $\overline{I/O}$ signal to be FALSE for the CA53C80 to send data.

For each transfer, the data is loaded into the Output Data register (port 0). The CPU then waits for the REQ bit (port 4, bit 5) to become active. Once REQ goes active, the PHASE MATCH bit (port 5, bit 3) is checked and the ASSERT ACK bit (port 1, bit 4) is set. The REQ bit is sampled until it becomes FALSE and the CPU resets the ASSERT ACK bit to complete the transfer.

Normal DMA Mode

DMA transfers are normally used for large block transfers. The SCSI chip outputs a DMA request (DRQ) whenever it is ready for a byte transfer. External DMA logic uses this DRQ signal to generate DACK and an IOR or an IOW pulse to the CA53C80. DRQ goes inactive when DACK is asserted and DACK goes inactive some time after the minimum read or write pulse width. This process is repeated for every byte. For this mode, DACK should not be allowed to cycle unless a transfer is taking place.

Block Mode DMA

Some popular DMA controllers such as the CA82C37A provide a Block Mode DMA transfer. This type of transfer allows the DMA controller to transfer blocks of data without relinquishing the use of the Data Bus to the CPU after each byte is transferred; thus, faster transfer rates are achieved by eliminating the repetitive access and release of the CPU Bus.

If the BLOCK MODE DMA bit (port 2, bit 7) is active, the CA53C80 will begin the transfer by asserting DRQ. The DMA controller then asserts DACK for the remainder of the block transfer. DRQ goes inactive for the duration of the transfer. The READY output is used to control the transfer rate.

Non-Block Mode DMA transfers end when DACK goes FALSE, whereas Block Mode transfers end when IOR or IOW becomes inactive. Since this is the case, DMA transfers may be started sooner in a Block Mode transfer.

To obtain optimum performance in Block Mode operation, the DMA logic may optionally use the normal DMA mode interlocking handshake. READY is still available to throttle the DMA transfer, but DRQ is 30 to 40 ns faster than READY and may be used to start the cycle sooner.

The methods described under *Halting a DMA Operation* apply for all DMA operations.

Pseudo DMA Mode

To avoid the tedium of monitoring and asserting the request/acknowledge handshake signals for programmed I/O transfers, the system may be designed to implement a pseudo DMA mode. This mode is implemented by programming the CA53C80 to operate in the DMA mode, but using the CPU to emulate the DMA handshake. DRQ may be detected by polling the DMA REQUEST bit (bit 6) in the Bus and Status register (port 5), by sampling the signal through an external port or by using it to generate a CPU interrupt. Once DRQ is detected, the CPU can perform a read or write data transfer. This CPU read/write is externally decoded to generate the appropriate DACK and IOR or IOW signals.

Often, external decoding logic is necessary to generate the CA53C80 CS signal. This same logic may be used to generate DACK at no extra system cost and provide an increased performance in programmed I/O transfers.

HALTING A DMA OPERATION

The EOP signal is not the only way to halt a DMA transfer. A bus phase mismatch or a reset of the DMA MODE bit (port 2, bit 1) can also terminate a DMA cycle for the current bus phase. These methods are described below:

Using the EOP Signal

If EOP is used, it should be asserted for at least 100 ns while DACK and IOR or IOW are simultaneously active. Note, however, that if IOR or IOW is not active an interrupt will be generated, but the DMA activity will continue. The EOP signal does not reset the DMA MODE bit. Since the EOP signal can occur during the last byte sent to the Output Data register (Port 0), the REQ and ACK signals should be monitored to ensure that the last byte has been transferred.

Bus Phase Mismatch Interrupt

A bus phase mismatch interrupt may be used to halt the transfer if operating as an Initiator. This method frees the host from maintaining a data length counter and frees the DMA logic from providing the EOP signal. If performing an Initiator send operation, the CA53C80 requires DACK to cycle before ACK goes *high*. Since phase changes cannot occur if ACK is *low*, either DACK must be cycled after the last byte is sent or the DMA MODE bit must be reset in order to receive the phase mismatch interrupt.

Resetting the DMA MODE Bit

A DMA operation may be halted at any time by resetting the DMA MODE bit. It is recommended that the DMA MODE bit be reset after receiving an EOP or bus phase-mismatch interrupt. The DMA MODE bit must then be set before writing any of the start DMA registers for subsequent bus phases.

If the DMA MODE bit reset method is used instead of the EOP for Target role operation, care must be taken to reset this bit at the proper time. If receiving data as a Target device, the DMA MODE bit must be reset after the last DRQ is received and before DACK is asserted to prevent an additional REQ from occurring. Resetting this bit causes DRQ to go *low*. However, the last byte received remains in the Input Data register and may be obtained by performing a normal CPU read or by cycling DACK and IOR. Usually, EOP is easier to use when operating as a Target device.