Product Overview

About This Document

This document contains information on the 9802A, a new product. The specifications and other information herein are subject to change without notice. Please contact Stac Electronics to obtain the latest information on the 9802A before implementing any design that includes this product.

The Product Overview touches on various applications for the 9802A. Detailed pin and register descriptions follow the overview. Examples of ECC and DMA operations are given after the register descriptions, and electrical specifications and physical dimensions are provided in the back of the document.

The conventions used in this data sheet are as follows: register names are written with the first letter of each word capitalized (Byte Increment Register); bit names are written in small capitals (COMPARE MODE ENABLE bit); pins and signals are in all capital letters in a smaller Times Roman font ($\overline{\text{CSI}}$); and references to main sections of this document are in italics (*Electrical Specifications*). Standard abbreviations are used wherever possible.

Applications

The 9802A Buffer Manager controls data flow in host-resident or embedded tape controller operations. It is a highly integrated device, suited to high performance tape controller architectures. In the typical embedded tape controller application, the 9802A is used in conjunction with a tape formatter, a microprocessor, a buffer memory (DRAM), and a host. It will support the Small Computer Systems Interface (SCSI) or a Quarter-Inch Cartridge (QIC™) interface such as the QIC-02.

In Figure 1-1 below, the 9802A is shown in an embedded quarter-inch cartridge tape controller application using the Stac 9820 QIC Formatter. This application illustrates the peripheral access feature of the 9802A. Using peripheral access, the SCSI chip appears to have separate busses for DMA and MPU data transfers. Actually, the MPU accesses the registers of the SCSI chip through the 9802A by using the single 8-bit data bus between the 9802A and the SCSI control logic.

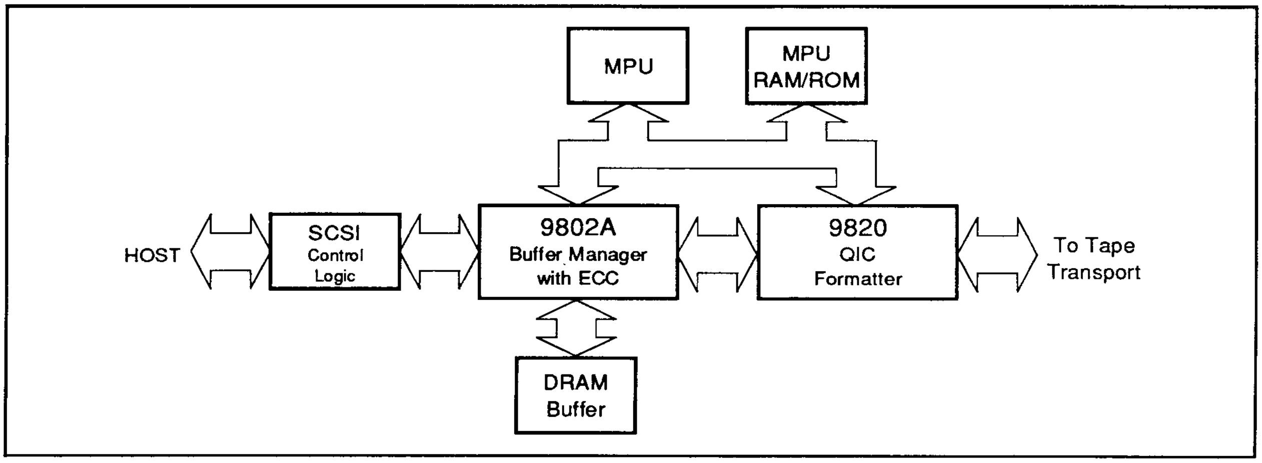


Figure 1-1. Embedded QIC Controller

QIC™ is a trademark of Quarter-Inch Cartridge Drive Standards, Inc.

The MPU sends requests for data from the SCSI via the 9802A, and the 9802A permits the requests when the 9802A has accessed the SCSI chip. When the SCSI is ready to transfer data and the 9802A has acknowledged the SCSI ready signal, the data is sent via the 9802A to the DRAM Buffer, where it may be stored in matrices which match the physical tape formats.

While data transfers are continuing into the 9802A, ECC may be performed on data already in the buffer. Thus ECC does not take up valuable DMA bandwidth, and is not a limiting factor in the rate of transfer.

All of the control signals necessary for DRAM interfacing are provided by the 9802A. These include a 24-bit multiplexed address bus, an 8-bit data bus with parity generation and checking logic, two row address strobe signals for up to 2 banks of DRAM, a column address strobe, and a write enable signal.

The application shown in Figure 1-2 below is very much like the example described above, but it is for a digital audio tape (DAT) drive rather than a QIC tape drive. It makes use of the peripheral access capabilities of the 9802A and takes advantage of DMA channel 3 as an additional tape formatter channel.

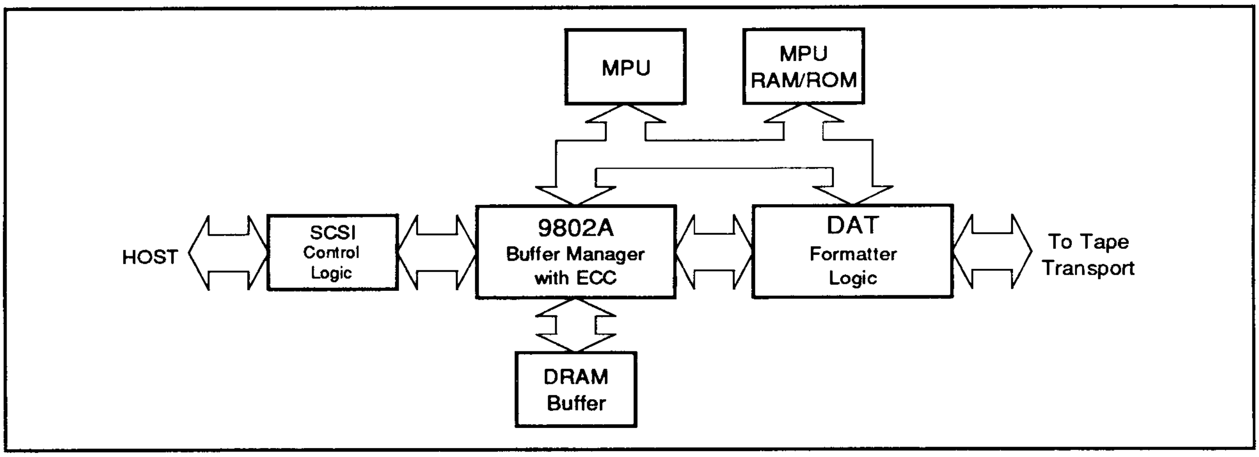


Figure 1-2. Embedded DAT Controller

Operational Overview

The functions of the 9802A fall into four categories: DMA Interface, ECC Processor, MPU Buffer Access, and DRAM Interface. Refer to Figure 1-3 below. The three DMA channels and the ECC Processor can be prearmed, which means that their registers can be written even while DMA or ECC operations are in progress. Prearming is described in detail in *Register Descriptions*.

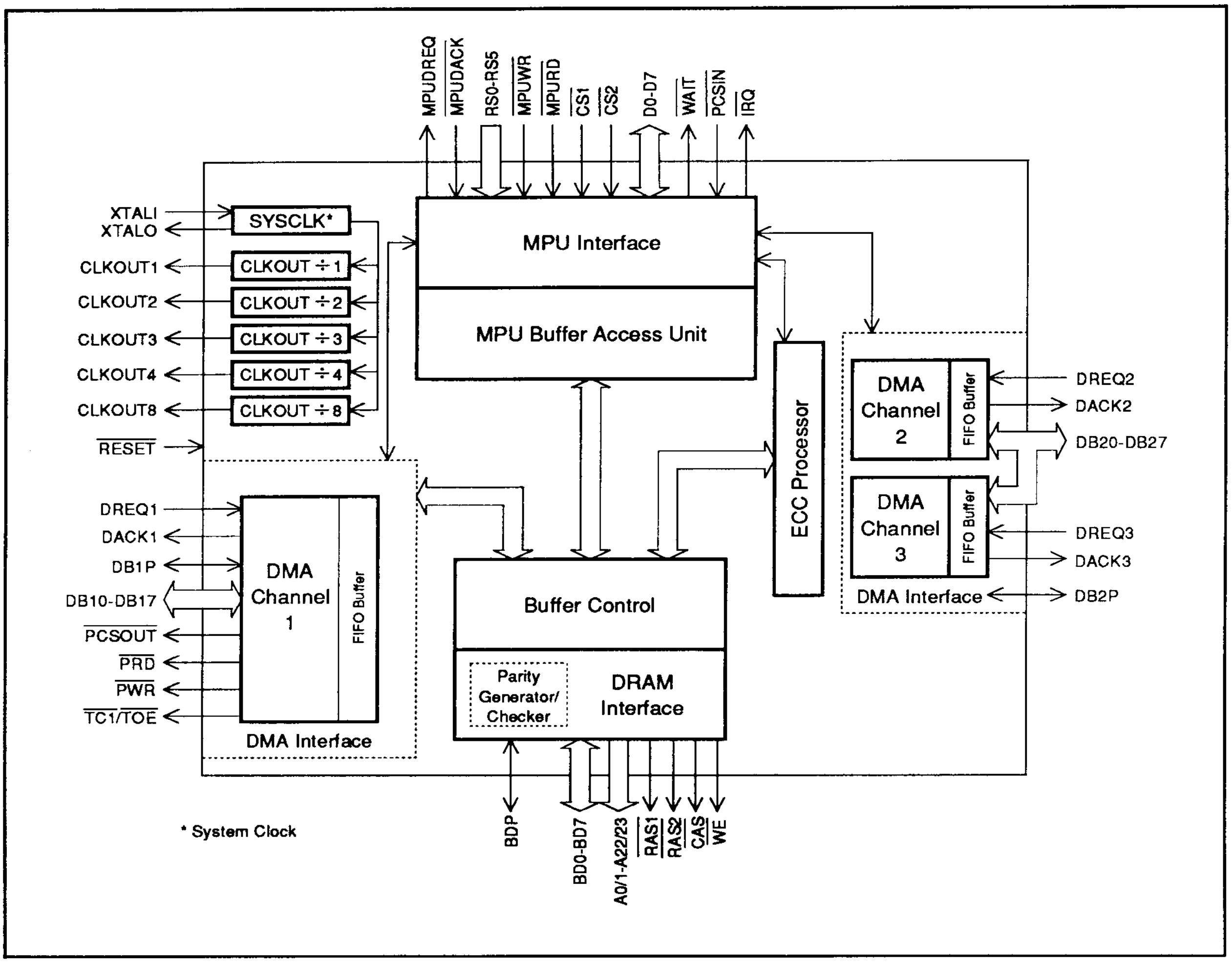


Figure 1-3. 9802A Block Diagram

DMA Interface

The 9802A has three DMA channels, each containing programmable Command, Address, and Transfer Length registers. The address increment between byte transfers and rows (in matrix mode) is also programmable. An interrupt can be generated 0, 2, 4, or 8 bytes before the end of a transfer. The DMA data busses can be connected to any device, but each has been optimized for a specific application. DMA channel 1 has a 7-byte internal FIFO buffer and is useful for high-speed interfaces such as the SCSI. DMA channel 2 has a 1-byte internal FIFO buffer and is best suited for slower peripherals such as tape formatters. DMA channel 3 also has a 1-byte internal FIFO buffer, and can be used as a second formatter channel or to interface to another peripheral, such as the Stac 9703 Data Compression Coprocessor. Additional request and acknowledge signals allow DMA Channels 2 and 3 to operate at the same time over the DMA data bus. A byte-by-byte compare function validates data and can generate an interrupt on compare failure. Parity checking is provided for all three DMA Channels, along with a parity error flag. Refer to *Register Descriptions* for greater detail.

The 9802A can make DMA transfers in linear or matrix mode. Matrix mode allows data to be transferred into the buffer in a logical matrix that contains a field for the user data and also contains a field for 1) control information, as required by QIC formats, or 2) system data, as required by the DAT formats. The row size and column size of both the data and the control/system fields are programmable.

The data is transferred to tape leaving the required space for the control field. The control field can then be filled in later by the MPU. An entire QIC 525 frame can be transferred in a single DMA operation.

The examples below illustrate the two methods used with the 9802A to transfer data into the buffer in matrix mode: row oriented transfers (for QIC formats) and column oriented transfers (for DAT formats). The first two examples, shown in Figure 1-4, illustrate how data is laid byte by byte into the user data field. The two examples which follow, Figures 1-5 and 1-6, show the entire QIC and DAT matrices (minus the syndrome field, which does not get transferred, but is used internally by the ECC Processor in the 9802A chip).

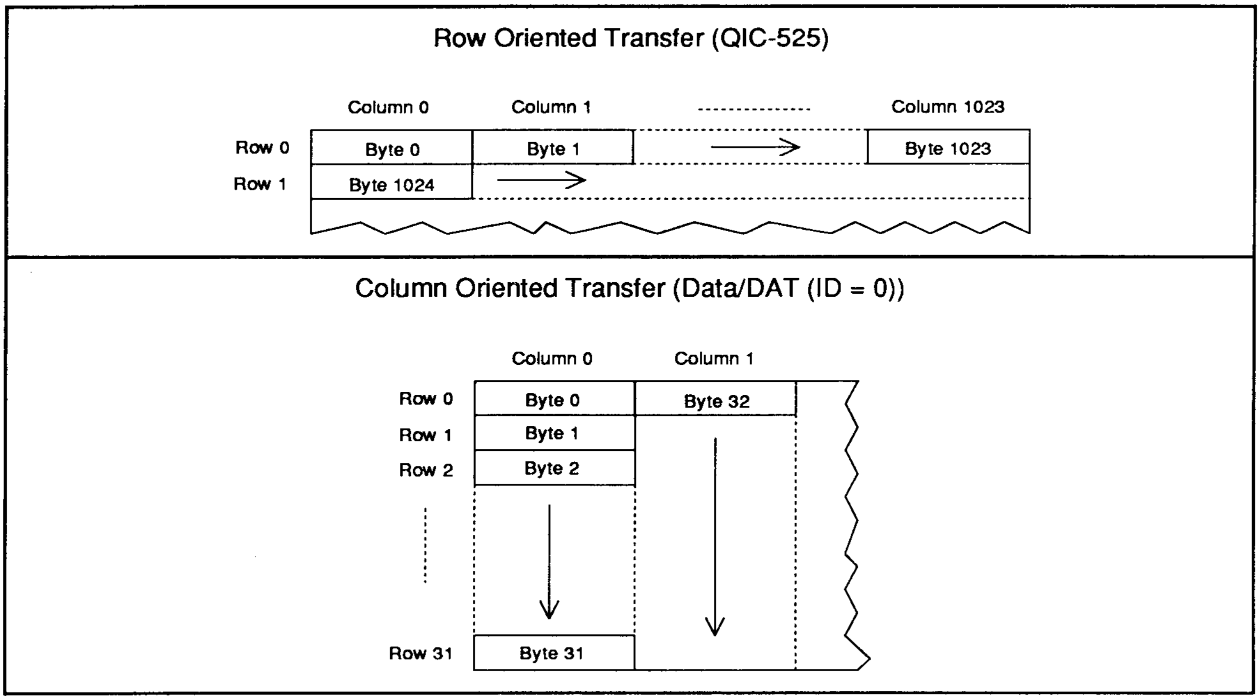


Figure 1-4. Transferring Data into the User Data Field by Bytes

Figure 1-5 illustrates a QIC-525 application using row oriented transfer. Data is transferred into the buffer a byte at a time, starting with the "start address" and proceeding in the same row. Once the end of the first user data field row is reached, the address automatically advances to the beginning of the next row. Each row of the data field is filled without continuing into that portion of the row that is reserved for the control field.

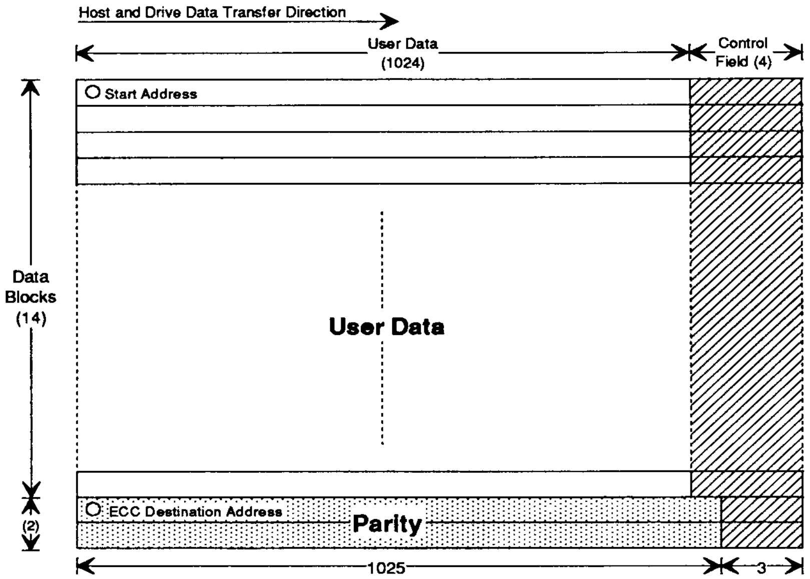


Figure 1-5. Row Oriented Matrix Mode Transfer — QIC-525 Frame

The next example shows a digital audio tape application using column oriented transfer for host data. (See Figure 1-6.) Data is transferred from the host to the buffer a byte at a time, starting at the starting address and proceeding down the first column. The column is filled, and then the address automatically advances to the beginning of the next column. Each column, in turn, is filled before the next column is begun. The data fills only the user data field and does not encroach on the parity field.

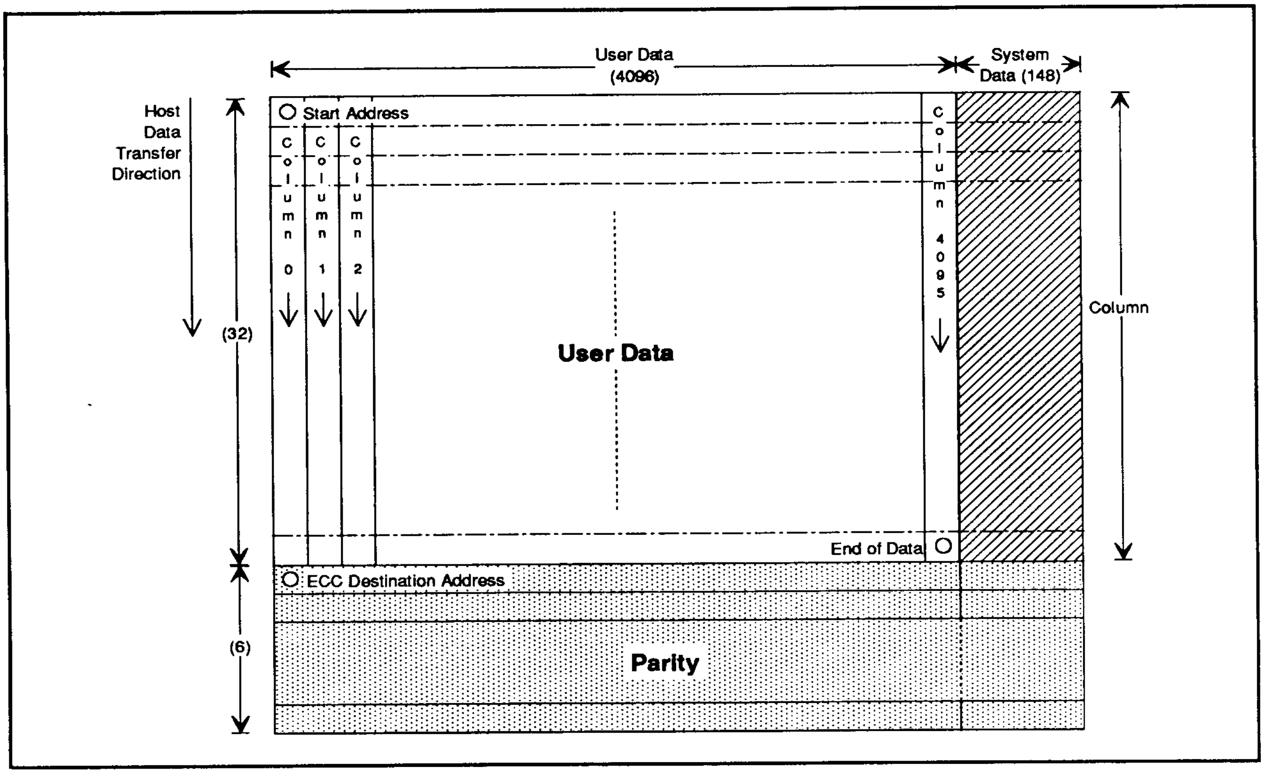


Figure 1-6. Column-Oriented Matrix-Mode Host-Data Transfer — Data/DAT Group (ID = 0)

As in the QIC example above, the drive data-transfer direction is still along the rows. (See Figure 1-7.)

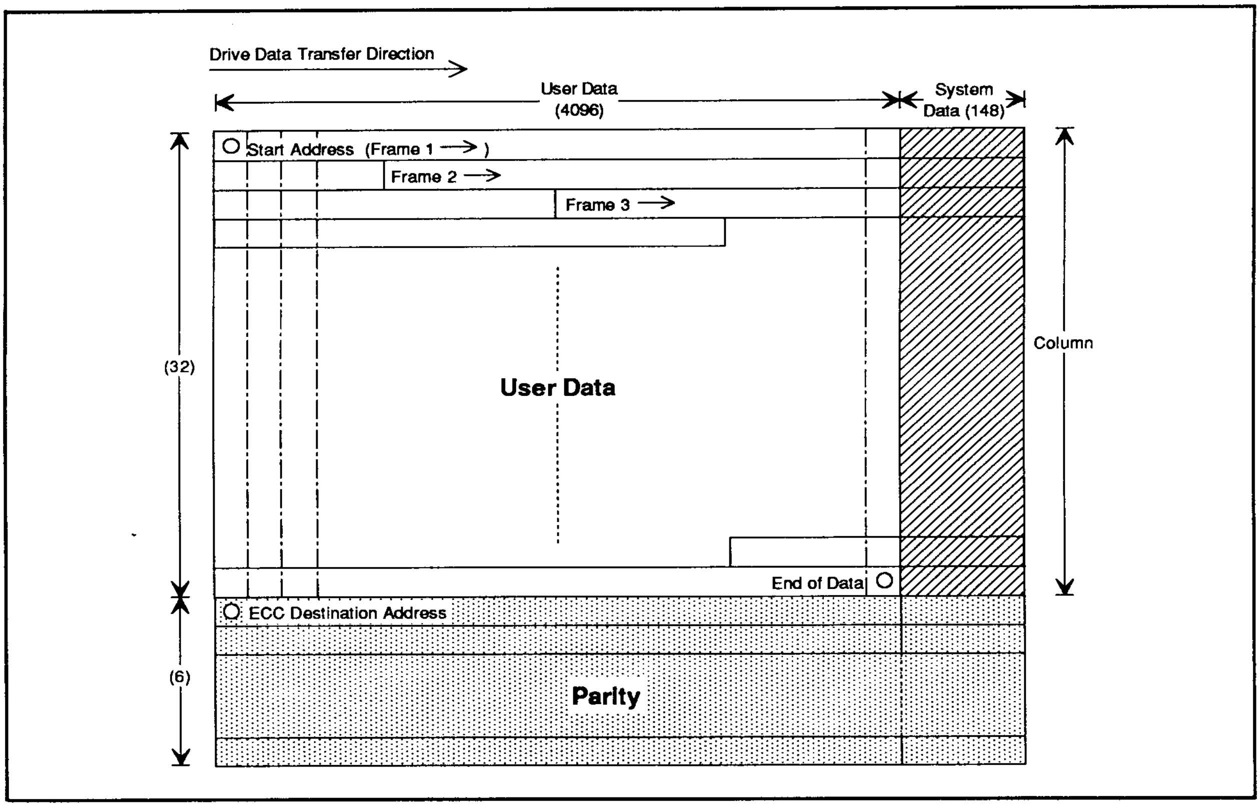


Figure 1-7. Column-Oriented Matrix-Mode Drive-Data Transfer — Data/DAT Group (ID = 0))

ECC Processor

The 9802A implements advanced Reed-Solomon error correction codes (ECC) which adapt to QIC and DAT tape formats. Reed-Solomon codes up to redundancy 8 are supported. The MPU can program the GF(256) field representation and the Reed-Solomon code generator polynomial as well as the starting addresses of the data, syndrome and parity blocks. For all ECC operations, a byte can be read or written approximately every 9 clock cycles regardless of the redundancy.

The ECC processor performs its calculations on an entire matrix of data. A matrix is specified by the number of rows and columns of data. Using QIC terminology, a matrix of data is an entire frame of data consisting of blocks. For example, for the QIC-525 format, a matrix might consist of 14 rows (data blocks) and 1025 columns (bytes in a data block covered by ECC). Using DAT terminology, a matrix of data is an entire group of data consisting of frames. The ECC processor performs encode (parity or syndrome generation) or error correction operations, and can be programmed to interrupt the processor upon completion of the operation.

The ECC processor can either write its results to the destination address or XOR them with the value already there. Typically, writing is used for parity and syndrome generation and XOR for error correction. The ECC processor is useful in QIC applications to correct re-ordering associated with rewritten and unreadable blocks. (Block moves are illustrated in the *ECC Examples* section.)

For a QIC 525 frame or a Data/DAT group, data is read starting at the ECC source address ("Start Address" on Figures 1-5, 1-6 and 1-7) and the parity blocks are written starting at the ECC destination address. The resulting parity field in the matrix for a QIC 525 frame is illustrated in Figure 1-5, above. The resulting parity field in a Data/DAT group matrix is illustrated in Figures 1-6 and 1-7, above.

The syndrome field in each case is the same size as the parity field, but is not a continuous field, and is not transferred in or out of the 9802A.

MPU Interface and MPU Buffer Access

The MPU Interface handles the signals necessary to coordinate the activities of the MPU and the 9802A. The $\overline{\text{CS1}}$, CS2, $\overline{\text{MPURD}}$ and $\overline{\text{MPUWR}}$, along with the Register Select inputs RS5-RS0, allow the MPU to access the 9802A control registers via the MPU Data Bus (D7-D0.

The MPU has access to the buffer via the MPU Buffer Command Register, the MPU Buffer Address Register, and the MPU Buffer Data Register. The MPUDREQ and $\overline{\text{MPUDACK}}$ pins allow the MPU to transfer data to and from the buffer using DMA. The MPU Buffer Command Register controls the direction of transfer, the amount by which to increment or decrement the address, and the continue mode variables (pre-fetching). When continue mode is enabled, the next byte is read from or written to the buffer after every access of the MPU Buffer Data Register. The address is updated after fetching or writing the byte. If continue mode is disabled, the buffer is read from or written to when the MPU Buffer Command Register is written. An MPU data ready indication is provided in the Status Register.

When CS2 and $\overline{\text{PCSIN}}$ are activated, $\overline{\text{PCSOUT}}$ and the $\overline{\text{WAIT}}$ signal allow data to pass data through to the peripheral connected to DMA Channel 1. The $\overline{\text{IRQ}}$ signal can be used to coordinate the transfer of interrupt vectors to the MPU. The vector returned can be used to determine the source of the interrupt.

DRAM Interface and Buffer Control

The 9802A provides all of the control signals necessary for DRAM interfacing. These include a 24-bit multiplexed address bus, an 8-bit data bus with parity, two row address strobe signals for up to 2 banks of DRAM, a column address strobe, and a write enable signal. Buffer access is requested through external signals (DMA channels) and internal signals (refresh, MPU, or ECC). Access is granted to the buffer according to one of two priority schemes. The priority scheme is selected in the Configuration Register.

The number of clock cycles used for DRAM accesses is programmed in the Configuration Register, allowing for the use of slower RAMs when buffer bandwidth is not critical. Refer to Interfacing to the DRAM Buffer in *Interfacing with the 9802A*.

Signal Descriptions

For this description, the 9802A I/O pins are divided into four groups: MPU Interface, DMA Interface, Buffer Interface, and Miscellaneous Control.

MPU Interface

Chip Selects ($\overline{\text{CS1}}$, CS2) (Pin Numbers 65, 68)

When the Chips Select signals are both active, the MPU can access the registers of the 9802A. These pins are typically connected to the outputs of an address decoder.

Register Selects (RS0 - RS5) (58–63)

The Register Select signals determine which of the registers will be accessed by the MPU when the 9802A is selected (when both chip selects are active). During a write cycle these signals are latched by the chip on the leading edge of the last active chip select signal. These pins are typically connected to the six least significant address pins of the MPU.

MPU Write ($\overline{\text{MPUWR}}$) (64)

The MPU Write signal performs two different functions depending on the type of MPU used. For processors with one R/$\overline{\text{W}}$ signal, the MPU Write signal serves as the R/$\overline{\text{W}}$ input to the 9802A. The MPU generates this signal to select either a read or a write cycle. When the 9802A is configured for a single R/$\overline{\text{W}}$ signal, this pin can be connected to the R/$\overline{\text{W}}$ pin of the MPU.

For processors with separate $\overline{\text{RD}}$ and $\overline{\text{WR}}$ signals, the MPU Write signal serves as the $\overline{\text{MPUWR}}$ input into the 9802A. The MPU asserts this signal to select a write cycle. This signal must be stable before both chip selects are active simultaneously. When the 9802A is configured for separate $\overline{\text{RD}}$ and $\overline{\text{WR}}$ signals, this pin can be connected to the $\overline{\text{WR}}$ pin of the MPU.

MPU Read ($\overline{\text{MPURD}}$) (38)

Processors that have separate $\overline{\text{RD}}$ and $\overline{\text{WR}}$ output signals generate the MPU Read signal to select a read cycle. When using a processor that has only one R/$\overline{\text{W}}$ signal, $\overline{\text{MPURD}}$ must be tied to ground.

MPU Data Bus (D7 - D0) (72–79)

The MPU Data Bus pins connect to the MPU data bus, over which the MPU can access the registers of the 9802A. Data is latched at the end of an MPU write cycle. Data is driven when $\overline{\text{MPUWR}}$ is high, $\overline{\text{MPURD}}$ is low, CS2 is high, and $\overline{\text{CS1}}$ or $\overline{\text{PCSIN}}$ is low. These pins are typically connected to the data bus pins of the MPU.

Interrupt Request ($\overline{\text{IRQ}}$) (80)

The Interrupt Request signal is output by the 9802A and is typically connected to one of the MPU's interrupt pins. This pin is an open drain output and can be wire-OR'd with other open drain outputs and requires an external pull-up resistor.

Wait Signal ($\overline{\text{WAIT}}$) (71)

The $\overline{\text{WAIT}}$ signal, output by the 9802A, indicates that the MPU must wait before attempting certain read and write accesses, described below. $\overline{\text{WAIT}}$ should be connected either to the MPU Wait input or to a circuit for stretching the MPU clock. This pin is an open drain output that can be wire-OR'd with other open drain outputs. It requires an external pull-up resistor.

When the MPU requests a peripheral access (when the MPU asserts $\overline{\text{PCSIN}}$ and CS2), the 9802A asserts $\overline{\text{WAIT}}$. For processors with one R/$\overline{\text{W}}$ output, the peripheral access cycle will begin as soon as chip selects $\overline{\text{PCSIN}}$ and CS2 are asserted. For processors with separate $\overline{\text{RD}}$ and $\overline{\text{WR}}$ outputs, the peripheral access cycle will begin as soon as $\overline{\text{MPUWR}}$ or $\overline{\text{MPURD}}$ is active and $\overline{\text{PCSIN}}$ and CS2 are active. For both processors, $\overline{\text{WAIT}}$ will be removed shortly after access to the peripheral is granted. Refer to *Interfacing with the 9802A* and *Electrical Specifications* for more information on peripheral access.

During MPU buffer accesses, the 9802A asserts $\overline{\text{WAIT}}$ whenever the MPU tries to read or write the MPU Buffer Data Register while the MPU DATA READY bit in the Status Register is clear. $\overline{\text{WAIT}}$ will be removed as soon as the MPU DATA READY bit is set by the 9802A. Refer to the description of the MPU Buffer Data Register for more information.

When the 9802A is selected and the MPU tries to access a RAM-based register, the 9802A asserts $\overline{\text{WAIT}}$. If the MPU is trying to write a RAM-based register, $\overline{\text{WAIT}}$ is removed immediately. If the MPU is trying to read a RAM-based register, $\overline{\text{WAIT}}$ is removed after the 9802A drives the data onto the MPU Data Bus. Refer to Table 1, Register Groups and Addresses, in *Register Descriptions* for addresses of RAM-based registers.

MPU DMA Request (MPUDREQ) (70)

The MPU DMA Request output signal is used for DMA transfers using the MPU Access Unit across the MPU data bus. MPUDREQ always has the same state as the MPU DATA READY bit indicating whether the 9802A is ready to transfer a byte of data between the MPU and the MPU Buffer Data Register. Asserting $\overline{\text{MPUDACK}}$ or accessing the MPU Buffer Data Register clears the MPU DATA READY bit and forces this signal inactive. This signal is active high.

MPU DMA Acknowledge ($\overline{\text{MPUDACK}}$) (39)

The MPU generates the MPU DMA Acknowledge signal to acknowledge the MPU DMA Request signal generated by the 9802A. Asserting this signal selects the MPU Buffer Data Register, no matter which register is selected by the Register Select signals. The MPU Buffer Data Register is then read or written as determined by the DIRECTION SELECT bit in the MPU Buffer Command Register. Note that the values of the $\overline{\text{MPUWR}}$ and the $\overline{\text{MPURD}}$ are ignored.

DMA Channel Interface

DMA Channel 1 Data Bus (DB17 - DB10) (82–88, 91)

The DMA Channel 1 Data Bus pins connect channel 1 to the DMA data bus. Data coming into the 9802A is latched on the trailing edge of DACK1. Data leaves the 9802A while DACK1 is active. DMA channel 1 is a high-speed channel with a 7-byte internal FIFO buffer.

DMA Channel 1 Parity (DB1P) (1)

The DMA Channel 1 Parity signal is the parity bit for the DMA Channel 1 Data Bus. During a write, odd parity is generated on every DMA transfer out of the 9802A DMA Channel 1, such that the sum of the DB1P and the DB10–DB17 states will be an odd number. Parity errors are reported via the Interrupt Status Register and $\overline{\text{IRQ}}$ when enabled via the DMA Compare/Link Register.

DMA Channel 1 Request (DREQ1) (92)

The DMA Channel 1 Request signal is an input to the 9802A and indicates that the device connected to DMA channel 1 is requesting a DMA transfer. The polarity of the DREQ1 pin is programmable through the DMA Configuration Register.

DMA Channel 1 Acknowledge (DACK1) (93)

The DMA Channel 1 Acknowledge signal output by the 9802A indicates that the 9802A is ready to transfer data over DMA channel 1. DACK1 signal polarity is programmable through the DMA Configuration Register. The signal length is programmable through the DMA Handshake Register. This pin can be put into a high-impedance state via the DMA Handshake Register. On reset, this pin is in a high-impedance state.

DMA Channel 1 Terminal Count/Trigger on Error ($\overline{\text{TC1}}/\overline{\text{TOE}}$) (96)

The TC1/TOE SELECT bit in the DMA Compare/Link Register determines whether this $\overline{\text{TC1}}/\overline{\text{TOE}}$ output is the DMA Channel 1 Terminal Count signal ($\overline{\text{TC1}}$) or the Trigger on Error signal ($\overline{\text{TOE}}$). When the TC1/TOE SELECT bit is clear and DACK1 is active, the DMA Channel 1 Terminal Count signal is asserted on the last transfer of a DMA operation. The last transition of DACK1 causes $\overline{\text{TC1}}$ to go active. Refer to DMA Interface Timing in the *Electrical Specifications* for complete timing information.

When the TC1/TOE SELECT bit is set and the 9802A is in compare mode (when the COMPARE MODE ENABLE bit in the DMA Command Register is set), $\overline{\text{TOE}}$ flags compare failures on data transferred into the 9802A. After a byte is transferred via DMA into the 9802A, it is compared with the corresponding byte in the buffer RAM. If a compare failure is detected, $\overline{\text{TOE}}$ is asserted before the next byte is transferred into the 9802A. This signal can be used with any or all DMA channels. Refer to the COMPARE MODE ENABLE bit for more information on compare mode, and to DMA Interface Timing in *Electrical Specifications* for complete timing information.

Peripheral Read ($\overline{\text{PRD}}$) (94)

The Peripheral Read signal is active during data transfers (DMA and peripheral access) into the 9802A over DMA channel 1. During DMA transfers into the 9802A, this signal is active while DACK1 is active. During a peripheral access, this signal is asserted when $\overline{\text{PCSOUT}}$ is low and when the MPU performs a read access of the peripheral ($\overline{\text{PCSIN}}$ low, CS2 high, $\overline{\text{MPUWR}}$ high, and $\overline{\text{MPURD}}$ low). Refer to *Electrical Specifications* for complete timing information.

Peripheral Write ($\overline{\text{PWR}}$) (95)

The Peripheral Write signal is active during data transfers (DMA and peripheral access) out of the 9802A over DMA channel 1. During DMA transfers out of the 9802A, this signal is active while DACK1 is active. During a peripheral access, this signal is asserted when $\overline{\text{PCSOUT}}$ is low and when the MPU is performing a write access of the peripheral ($\overline{\text{PCSIN}}$ low, CS2 high, and $\overline{\text{MPUWR}}$ low). Refer to *Electrical Specifications* for complete timing information.

Peripheral Chip Select In (PCSIN) (69)

When CS2 is active, the MPU asserts the Peripheral Chip Select In signal to access the peripheral connected to DMA channel 1. WAIT will immediately become active and will stay active until the peripheral is available. When the peripheral becomes available, WAIT goes inactive and PCSOUT goes active, selecting the peripheral and allowing the data transfer.

Peripheral Chip Select Out (PCSOUT) (97)

PCSOUT is output by the 9802A to select the peripheral connected to DMA channel 1 for an I/O transfer to or from the MPU. This pin is active after PCSIN and CS2 are asserted and while DACK1 is inactive. PCSOUT should be connected to the chip select input of the peripheral chip.

DMA Channel 2/3 Data Bus (DB27 - DB20) (9–13, 16–18)

The DMA Channel 2/3 Data Bus pins connect channels 2 and 3 to the DMA data bus. Data coming into the 9802A is latched onto the trailing edge of DACK2/DACK3. Data leaves the 9802A while DACK2/DACK3 is active. Both are low-speed channels with a 1-byte internal FIFO buffer.

DMA Channel 2/3 Parity (DB2P) (100)

The DMA Channel 2/3 Parity pin provides parity for the DMA Channel 2/3 Data Bus. During a write, odd parity is generated on every DMA transfer out of the 9802A DMA Channel 1, such that the sum of the DB2P and the DB27–DB20 states will be an odd number. Parity errors are reported via the Interrupt Status Register and IRQ when enabled via the DMA Compare/Link Register.

DMA Channel 2 Request (DREQ2) (19)

The DMA Channel 2 Request signal input to the 9802A indicates that the device connected to DMA channel 2 is requesting a DMA transfer. The polarity of the DREQ2 pin is programmable through the DMA Configuration Register.

DMA Channel 2 Acknowledge (DACK2) (20)

The DMA Channel 2 Acknowledge signal output by the 9802A indicates that the 9802A is ready to transfer data over DMA channel 2. DACK2 signal polarity can be programmed through the DMA Configuration Register. The maximum signal length can be programmed via the DMA Handshake Configuration Register. This pin can also be put into a high-impedance state via the DMA Handshake Configuration Register. On reset, this pin is in a high-impedance state.

DMA Channel 3 Request (DREQ3) (21)

The DMA Channel 3 Request signal is an input to the 9802A and indicates that the device connected to DMA channel 3 is requesting a DMA transfer. The polarity of the DREQ3 pin can be programmed through the DMA Configuration Register.

DMA Channel 3 Acknowledge (DACK3) (22)

The DMA Channel 3 Acknowledge signal output by the 9802A indicates that the 9802A is ready to transfer data over DMA channel 3. DACK3 signal polarity can be programmed through the DMA Configuration Register. The maximum signal length can be programmed via the DMA Handshake Configuration Register. This pin can also be put into a high-impedance state via the DMA Handshake Configuration Register. On reset, this pin is in a high-impedance state.

Buffer Interface

Buffer Address Bus (A1/0 - A23/22) (42–46, 49–51, 54–57)

The Buffer Address Bus pins connect to the buffer address bus. The 24-bit address is multiplexed on these 12 pins with the even address bits output as the column address (latched on $\overline{\mathrm{CAS}}$ going low) and the odd address bits output as the row address (latched on $\overline{\mathrm{RAS1}}$ and $\overline{\mathrm{RAS2}}$ going low). A 10-bit refresh address is output on these pins during a $\overline{\mathrm{RAS}}$ only refresh cycle. A20/21 and A22/23 are high during a refresh cycle.

Buffer Data Bus (BD7 - BD0) (23–25, 28–32)

The Buffer Data Bus pins connect to the buffer RAM data bus. During a write to the buffer, these pins are driven by the 9802A. During a read, the data on these pins is latched by the 9802A before $\overline{\mathrm{CAS}}$ goes inactive. When $\overline{\mathrm{RAS1}}$ and $\overline{\mathrm{RAS2}}$ are inactive, these pins are driven by the 9802A. Refer to *Electrical Specifications* for more complete timing information.

Buffer Data Bus Parity (BDP) (33)

The Buffer Data Bus Parity pin provides parity for the buffer RAM. During a write to the buffer RAM, this pin is driven with the proper value for odd parity. Odd parity means that the sum of the BDP and the BD0-BD7 states will be an odd number. During a read of the buffer RAM, this signal is checked to be sure the data has odd parity. Parity errors are reported via the Interrupt Status Register and $\overline{\mathrm{IRQ}}$ when enabled via the Configuration Register.

Buffer Write Enable ($\overline{\mathrm{WE}}$) (37)

The Buffer Write Enable signal is an output used to indicate that a write cycle is being performed on the buffer. It is active before $\overline{\mathrm{CAS}}$ goes active, implementing an early write cycle on the buffer DRAM. It should be connected to the write enable input of the DRAM.

Buffer Column Address Strobe ($\overline{\mathrm{CAS}}$) (36)

The Buffer Column Address Strobe signal indicates the column address is present on the buffer address pins and may be latched. This signal should be connected to the $\overline{\mathrm{CAS}}$ input of the DRAM.

Buffer Bank Row Address Strobe ($\overline{\mathrm{RAS1}}$, $\overline{\mathrm{RAS2}}$) (34, 35)

The Buffer Bank Row Address Strobe signals indicate that a row address is present on the buffer address pins and may be latched. Whether $\overline{\mathrm{RAS1}}$ or $\overline{\mathrm{RAS2}}$ goes active is determined by the RAM SIZE SELECT bits in the Configuration Register. The 9802A can support up to 2 banks of DRAM. To use only one bank of DRAM, connect $\overline{\mathrm{RAS1}}$ to the $\overline{\mathrm{RAS}}$ pin of the DRAM. To use two banks of DRAM, connect $\overline{\mathrm{RAS1}}$ to the $\overline{\mathrm{RAS}}$ pins of the DRAM in bank one and connect $\overline{\mathrm{RAS2}}$ to the $\overline{\mathrm{RAS}}$ pins of the DRAM in bank two. $\overline{\mathrm{RAS1}}$ and $\overline{\mathrm{RAS2}}$ are active during DRAM refresh cycles.

Miscellaneous Control

Hardware Reset ($\overline{\text{RESET}}$) (81)

The Hardware Reset signal input forces the 9802A into a known state. Note that asserting ($\overline{\text{RESET}}$) sets the MASTER RESET bit in the Configuration Register. Refer to *Register Values on Reset* for details on the value of the registers of the 9802A after $\overline{\text{RESET}}$ is asserted.

Crystal/External Clock (XTALI, XTALO) (99, 98)

The Crystal/External Clock pins provide control input for the on-chip clock oscillator circuit. A crystal or an external signal can be connected to these pins to provide input to the internal oscillator. Note that the crystal oscillator generates a signal referred to as the system clock. Stray capacitance on these two pins should be minimized. Refer to *Internal Oscillator* for recommended circuit configurations.

Buffered Clock Outputs (CLKOUT1, CLKOUT2, CLKOUT3, CLKOUT4, CLKOUT8) (8,7,6,5,4)

The Buffered Clock Output pins provide the system designer with five derivatives of the clock signal generated by the on chip crystal oscillator (system clock). CLKOUT1 is a buffered output of the system clock signal. CLKOUT2 provides a system clock frequency divided by 2. Similarly, CLKOUT3, CLKOUT4, and CLKOUT8 provide system clock frequencies divided by 3, 4, and 8 respectively. These can be used to provide clock signals to other elements of the tape controller system such as the MPU.

Register Descriptions

Overview

The registers of the 9802A have been divided into four groups: Control/Status, DMA Interface, MPU Buffer Access, and ECC. The following subsections give a detailed description of all bit fields within the 9802A register set. Table 3-1 below lists the registers and gives their associated groups.

Table 3-1. Register Groups and Addresses

RS5...RS0	RS (hex)	Register	Register Group
000000	00	Configuration	Control/Status
000001	01	Interrupt Status	Control/Status
000010	02	Status	Control/Status
000011	03	Prearm Status	Control/Status
000100	04	DMA Configuration	DMA
000101	05	DMA Handshake Configuration	DMA
000110	06	DMA Compare/Link	DMA
000111	07	ECC Byte Increment (high byte)	ECC
001000	08	ECC Byte Increment (middle byte)	ECC
001001	09	ECC Byte Increment (low byte)	ECC
001010	0A	Byte Increment (high byte)	Control/Status
001011	0B	Byte Increment (middle byte)	Control/Status
001100	0C	Byte Increment (low byte)	Control/Status
001101	0D	Row Increment (high byte)	Control/Status
001110	0E	Row Increment (middle byte)	Control/Status
001111	0F	Row Increment (low byte)	Control/Status
010000	10	reserved	
010001	11	reserved	
010010	12	*†DMA 1 Command	DMA
010011	13	*†DMA 1 Address (high byte)	DMA
010100	14	*†DMA 1 Address (middle byte)	DMA
010101	15	*†DMA 1 Address (low byte)	DMA
010110	16	*†DMA 1 Transfer Length (high byte)	DMA
010111	17	*†DMA 1 Transfer Length (low byte)	DMA
011000	18	reserved	
011001	19	reserved	
011010	1A	*†DMA 2 Command	DMA

*Note: These registers are prearmable.
†Note: These registers are RAM based.

continued

Register Groups and Addresses (continued)

RS5...RS0	RS (hex)	Register	Register Group
011011	1B	*†DMA 2 Address (high byte)	DMA
011100	1C	*†DMA 2 Address (middle byte)	DMA
011101	1D	*†DMA 2 Address (low byte)	DMA
011110	1E	*†DMA 2 Transfer Length (high byte)	DMA
011111	1F	*†DMA 2 Transfer Length (low byte)	DMA
100000	20	reserved	
100001	21	reserved	
100010	22	*†DMA 3 Command	DMA
100011	23	*†DMA 3 Address (high byte)	DMA
100100	24	*†DMA 3 Address (middle byte)	DMA
100101	25	*†DMA 3 Address (low byte)	DMA
100110	26	*†DMA 3 Transfer Length (high byte)	DMA
100111	27	*†DMA 3 Transfer Length (low byte)	DMA
101000	28	reserved	
101001	29	reserved	
101010	2A	†MPU Buffer Command	MPU
101011	2B	†MPU Buffer Address (high byte)	MPU
101100	2C	†MPU Buffer Address (middle byte)	MPU
101101	2D	†MPU Buffer Address (low byte)	MPU
101110	2E	†reserved	
101111	2F	†reserved	
110000	30	MPU Buffer Data	MPU
110001	31	ECC Polynomial Coefficient Stack	ECC
110010	32	*†ECC Command	ECC
110011	33	*†ECC Source Address (high byte)	ECC
110100	34	*†ECC Source Address (middle byte)	ECC
110101	35	*†ECC Source Address (low byte)	ECC
110110	36	†reserved	
110111	37	†ECC Row/Column Size	ECC
111000	38	ECC GF(256) Feedback	ECC
111001	39	ECC Redundancy	ECC
111010	3A	†reserved	
111011	3B	*†ECC Destination Address (high byte)	ECC
111100	3C	*†ECC Destination Address (middle byte)	ECC
111101	3D	*†ECC Destination Address (low byte)	ECC
111110	3E	†ECC Matrix Size (high byte)	ECC
111111	3F	†ECC Matrix Size (low byte)	ECC

*Note: These registers are prearmable.
†Note: These registers are RAM based.

Prearming

Four functional units within the 9802A are prearmable: the three DMA channels and the ECC Processor. These four units are prearmed via the following prearmable registers:

Three DMA Command Registers (one per channel)
Three DMA Address Registers (one per channel)
Three DMA Transfer Length Registers (one per channel)
ECC Command Register
ECC Source Address Register
ECC Destination Address Register.

Definitions

A prearmable register has two parts. One part is called the "shadow" register and the other the "working" register. A value written to a prearmable register by the MPU is written to the shadow register. Whenever the unit is started, the content of the shadow register is automatically written to the working register. The shadow register can then be rewritten, even while the working register is being used by the 9802A.

The value in the working register is used in the current operation. The value in the shadow register is used in the next operation, which begins automatically upon completion of the current operation if the register is prearmed. Whenever a new value is written to a Command Register with the HALT bit cleared, the associated prearmable unit is considered "prearmed." Once the values in the unit's prearmable registers are transferred to the working registers, the unit is "disarmed."

The Prearm Status Register indicates which of the four units is prearmed.

Interfacing with 9802A Prearmable Units

The MPU can request an operation from a prearmable unit by writing to one of the four Command Registers. When the current operation is complete, the values in the working registers are replaced by the values in the shadow registers. However, if the unit is not prearmed, the 9802A sets the HALT bit, and the unit stops.

The MPU can abort a requested operation not yet begun (still in the shadow registers) by clearing the appropriate Prearm Status bits in the Prearm Status Register.

The MPU can abort both the pending (shadow register) and current (working register) operations by setting the HALT bit to 1 in the Command Register associated with that unit. The HALT bit takes effect immediately. To continue the halted operation, the contents of the working registers must be read and then rewritten into the shadow registers.

Table 3-2 below describes a short but typical interaction between the 9802A and the MPU involving two transfers. In the table, X1 refers to the first of two values transferred, X2 refers to the second of two values transferred, and Und means undefined.

Table 3-2. Typical Interaction Via a Prearmable Register.

Register Transfers	HALT Bit (In Command Register)	Prearm Status Bit	Comments
Shadow: Und; Working: Und	1	0	The 9802A sets the Prearm Status bit to 0.
Shadow: X1; Working: Und	0	1	1) MPU writes value X1 to shadow Command Register. 2) MPU clears HALT bit. 3) 9802A sets Prearm Status Bit
Shadow: X1; Working: X1	0	0	Clearing HALT starts 9802A; X1 is transferred to working register by 9802A.
Shadow: X2; Working: X1	0	1	1) MPU writes value X2 to shadow register. 2) 9802A sets Prearm Status bit.
Shadow: X2; Working :Und	0	1	9802A completes operation.
Shadow: X2; Working: X2	0	0	9802A transfers shadow contents to working registers and clears Prearm Status bit.
Shadow: Und; Working: Und	1	0	9802A completes operation, sets HALT bit, awaits next transfer.

Control/Status Registers

Configuration Register (RS = 00H)

The Configuration Register, diagrammed in Figure 3-1, contains bits for setting the buffer arbitration priority, the RAM size, the buffer RAM cycle time and the DRAM refresh rate. This register also contains the MASTER RESET bit and a bit to enable parity error interrupts. This register is typically written during system initialization.

7	6	5	4	3	2	1	0
Master Reset	Buffer Arbitration Priority Select	Buffer Parity Error Interrupt Enable	RAM Size Select		Buffer RAM Cycle Time Select	DRAM Refresh Rate Select	

Figure 3-1. Configuration Register

MASTER RESET - bit 7

Writing a one to the MASTER RESET bit causes the 9802A to stop all pending operations and remain in a reset state until the MPU clears this bit. Refer to *Register Values on Reset* for the definition of the reset state. Hardware reset sets this bit.

BUFFER ARBITRATION PRIORITY SELECT - bit 6

The BUFFER ARBITRATION PRIORITY SELECT bit determines the priority for buffer access. Table 3-3 below shows the two priority schemes, listing the register groups in descending order of priority. In both schemes, when multiple requests for the bus are made, the device with the highest priority will be granted access first. Both schemes are fixed priority. On hardware reset, this bit is unchanged.

Table 3-3. Buffer Access Priority

	BUFFER ARBITRATION PRIORITY SELECT	
Priority	0	1
highest	refresh	refresh
	DMA channel 2	DMA channel 2
	DMA channel 3	DMA channel 3
	MPU Buffer Access Unit	DMA channel 1
	DMA channel 1	MPU Buffer Access Unit
lowest	ECC processor	ECC processor

BUFFER PARITY ERROR INTERRUPT ENABLE - bit 5

When the BUFFER PARITY ERROR INTERRUPT ENABLE bit is clear, an interrupt will not be generated when a buffer RAM parity error is detected. When this bit is set, an interrupt will be generated when a buffer RAM parity error is detected. Note that parity is odd (see the description of the Buffer Data Bus Parity pin (BDP) in *Signal Descriptions*). Hardware reset clears this bit.

RAM SIZE SELECT - bits 4,3

The RAM SIZE SELECT bits indicate the size of the RAM chips being used for the buffer RAM. (See Table 3-4 below.) The 9802A uses this information to select which Row Address Strobe ($\overline{RAS1}$ or $\overline{RAS2}$) will be active for a given address. For example, if the RAM size is 256K, buffer addresses from 000000 to

03FFFF will cause $\overline{\text{RAS1}}$ to be active while addresses from 040000 to 07FFFF will cause $\overline{\text{RAS2}}$ to be active. This allows the use of two banks of RAM to form the buffer. Note that for use with 16 Mbit chips, external circuitry is required to force $\overline{\text{RAS}}$ on the RAM chips to be active whenever $\overline{\text{RAS1}}$ or $\overline{\text{RAS2}}$ is active. On hardware reset, these bits are unchanged.

Table 3-4. Buffer RAM Size Selection

bits 4,3	DRAM Size	Buffer Size	Bank 1 ($\overline{\text{RAS1}}$)	Bank 2 ($\overline{\text{RAS2}}$)
00	64K	64K - 128K	000000-00FFFF	010000-01FFFF
01	256K	256K - 512K	000000-03FFFF	040000-07FFFF
10	1M	1M - 2M	000000-0FFFFF	100000-1FFFFF
11	4M	4M - 8M	000000-3FFFFF	400000-7FFFFF
--	16M	16M	000000-FFFFFF	

BUFFER RAM CYCLE TIME SELECT - bit 2

The BUFFER RAM CYCLE TIME SELECT bit selects the number of 9802A system clock cycles per RAM cycle. This allows a faster system clock to be used to lower the ECC calculation time without requiring faster access RAM. When this bit is clear, each RAM cycle is 7 system clock cycles long. When this bit is set, each RAM cycle is 9 system clock cycles long. Hardware reset sets this bit.

DRAM REFRESH RATE SELECT - bits 1,0

The DRAM REFRESH RATE SELECT bits set the refresh rate for the dynamic buffer RAM. The 9802A generates a RAS-only refresh cycle (including the row address to be refreshed) every 192 to 512 system clock cycles depending on the value of these bits. The value used should allow refresh operations to occur frequently enough so that the DRAM refresh specification is met, but not so often that refresh uses a significant portion of the buffer bandwidth. Table 3-5 below indicates the refresh rate resulting from the given values for these bits.

Table 3-5. DRAM Refresh Rate

bits 1,0	Refresh Rate
00	System Clock Frequency ÷ 192
01	System Clock Frequency ÷ 256
10	System Clock Frequency ÷ 384
11	System Clock Frequency ÷ 512

For example, a 256 Kbit dynamic RAM requires 256 refresh cycles every 4 ms. With a clock frequency of 25 MHz, this equates to one refresh cycle every 390.6 system clocks. So the DRAM REFRESH RATE SELECT bits could be set to 10 to give one refresh clock every 384 clocks (or 256 refresh cycles every 3.93 ms), thereby meeting the refresh specification.

Refresh operations will not begin until hardware reset becomes inactive. Refresh operations continue unaffected by master reset. Hardware reset clears these bits.

Interrupt Status Register (RS = 01H)

The bits in the Interrupt Status Register, diagrammed in Figure 3-2, are set by the 9802A when the indicated interrupt condition occurs and the interrupt is enabled. If any bit in this register is set, IRQ will be asserted. The MPU clears each bit (except for the COMPARE FAILURE INTERRUPT bit) by writing a one to it. On hardware and master reset, all bits are cleared.

7	6	5	4	3	2	1	0
Compare Failure Interrupt	ECC Interrupt	Buffer Parity Error Interrupt	DMA 3 Interrupt	DMA 2 Interrupt	DMA 1 Interrupt	DMA Channel 2/3 Parity Error Interrupt	DMA Channel 1 Parity Error Interrupt

Figure 3-2. Interrupt Status Register

COMPARE FAILURE INTERRUPT - bit 7

The COMPARE FAILURE INTERRUPT bit is set by the 9802A if any COMPARE STATUS bits in the DMA Compare/Link Register are set. When this bit is set, there is a comparison failure on a DMA channel that is operating in comparison mode (when the COMPARE MODE ENABLE bit in the DMA Command Register is set). To clear this bit, a one must be written to all COMPARE STATUS bits that are set in the DMA Compare/Link Register.

ECC INTERRUPT - bit 6

The ECC INTERRUPT bit is set by the 9802A at the completion of an ECC operation if the ECC INTERRUPT ENABLE bit in the ECC Command Register is set.

BUFFER PARITY ERROR INTERRUPT - bit 5

If the BUFFER PARITY ERROR INTERRUPT ENABLE bit in the Configuration Register is set, the BUFFER PARITY ERROR INTERRUPT bit will be set by the 9802A when a parity error is detected on the Buffer Data Bus during a read from the DRAM. Note that parity is odd (see the description of the Buffer Data Bus Parity pin (BDP) in *Signal Descriptions*).

DMA 3 INTERRUPT - bit 4

If the DMA INTERRUPT ENABLE bit in the DMA 3 Command Register is set, the DMA 3 INTERRUPT bit will be set by the 9802A at the time determined by the IRQ TIME SELECT bits in the DMA 3 Command Register. Depending on the value of the IRQ TIME SELECT bits, this interrupt may occur from 0 to 8 bytes before the end of the DMA transfer.

DMA 2 INTERRUPT - bit 3

If the DMA INTERRUPT ENABLE bit in the DMA 2 Command Register is set, the DMA 2 INTERRUPT bit will be set by the 9802A at the time determined by the IRQ TIME SELECT bits in the DMA 2 Command Register. Depending on the value of the IRQ TIME SELECT bits, this interrupt may occur from 0 to 8 bytes before the end of the DMA transfer.

DMA 1 INTERRUPT - bit 2

If the DMA INTERRUPT ENABLE bit in the DMA 1 Command Register is set, the DMA 1 INTERRUPT bit will be set by the 9802A at the time determined by the IRQ TIME SELECT bits in the DMA 1 Command Register. Depending on the value of the IRQ TIME SELECT bits, this interrupt may occur from 0 to 8 bytes before the end of the DMA transfer.

DMA CHANNEL 2/3 PARITY ERROR INTERRUPT - bit 1

If the DMA PARITY ERROR INTERRUPT ENABLE bit in the DMA Compare/Link Register is set, the DMA CHANNEL 2/3 PARITY ERROR INTERRUPT bit will be set by the 9802A when a parity error is detected on the DMA Channel 2/3 Data Bus during a read. Note that parity is odd (see the description of the DMA Channel 2/3 Parity pin (DB2P) in *Signal Descriptions*).

DMA CHANNEL 1 PARITY ERROR INTERRUPT - bit 0

If the DMA PARITY ERROR INTERRUPT ENABLE bit in the DMA Compare/Link Register is set, the DMA CHANNEL 1 PARITY ERROR INTERRUPT bit will be set by the 9802A when a parity error is detected on the DMA Channel 1 Data Bus during a read. Note that parity is odd (see the description of the DMA Channel 1 Parity pin (DB1P) in *Signal Descriptions*).

Status Register (RS = 02H)

The Status Register, diagrammed in Figure 3-3, shows the status of operations. Each bit is set when the condition in its description below is met, and each is cleared by the MPU writing a 1 to it (except in the case of the MPU DATA READY bit which is unaffected by Status Register write operations). On hardware and master reset, all bits are cleared.

7	6	5	4	3	2	1	0
Non-Zero	ECC Operation Done	MPU Data Ready	DMA 3 Done	DMA 2 Done	DMA 1 Done	(reserved)	

Figure 3-3. Status Register

NON-ZERO - bit 7

The NON-ZERO bit is set when the ECC processor has written at least one non-zero byte to its destination. This bit must always be cleared before beginning syndrome calculations in order to easily detect errors and CRC failures.

ECC OPERATION DONE - bit 6

The ECC OPERATION DONE bit is set when the ECC processor has completed an operation (and has started on the next operation if the ECC processor was prearmed).

MPU DATA READY - bit 5

The MPU DATA READY bit is set when the MPU Buffer Data Register contains valid data from the buffer (when reading from the buffer) or when the 9802A is ready to accept write data from the MPU. Reading from or writing to the MPU Buffer Data Register clears this bit. Refer to the description of the DIRECTION SELECT bit in the MPU Buffer Command Register.

> **Note: Corruption of buffer data may result if the MPU reads or writes the MPU Buffer Data Register before checking that the MPU DATA READY bit in the Status Register is set. This is not important if the 9802A $\overline{\text{WAIT}}$ output is connected to the MPU $\overline{\text{WAIT}}$ input.**

DMA 3 DONE - bit 4

The DMA 3 DONE bit is set when DMA channel 3 has completed a transfer or comparison. When this bit is set, it indicates that DMA channel 3 is ready for the next transfer or comparison or has started on the next transfer or comparison if it was prearmed.

DMA 2 DONE - bit 3

The DMA 2 DONE bit is set when DMA channel 2 has completed a transfer or comparison. When this bit is set, it indicates that DMA channel 2 is ready for the next transfer or comparison or has started on the next transfer or comparison if it was prearmed.

DMA 1 DONE - bit 2

The DMA 1 DONE bit is set when DMA channel 1 has completed a transfer or comparison. When this bit is set, it indicates that DMA channel 1 is ready for the next transfer or comparison or has started on the next transfer or comparison if it was prearmed.

RESERVED - bits 1,0

These bits are reserved and must be written with zeros.

Prearm Status Register (RS = 03H)

Prearming can be checked and/or aborted via the Prearm Status Register. There is one bit in this register for each unit and each bit is set by the 9802A whenever the associated command register is prearmed. (See Figure 3-4.) The MPU should check the appropriate bit in the Prearm Status Register before writing to a prearmable register. The MPU can clear each bit by writing a one to it to disarm the associated unit. On hardware and master reset, all bits are cleared.

The bit in the Prearm Status Register associated with a given unit will be set during an operation if the command register is written with the HALT bit clear. Writing to the command register of an idle unit with the HALT bit clear will cause the shadow registers to be loaded into the working registers. This bit in the Prearm Status Register will then be cleared by the 9802A after the shadow-to-working transfer occurs. Writing to a command register with the HALT bit set halts the unit and clears the associated PREARMED bit. Note that all of the registers of the unit will contain their "running" values, not the values in the shadow registers when the HALT bit was set.

7	6	5	4	3	2	1	0
(reserved)	ECC Prearmed	(reserved)	DMA 3 Prearmed	DMA 2 Prearmed	DMA 1 Prearmed	(reserved)	

Figure 3-4. Prearm Status Register

RESERVED - bit 7

This bit is reserved and must be written with a zero.

ECC PREARMED - bit 6

The ECC PREARMED bit will be set by the 9802A when the ECC processor is in operation and the ECC Command Register is written with the HALT bit clear. This indicates that the ECC processor has been prearmed.

RESERVED - blt 5

This bit is reserved and must be written with a zero.

DMA 3 PREARMED - blt 4

The DMA 3 PREARMED bit will be set by the 9802A when the DMA channel 3 is in operation and the DMA 3 Command Register is written with the HALT bit clear. This indicates that DMA channel 3 has been prearmed.

DMA 2 PREARMED - blt 3

The DMA 2 PREARMED bit will be set when the DMA channel 2 is in operation and the DMA 2 Command Register is written with the HALT bit clear. This indicates that DMA channel 2 has been prearmed.

DMA 1 PREARMED - blt 2

The DMA 1 PREARMED bit will be set when the DMA channel 1 is in operation and the DMA 1 Command Register is written with the HALT bit clear. This indicates that DMA channel 1 has been prearmed.

RESERVED - blts 1,0

These bits are reserved and must be written with zeros.

Byte Increment Register (RS = 0AH, 0BH, 0CH)

The Byte Increment Register can be used by the DMA channels, the MPU Buffer Access Unit and the ECC processor. When the ADDRESS INCREMENT AMOUNT SELECT bit in the DMA Command Register is set, this register contains the signed value which is used to compute the address of the next byte for DMA and MPU transfers. For linear mode DMA transfers, this register may be added to the buffer address after every access. For matrix mode DMA transfers, this register may be added to the buffer address after every access within a row.

For MPU buffer access, it may be added to or subtracted from the buffer address after every access in continue mode. If the BYTE INCREMENT REGISTER SELECT bit in the ECC Command Register is clear, the ECC Processor uses this register for ECC operations. The ECC Processor assumes this register contains the separation between rows. Note that the address wraps. For example, if the current address is hex FF3271 and the Byte Increment Register contains 00CF00, adding the Byte Increment Register to the address gives a new address of 000171 to be used for the next transfer. On hardware and master reset, this register remains unchanged.

Row Increment Register (RS = 0DH, 0EH, 0FH)

The Row Increment Register contains the value by which to increment an address for matrix mode DMA transfers. At the end of a row, the Row Increment Register is added to the current address to generate the starting address of the next row. Note that the address wraps like the Byte Increment Register. On hardware and master reset, this register remains unchanged.

DMA Interface Registers

The 9802A has three DMA channels (1,2 and 3). DMA channel 1 is a high-speed 8-bit DMA bus with a 7-byte internal FIFO buffer. DMA channel 2 is a low-speed 8-bit DMA bus with a 1-byte internal FIFO buffer for tape formatters. DMA channel 3 shares the DMA channel 2 data bus but has its own handshaking pins. It can be used as a second formatter channel or to interface to another DMA device.

The two DMA data busses may be externally connected to create a DMA controller with a single data bus. The DMA LINKED bit in the DMA Compare/Link Register indicates this condition. When the LINKED bit is set, the 9802A will output a DMA acknowledge on only one channel at a time. The priority for multiple DMA requests on linked channels is the same as for DRAM access (channel 2, channel 3, channel 1).

The DMA Channel 1 Parity pin (DB1P) provides odd parity for the DMA Channel 1 Data Bus, and the DMA Channel 2/3 Parity pin (DB2P) provides odd parity for the DMA Channel 2/3 Data Bus. Parity errors are reported via the Interrupt Status Register and $\overline{\text{IRQ}}$ when enabled via the DMA Compare/Link Register.

The DMA of the 9802A is capable of transferring data in two modes: linear and matrix. Refer to the description of the LINEAR/MATRIX MODE SELECT bit in the DMA Command Registers for a definition of both modes.

For DMA channel 1, $\overline{\text{TC1}}$ transits high to low while DACK1 is active on the last byte of a transfer. $\overline{\text{TC1}}$ will remain active until after DACK1 goes inactive.

DMA Configuration Register (RS = 04H)

The DMA Configuration Register, diagrammed in Figure 3-5, configures the DMA channels for signal polarity, arbitration scheme and buffer RAM cycle time. This register is written during system initialization and is not prearmable. Except for the TRANSFER LENGTH ALLOCATION SELECT bits, on hardware reset these bits are set and on master reset these bits remain unchanged.

7	6	5	4	3	2	1	0
DACK3 Polarity Select	DREQ3 Polarity Select	DACK2 Polarity Select	DREQ2 Polarity Select	DACK1 Polarity Select	DREQ1 Polarity Select	Transfer Length Allocation Select	

Figure 3-5. DMA Configuration Register

DACK3 POLARITY SELECT - bit 7

Writing a one to the DACK3 POLARITY SELECT bit causes the DMA Channel 3 Acknowledge signal (DACK3) to be output as an active-high signal. Data is either driven while DACK3 is high or it is latched on the falling edge of DACK3 (depending on the value of the DIRECTION SELECT bit in the DMA 3 Command Register).

Writing a zero to the DACK3 POLARITY SELECT bit causes the DACK3 output to be active-low. Data is either driven while DACK3 is low or it is latched on the rising edge of DACK3 (depending on the direction of the transfer).

DREQ3 POLARITY SELECT - bit 6

Writing a one to the DREQ3 POLARITY SELECT bit causes the DMA Channel 3 Request signal (DREQ3) to be interpreted as an active-high signal. Writing a zero to this bit causes DREQ3 to be interpreted as active-low.

DACK2 POLARITY SELECT - bit 5

Writing a one to the DACK2 POLARITY SELECT bit causes the DMA Channel 2 Acknowledge signal (DACK2) to be output as an active-high signal. Data is either driven while DACK2 is high or it is latched on the falling edge of DACK2 (depending on the value of the DIRECTION SELECT bit in the DMA 2 Command Register).

Writing a zero to the DACK2 POLARITY SELECT bit causes the DACK2 output to be active-low. Data is either driven while DACK2 is low or it is latched on the rising edge of DACK2 (depending on the direction of the transfer).

DREQ2 POLARITY SELECT - bit 4

Writing a one to the DREQ2 POLARITY SELECT bit causes the DMA Channel 2 Request input (DREQ2) to be interpreted as an active-high signal. Writing a zero to this bit causes DREQ2 to be interpreted as active-low.

DACK1 POLARITY SELECT - bit 3

Writing a one to the DACK1 POLARITY SELECT bit causes the DMA Channel 1 Acknowledge output (DACK1) to be output as an active-high signal. Data is either driven while DACK1 is high or it is latched on the falling edge of DACK1 (depending on the value of the DIRECTION SELECT bit in the DMA 1 Command Register).

Writing a zero to the DACK1 POLARITY SELECT bit causes the DACK1 output to be active-low. Data is either driven while DACK1 is low or it is latched on the rising edge of DACK1 (depending on the direction of the transfer).

DREQ1 POLARITY SELECT - bit 2

Writing a one to the DREQ1 POLARITY SELECT bit causes the DMA Channel 1 Request signal (DREQ1) to be interpreted as an active-high signal. Writing a zero to this bit causes DREQ1 to be interpreted as active-low.

TRANSFER LENGTH ALLOCATION SELECT - bits 1,0

The TRANSFER LENGTH ALLOCATION SELECT bits determine which bits of the DMA Transfer Length Register hold the column size and which hold the row size when any DMA channel is in the matrix mode of operation. Refer to Table 3-6 below. Note that if a zero is written here, the maximum value will be used. On master and hardware reset these bits are unchanged.

Table 3-6. Transfer Length Allocation

<table>
<tr><th rowspan="2">bit 1,0</th><th colspan="16">DMA Transfer Length Register bits</th></tr>
<tr><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>9</th><th>8</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td>00</td><td colspan="10">Column Size</td><td colspan="6">Row Size</td></tr>
<tr><td>01</td><td colspan="7">Column Size</td><td colspan="9">Row Size</td></tr>
<tr><td>10</td><td colspan="5">Column Size</td><td colspan="11">Row Size</td></tr>
<tr><td>11</td><td colspan="3">Column Size</td><td colspan="13">Row Size</td></tr>
</table>

DMA Handshake Configuration Register (RS = 05H)

The DMA Handshake Configuration Register, diagrammed in Figure 3-8 below, configures the handshaking signals (DMA request and acknowledge). The LENGTH SELECT bits allow the DMA acknowledge signals to operate in either demand or four-cycle handshake mode.

In demand mode, the 9802A receives a DMA request (DREQ goes high) and acknowledges it (DACK goes high). Then, DACK is held active for the amount of time programmed in the LENGTH SELECT bits. DACK and DREQ are allowed to go inactive independently. Refer to Figure 3-6 below. This register is not prearmable.

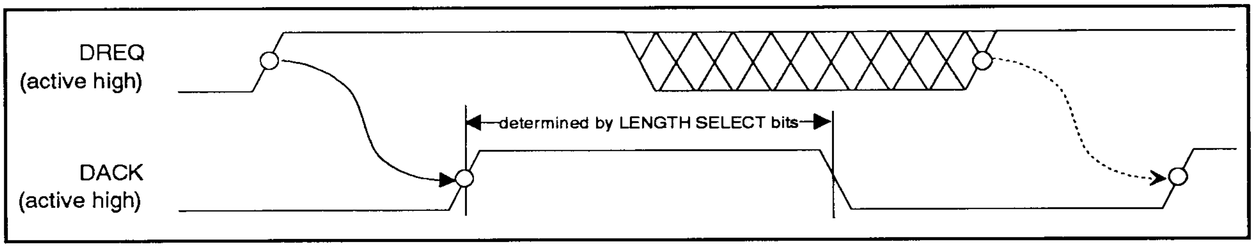


Figure 3-6. Demand Mode Handshaking

In four-cycle handshake mode, the 9802A receives a DMA request (DREQ goes high) and acknowledges it (DACK goes high). When the DMA request signal becomes inactive, the DMA acknowledge signal of the 9802A goes inactive. Refer to Figure 3-7 below.

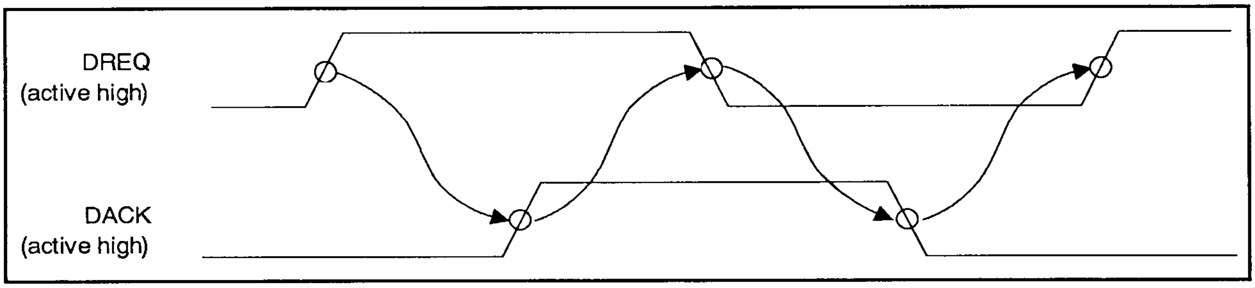


Figure 3-7. Four-Cycle Handshaking

7	6	5	4	3	2	1	0
DACK3 Length Select		DACK2 Length Select	DACK1 Length Select		DACK3 Enable	DACK2 Enable	DACK1 Enable

Figure 3-8. DMA Handshake Configuration Register

DACK3 LENGTH SELECT - bits 7, 6

The DACK3 LENGTH SELECT bits determine whether DACK3 will remain active during demand mode handshaking (and for how long) or whether DACK3 will follow four-cycle handshaking protocol. Table 3-7 below shows how these bits are interpreted. On hardware and master reset these bits are unchanged.

Table 3-7. DACK3 Length Select Interpretation

Bit 7,6	DACK3 Length
00	3 System Clock Cycles
01	5 System Clock Cycles
10	7 System Clock Cycles
11	Four-cycle

DACK2 LENGTH SELECT - bit 5

When the DACK2 LENGTH SELECT bit is clear, DACK2 will remain active for 7 clock cycles. When this bit is set, DACK2 follows four-cycle handshaking protocol. On hardware and master reset this bit is unchanged.

DACK1 LENGTH SELECT - bits 4, 3

The DACK1 LENGTH SELECT bits determine for how many clock cycles DACK1 will remain active or whether DACK1 will follow four-cycle protocol. Table 3-8 below shows how these bits are interpreted. On hardware and master reset these bits are unchanged.

Table 3-8. DACK1 Length Select Interpretation

Bit 4,3	DACK1 Length
00	3 System Clock Cycles
01	5 System Clock Cycles
10	7 System Clock Cycles
11	Four-cycle

DACK3 ENABLE - bit 2

When the DACK3 ENABLE bit is cleared, DACK3 is in a high impedance state and any requests from DREQ3 are ignored. When this bit is set, DACK3 is driven and requests from DREQ3 are acknowledged when the channel is ready. This bit can allow more than one device to act as the bus master of DMA channel 3. On hardware reset this bit is cleared, and on master reset this bit is unchanged.

DACK2 ENABLE - bit 1

When the DACK2 ENABLE bit is cleared, DACK2 is in a high impedance state and any requests from DREQ2 are ignored. When this bit is set, DACK2 is driven and requests from DREQ2 are acknowledged when the channel is ready. This bit can allow more than one device to act as the bus master of DMA channel 2. On hardware reset this bit is cleared, and on master reset this bit is unchanged.

DACK1 ENABLE - bit 0

When the DACK1 ENABLE bit is cleared, DACK1 is in a high impedance state and any requests from DREQ1 are ignored. When this bit is set, DACK1 is driven and requests from DREQ1 are acknowledged when the channel is ready. This bit can allow more than one device to act as the bus master of DMA channel 1. On hardware reset this bit is cleared, and on master reset this bit is unchanged.

DMA Compare/Link Register (RS = 06H)

The DMA Compare/Link Register, diagrammed in Figure 3-9 below, allows the two DMA data busses to be linked externally. It also reports on the status of byte-by-byte comparisons of data in the buffer with data read in over a DMA channel. This register is not prearmable.

7	6	5	4	3	2	1	0
DMA Linked	(reserved)	DMA Parity Error Interrupt Enable	TC1\TOE Select	Compare IRQ Enable	DMA 3 Compare Status	DMA 2 Compare Status	DMA 1 Compare Status

Figure 3-9. DMA Compare/Link Register

DMA LINKED - bit 7

A one is written to the DMA LINKED bit to indicate that the two DMA data busses (DB10-DB17 and DB20-DB27) are physically connected outside of the 9802A. Only one of the two channels should be acknowledged at a time and only one DMA acknowledge signal (DACK1, DACK2 or DACK3) will be active at a time. If two or more channels have pending requests, the channel with the highest priority will be acknowledged first. The priority order is: DMA channel 2, DMA channel 3, DMA channel 1.

A zero is written to the DMA LINKED bit to indicate that the two DMA data busses are not externally connected so DACK1 may be active simultaneously with DACK2 or DACK3. Note that DMA channels 2 and 3 always share the same DMA data bus (DB20-DB27), and therefore DACK2 and DACK3 will never be active simultaneously (DMA Channel 2 has higher priority). On hardware and master reset this bit is unchanged.

RESERVED - bit 6

This bits is reserved and must be written with a zero.

DMA PARITY ERROR INTERRUPT ENABLE - bit 5

When the DMA PARITY ERROR INTERRUPT ENABLE bit is clear, an interrupt will not be generated when a DMA Data Bus parity error is detected. When this bit is set, $\overline{\text{IRQ}}$ will be generated when a DMA Data Bus parity error is detected. Refer to the description of the DMA Data Bus parity pins (DB1P and DB2P) in *Signal Descriptions*. Hardware reset clears this bit. Master reset has no effect.

TC1/TOE SELECT - bit 4

The TC1/TOE SELECT bit determines whether the $\overline{TC1/TOE}$ output is the DMA Channel 1 Terminal Count signal ($\overline{TC1}$) or the Trigger on Error signal ($\overline{TOE}$). When this bit is clear, the $\overline{TC1/TOE}$ signal is the DMA Channel 1 Terminal Count signal. $\overline{TC1}$ is asserted on the last transfer of a DMA operation only while DACK is active.

When the TC1/TOE SELECT bit is set, the $\overline{TC1/TOE}$ signal is the Trigger on Error output signal. When the 9802A is in compare mode, $\overline{TOE}$ flags compare failures on data transferred into the 9802A. After a byte is transferred via DMA into the 9802A, it is compared with the corresponding byte in the buffer RAM. If a compare failure is detected, $\overline{TOE}$ is asserted before the next byte is transferred into the 9802A. Hardware reset clears this bit. Master reset has no effect.

COMPARE IRQ ENABLE - bit 3

When the COMPARE IRQ ENABLE bit is set, an interrupt is generated by a compare failure on any of the DMA channels whose DMA COMPARE STATUS bit is set. When this bit is clear, no interrupt is generated by a data mismatch. $\overline{TOE}$ can also be asserted when a compare failure is detected. Refer to the description of the COMPARE MODE ENABLE bit in the DMA Command Register. This bit is cleared on hardware and master reset.

DMA 3 COMPARE STATUS - bit 2

The DMA 3 COMPARE STATUS bit is used with read-after-write. If the COMPARE MODE ENABLE bit in the DMA 3 Command Register is set, this bit will be set by the 9802A on the first byte of a transfer that does not match the buffer contents. When this bit is clear, there has not been a data mismatch.

If the COMPARE IRQ ENABLE bit is set, the setting of the DMA 3 COMPARE STATUS bit by the 9802A causes the COMPARE FAILURE INTERRUPT bit in the Interrupt Status Register to be set and an interrupt to be generated. The MPU can clear this bit by writing a one here. This bit is cleared on hardware and master reset. Note that the buffer locations used are determined by the Address, DMA 3 Command and Transfer Length registers.

DMA 2 COMPARE STATUS - bit 1

The DMA 2 COMPARE STATUS bit is used with read-after-write. If the COMPARE MODE ENABLE bit in the DMA 2 Command Register is set, this bit will be set by the 9802A on the first byte of a transfer that does not match the buffer contents. When this bit is clear, it indicates there has not been a data mismatch.

If the COMPARE IRQ ENABLE bit is set, the setting of the DMA 2 COMPARE STATUS bit by the 9802A causes the COMPARE FAILURE INTERRUPT bit in the Interrupt Status Register to be set and an interrupt to be generated. The MPU can clear this bit by writing a one here. This bit is cleared on hardware and master reset. Note that the buffer locations used are determined by the Address, DMA 2 Command and Transfer Length registers.

DMA 1 COMPARE STATUS - bit 0

The DMA 1 COMPARE STATUS bit is used with read-after-write. If the COMPARE MODE ENABLE bit in the DMA 1 Command Register is set, this bit will be set by the 9802A on the first byte of a transfer that does not match the buffer contents. When this bit is clear, it indicates there has not been a data mismatch.

If the COMPARE IRQ ENABLE bit is set, the setting of the DMA 1 COMPARE STATUS bit by the 9802A causes the COMPARE FAILURE INTERRUPT bit in the Interrupt Status Register to be set and an interrupt to be generated. This bit can be cleared by writing a one here. This bit is cleared on hardware and master reset. Note that the buffer locations used are determined by the Address, DMA 1 Command and Transfer Length registers.

DMA Command Registers (RS = 12H, 1AH, 22H)

The command registers for DMA channels 1, 2 and 3 are the same. The following register description applies to all the DMA Channel Command Registers. Refer to Figure 3-10 below. These registers are prearmable.

7	6	5	4	3	2	1	0
Address Increment Amount Select	Halt	DMA Interrupt Enable	Compare Mode Enable	Direction Select	Linear/ Matrix Mode Select	IRQ Time Select	

Figure 3-10. DMA Command Registers

ADDRESS INCREMENT AMOUNT SELECT - bit 7

When the ADDRESS INCREMENT AMOUNT SELECT bit is clear, the address is incremented by 1 after each byte transfer. When this bit is set, the address is incremented by the amount in the Byte Increment Register after each byte transfer. On hardware and master reset, this bit is unchanged.

HALT - bit 6

When the HALT bit is cleared, the channel is started. When this bit is set, the channel is halted immediately. This bit is set by the 9802A at the end of a transfer if the channel is not prearmed. On hardware and master reset, this bit is set.

The bit in the Prearm Status Register associated with the DMA channel will be set by the 9802A when: (1) the channel is in operation, (2) the HALT bit is clear, and (3) the DMA Command Register is written. Writing to a DMA Command Register with the HALT bit set stops the DMA channel and clears the DMA PREARMED bit. Note that all of the registers of the DMA channel will contain their "running" values, not the values in the shadow registers when the HALT bit was set. When the channel is idle, writing to the DMA Command Register with the HALT bit clear will cause the shadow registers to be immediately loaded into the working registers.

DMA INTERRUPT ENABLE - bit 5

When the DMA INTERRUPT ENABLE bit is clear, interrupts are disabled. When this bit is set, interrupts will be generated at the end of transfers. Refer to the IRQ TIME SELECT bit in this register for the timing of interrupts. On hardware and master reset, this bit is unchanged.

COMPARE MODE ENABLE - bit 4

When the COMPARE MODE ENABLE bit is set, the 9802A operates in compare mode. Data is transferred from the DMA channel into the 9802A and compared byte-by-byte with data from the buffer RAM. No data is written to the buffer RAM. If a discrepancy is found, the corresponding DMA COMPARE STATUS bit in the DMA Compare/Link Register and the COMPARE FAILURE INTERRUPT bit in the Interrupt Status Register will be set by the 9802A and an interrupt will be generated.

When the 9802A is in compare mode and the TC1/TOE SELECT bit in the DMA Compare/Link Register is set, $\overline{\text{TOE}}$ also flags compare errors on data transferred into the 9802A. If a compare error is detected, $\overline{\text{TOE}}$ is asserted before the next byte is transferred into the 9802A. This signal can be used with any, of the DMA channels.

Note that when in compare mode, the DIRECTION SELECT bit must be cleared so data may be transferred from the buffer to the DMA channel. When the COMPARE MODE ENABLE bit is clear, data is transferred from the DMA channel to the buffer RAM and vice versa without being compared. On hardware and master reset this bit is unchanged.

DIRECTION SELECT - bit 3

When the DIRECTION SELECT bit is clear, data is transferred from the DMA channel to the buffer. When this bit is set, data is transferred from the buffer to the DMA channel. When in compare mode (when the COMPARE MODE ENABLE bit is set), this bit must be cleared so data may be transferred in from the DMA channel. On hardware reset this bit is set and on master reset this bit is unchanged.

LINEAR/MATRIX MODE SELECT - bit 2

When the LINEAR/MATRIX MODE SELECT bit is clear, data is transferred in linear mode: bytes are transferred in or out of memory "linearly". The address of the next byte to be transferred is incremented by one or by the value in the Byte Increment Register, as selected by the ADDRESS INCREMENT AMOUNT SELECT bit in this register.

When the LINEAR/MATRIX MODE SELECT bit is set, data is transferred in matrix mode: bytes arranged in a matrix of rows and columns are transferred row-by-row. The DMA Transfer Length Register contains the number/size of rows and columns in the matrix. Which bits define the rows and which define columns is programmable via the DMA Configuration Register. The address of the next byte in a row is computed by adding either a one or the value in the Byte Increment Register to the current address. The starting address of the next row is computed by adding the value in the Row Increment Register to the current address. On hardware and master reset, this bit is unchanged. Refer to *DMA Examples* for more information.

IRQ TIME SELECT - bits 1,0

The IRQ TIME SELECT bits allow the 9802A to generate interrupts based upon the number of bytes remaining to be transferred in the current block. Table 3-9 below shows the times when $\overline{\text{IRQ}}$ can be generated. Refer to *Electrical Specifications* for more information on the timing of $\overline{\text{IRQ}}$. Note that if the interrupt time (in bytes) is greater than the transfer length (for linear transfers) or the number of columns (for matrix transfers), an interrupt will not be generated. On hardware and master reset, these bits are unchanged.

Table 3-9. Interrupt Time Selection

Bits 1,0	Number of Bytes Remaining to be Transferred
00	0
01	2
10	4
11	8

DMA Address Registers (RS = 13H-15H, 1BH-1DH, 23H-25H)

The address registers for DMA channels 1, 2 and 3 function the same. The following register description applies to all of the DMA Address Registers.

Before a new DMA operation, the MPU must write the starting address of the operation to this register. During the operation, this register will contain the address of the next byte to be transferred to or from the buffer. At the end of the operation, this register will contain the address of the last byte transferred plus the increment amount. Refer to the description of the LINEAR/MATRIX MODE SELECT bit in the DMA Command Registers. This register is prearmable. On hardware and master reset this register is unchanged.

DMA Transfer Length Registers (RS = 16H-17H, 1EH-1FH, 26H-27H)

The DMA Transfer Length registers for DMA channels 1, 2 and 3 function the same. The following register description applies to all of the DMA Transfer Length Registers.

In linear mode this register contains the number of bytes to be transferred. In matrix mode it contains the column size (in bytes) in the high order bits and the row size (in bytes) in the low order bits. The TRANSFER LENGTH ALLOCATION SELECT bits in the DMA Configuration Register determine how many bits are used for the column size and how many for the row size. (Refer to Table 3-10 below.) The value read from this register is the number of bytes remaining to be transferred (if in linear mode) or the row and column sizes (if in matrix mode). Note that if a zero is written here, the maximum value will be used. This register is prearmable. On hardware and master reset this register is unchanged.

> **Note: For a row size greater than 1 and a column size equal to 1, a linear mode transfer with byte increment is recommended.**

Table 3-10. DMA Transfer Length Allocation

	DMA Transfer Length Register bits															
bit 1,0	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
00	Column Size (in bytes)										Row Size (in bytes)					
01	Column Size							Row Size								
10	Column Size					Row Size										
11	Column Size			Row Size												

MPU Buffer Access Unit

The MPU Buffer Access Unit gives the MPU access to the buffer. MPUDREQ is an active high request to the MPU. It indicates the value of the MPU DATA READY bit in the Status Register and may be used to transfer data via DMA between the MPU and the buffer. Refer to *Signal Descriptions* for more information about MPUDREQ. This unit can operate with continue mode enabled or disabled. Continue mode allows the MPU to perform multiple buffer accesses without having to program the MPU Buffer Command and Address Registers for each byte transferred.

MPU Buffer Command Register (RS = 2AH)

Figure 3-11 below diagrams the MPU Buffer Command Register. On hardware and master reset, the HALT and DIRECTION SELECT bits are set and all other bits remain unchanged.

7	6	5	4	3	2	1	0
Address Increment/ Decrement Amount Select	Halt	(reserved)	Address Increment/ Decrement Select	Direction Select	(reserved)	Continue Mode Enable	(reserved)

Figure 3-11. MPU Buffer Command Register

ADDRESS INCREMENT/DECREMENT AMOUNT SELECT - bit 7

When the ADDRESS INCREMENT/DECREMENT AMOUNT SELECT bit is clear, the address is incremented or decremented (as selected by the ADDRESS INCREMENT/DECREMENT SELECT bit) by 1. When this bit is set, the address is incremented or decremented by the amount in the Byte Increment Register.

HALT - bit 6

Writing a one to the HALT bit halts the MPU Buffer Access Unit. When this bit is clear, the MPU can access the buffer through the MPU Buffer Data Register described later. When the CONTINUE MODE ENABLE bit is clear, the HALT bit is set by the 9802A after a byte is read from the MPU Buffer Data Register by the MPU or written to the DRAM buffer by the 9802A.

RESERVED - bit 5

This bit is reserved and must be written with a zero.

ADDRESS INCREMENT/DECREMENT SELECT - bit 4

When the ADDRESS INCREMENT/DECREMENT SELECT bit is clear, the address is incremented, and the ADDRESS INCREMENT/DECREMENT AMOUNT SELECT bit selects the increment. When this bit is set, the address is decremented, and the ADDRESS INCREMENT/DECREMENT AMOUNT SELECT bit selects the decrement. The increment or decrement of the address takes place after each buffer access.

DIRECTION SELECT - bit 3

When the DIRECTION SELECT bit is clear, data is transferred from the MPU Buffer Data Register to the buffer. When this bit is set, data is transferred from the buffer to the MPU Buffer Data Register.

RESERVED - bit 2

This bit is reserved and must be written with a zero.

CONTINUE MODE ENABLE - bit 1

When the CONTINUE MODE ENABLE bit is clear, continue mode is disabled, and the MPU Buffer Command Register must be initialized for each MPU access. When the CONTINUE MODE ENABLE bit is set, continue mode is enabled, and the address is updated automatically after each access according to the settings of bits 4 and 7 in the MPU Buffer Command Register. The CONTINUE MODE ENABLE bit is unaffected by reset.

To write a byte to the buffer, the MPU clears the DIRECTION SELECT bit in the MPU Buffer Command Register. Then, the MPU must check that the MPU DATA READY bit in the Status Register is set. Next, the MPU writes a byte to the MPU Buffer Data Register. The 9802A will then write this byte to the buffer. Then, if continue mode is disabled, the 9802A will set the HALT bit. If continue mode is enabled, the 9802A will set the MPU DATA READY bit and update the address, and the MPU can write another byte to the MPU Buffer Data Register. Refer to Table 3-11 below.

To read a byte from the buffer, the MPU sets the DIRECTION SELECT bit. The 9802A will fetch a byte from the buffer and set the MPU DATA READY bit. Then, after the MPU checks that the MPU DATA READY bit is set, it can read the byte from the MPU Buffer Data Register. Then, if continue mode is disabled, the 9802A will set the HALT bit. If continue mode is enabled, the 9802A will update the address, fetch another byte from the buffer, and set the MPU DATA READY bit. Refer to Table 11 below.

Table 3-11. CONTINUE MODE ENABLE / DIRECTION SELECT **Interaction**

Continue Mode Enable	Direction Select	Direction	Operation
0	0	Write	1. The MPU waits for the 9802A to set the MPU DATA READY bit in the Status Register. 2. The MPU writes a byte to the MPU Buffer Data Register. 3. The 9802A writes the byte in the MPU Buffer Data Register to the buffer. 4. The 9802A sets the HALT bit.
0	1	Read	1. The 9802A fetches a byte from the buffer. 2. The MPU waits for the 9802A to set the MPU DATA READY bit. 3. The MPU reads the byte from the MPU Buffer Data Register. 4. The 9802A sets the HALT bit.
1	0	Write	1. The MPU waits for the 9802A to set the MPU DATA READY bit. 2. The MPU writes a byte to the MPU Buffer Data Register. 3. The 9802A writes the byte in the MPU Buffer Data Register to the buffer. 4. The 9802A sets the MPU DATA READY bit and updates the address. (The steps repeat.)
1	1	Read	1. The 9802A fetches a byte from the buffer. 2. The MPU waits for the 9802A to set the MPU DATA READY bit. 3. The MPU reads the byte from the MPU Buffer Data Register. 4. The 9802A updates the address and fetches the next byte from the buffer. (The steps repeat.)

RESERVED - bit 0

This bit is reserved and must be written with a zero.

MPU Buffer Address Register (RS = 2BH, 2CH, 2DH)

The MPU Buffer Address Register contains the buffer address for reading and writing data. When the CONTINUE MODE ENABLE bit is set, the address used for MPU buffer access is updated after each access. The value read from this register is the address of the next byte to be transferred. If the CONTINUE MODE ENABLE bit is clear, MPU buffer access does not effect the contents of this register. On hardware and master reset this register is unchanged.

MPU Buffer Data Register (RS = 30H)

When the DIRECTION SELECT bit in the MPU Buffer Command Register is clear, the MPU Buffer Data Register is a write-only register. The MPU writes to this register the data to be written to the buffer. When the DIRECTION SELECT bit in the MPU Buffer Command Register is set, this is a read-only register. The MPU can read the data just read from the buffer. To switch between reading and writing the MPU Buffer Data Register, rewrite the MPU Buffer Command Register as described in the *Recommended MPU Buffer Access Procedure* below. On hardware and master reset, this register is unchanged.

If the MPU tries to access this register while the MPU DATA READY bit is clear, the 9802A will assert $\overline{\text{WAIT}}$. $\overline{\text{WAIT}}$ is removed as soon as the MPU DATA READY bit is set by the 9802A. It is recommended that $\overline{\text{WAIT}}$ be connected to the processor. Unless the $\overline{\text{WAIT}}$ line is connected to the processor, the MPU must check that the MPU DATA READY bit in the Status Register is set before accessing the MPU Buffer Data Register.

Note: **Corruption of buffer data may result if the MPU reads or writes the MPU Buffer Data Register before checking that the MPU DATA READY bit in the Status Register is set. This is not important if the $\overline{\text{WAIT}}$ line is connected to the MPU $\overline{\text{WAIT}}$ input to halt the MPU.**

Recommended MPU Buffer Access Procedure

The following is the recommended procedure for accessing the 9802A buffer memory contents over the MPU interface.

To read or write one or more bytes of the buffer requires two steps:

(1) Load the MPU Buffer Address Register with the address for first access.
(2) Write to the MPU Buffer Command Register as follows:
  (a) HALT bit clear,
  (b) CONTINUE MODE ENABLE bit set,
  (c) DIRECTION SELECT bit indicating read/write,
  (d) ADDRESS INCREMENT/DECREMENT AMOUNT SELECT bit indicating the address step amount. (See Table 3-12 below.)

Table 3-12. Typical MPU Buffer Command Register Values

Step Amount	Read Buffer	Write Buffer
+1	0Ah	02h
-1	1Ah	12h
+Byte Inc Reg	8Ah	82h
-Byte Inc Reg	9Ah	92h

(3) Loop one or more times:

(a) IF the $\overline{\text{WAIT}}$ line is connected (which is recommended), ignore this step. IF NOT, wait for the MPU DATA READY bit in the Status Register to be set by the 9802A, then —

(b) Read/write MPU Buffer Data Register with data.

(4) IF the $\overline{\text{WAIT}}$ line is connected, ignore this step. IF NOT, wait for the MPU DATA READY bit in Status Register to be set by the 9802A.

(5) Write the MPU Buffer Command Register with the HALT bit set (e.g., write 40h).

ECC Processor

The ECC processor performs its calculations on a matrix of data using one of four addressing modes: row oriented; column oriented; column oriented, XOR 2; or column oriented, XOR 4. The mode is set by the ADDRESS MODE SELECT bits in the ECC Command Register.

Row oriented ECC calculations are performed across rows with the column size set in the ECC Matrix Size Register and the row size set in the ECC Row/Column Size Register. Column oriented ECC calculations are performed down columns with the column size set in the ECC Row/Column Size Register and the row size set in the ECC Matrix Size Register. Refer to the description of the ADDRESS MODE SELECT bits and to *ECC Examples* for more information.

The ECC processor can perform three types of operations: parity generation, syndrome generation and error correction. Table 3-13 below shows what values the MPU must write to the ECC registers for each operation.

Table 3-13. ECC Processor Operation Set-up

	Parity Generation	Syndrome Generation	Error Correction
ECC OPERATION SELECT bit	1	1	0
WRITE/XOR SELECT bit	0	0	1
ECC Source Address Register	Starting address of data to be encoded	Starting address of syndromes	Starting address of syndromes previously generated
ECC Destination Address Register	Starting address of parity	Starting address of parity	Address of row/column in error
ECC Matrix Size Register	Number of bytes in a block	Number of bytes in a block	Number of bytes in a block
ECC Row/Column Size Register	Number of rows/columns in an ECC matrix	Number of data rows/columns plus the number of parity rows/columns	Original redundancy value
ECC Redundancy Register	Redundancy of the code	Redundancy of the code	1
ECC Polynomial Coefficient Stack	Code generation polynomial	Code generation polynomial	Error correction vector
ECC Byte Increment Register	Separation between rows	Separation between rows	Separation between rows

The ECC processor can write its results to the destination address or XOR them with the value already there (determined by the WRITE/XOR SELECT bit in the ECC Command Register). Typically, writing is used for parity and syndrome generation and XOR for error correction.

For all ECC operations, the NON-ZERO bit in the Status Register is set any time the ECC processor writes a non-zero value to memory. This bit should be cleared before beginning syndrome calculations.

ECC Redundancy Register (RS = 39H)

For parity and syndrome generation, the MPU must load the ECC Redundancy Register with the redundancy of the code. The range of possible values for this register is from 1 to 8. Refer to Table 3-14 below for examples of values to use. For error correction, only one error may be corrected at a time so a 1 must be written here. On hardware and master reset this register is unchanged.

Table 3-14. ECC Redundancy Register Values

Format	ECC Redundancy Register (decimal)	
	Parity and Syndrome Generation	Error Correction
QIC-112 (QIC-24, QIC-120/150), QIC-525, QIC-2110	2	1
QIC-40/80	3	1
QIC-100	1	1
QIC-1350	6	1
Data/DAT (ID=0)	6	1
Data/DAT (ID=1)	5	1
DDS	2	1
8mm (vertical)	4	1
8mm (horizontal)	8	1

ECC GF(256) Feedback Register (RS = 38H)

The value written to the ECC GF(256) Feedback Register gives the GF(256) field representation for performing field multiplication. The bits loaded in this register are the coefficients of the primitive binary polynomial used to generate the field. For all QIC physical standards, the value written to this register is 87h. For Data/DAT standards, the value written is 1Dh. Refer to Table 3-15. On hardware and master reset this register is unchanged.

Table 3-15. ECC GF(256) Feedback Register Values

Format	Value (hex)
All QIC formats	87
Data/DAT, 8mm	1D
DDS	1D

ECC Polynomial Coefficient Stack (RS = 31H)

The ECC Polynomial Coefficient Stack contains the multipliers for ECC operations. All eight bytes of the stack must be written and each write to this register pushes a new entry on the stack. Any writes after the eighth write "push off" the oldest value of the stack. For parity and syndrome generation, the coefficients are written in descending order from the second-highest order coefficient to the zero order coefficient, and any unused bytes are written last with a value of 0. The highest order coefficient in all polynomial equations is a one so the coefficient stack always starts with an implicit leading one. For error correction, the error correction vector is written first and any unused bytes are written last with a value of 0.

When the ECC Polynomial Coefficient Stack is read, it returns the oldest value on the stack. Table 3-16 shows the values to use for the formats specified. The table reads from the oldest value on the left to the newest on the right. On hardware and master reset the stack is unchanged.

Table 3-16. ECC Polynomial Coefficient Stack Register Values

Format	ECC Polynomial Coefficient Stack (hex)							
QIC-112 (QIC-24, QIC-120/150), QIC-525, QIC-2110	03	02	00	00	00	00	00	00
QIC-40/80	C0	C0	01	00	00	00	00	00
QIC-100	01	00	00	00	00	00	00	00
QIC-1350	3F	28	A6	12	56	F4	00	00
Data/DAT (ID=0)	3F	01	DA	20	E3	26	00	00
Data/DAT (ID=1)	1F	C6	3F	93	74	00	00	00
DDS	03	02	00	00	00	00	00	00
8mm (vertical)	89	83	0F	04	00	00	00	00
8mm (horizontal)	00	89	00	83	00	0F	00	04

ECC Command Register (RS = 32H)

If the ECC Command Register, diagrammed in Figure 3-12, is prearmed when the HALT bit is clear, the ECC PREARMED bit in the Prearm Status Register will be set. If this register is written when the ECC processor is idle and the HALT bit is clear, the values in the shadow registers will be loaded into the working registers. Writing to this register when the HALT bit is set halts the processor and clears the ECC PREARMED bit. Note that all registers of the ECC processor will contain their "running" values and not the values in the shadow registers when the HALT bit was set.

7	6	5	4	3	2	1	0
Byte Increment Register Select	Halt	ECC Interrupt Enable	Destination Increment/ Decrement Select	ECC Operation Select	Write/XOR Select	Address Mode Select	

Figure 3-12. ECC Command Register

BYTE INCREMENT REGISTER SELECT - bit 7

When the BYTE INCREMENT REGISTER SELECT bit is clear, the Byte Increment Register will be used for ECC operations. When this bit is set, the ECC Byte Increment Register will be used for ECC operations. On hardware and master reset this bit is unchanged.

HALT - bit 6

When the HALT bit is clear, the ECC processor is started. When this bit is set, the processor is halted. This bit is set by the 9802A at the completion of an ECC operation if the ECC processor is not prearmed. On hardware and master reset this bit is set.

ECC INTERRUPT ENABLE - bit 5

When the ECC INTERRUPT ENABLE bit is clear, interrupts are disabled. When this bit is set, an interrupt will be generated upon completion of the ECC operation. On hardware and master reset this bit is unchanged.

DESTINATION INCREMENT/DECREMENT SELECT - bit 4

When the DESTINATION INCREMENT/DECREMENT SELECT bit is clear, the destination address is incremented between bytes according to the values of the ADDRESS MODE SELECT bits in this register. When the DESTINATION INCREMENT/DECREMENT SELECT bit is set, the destination address is decremented within a row or column according to the values of the ADDRESS MODE SELECT bits. This bit does not affect the operation of the column to column calculations in row oriented ECC or row to row calculations in column oriented ECC. On hardware and master reset this bit is unchanged.

ECC OPERATION SELECT - bit 3

The ECC OPERATION SELECT bit selects the type of ECC operation to be performed. This bit should be cleared to perform error correction and set to perform syndrome or parity block generation. On hardware and master reset this bit is unchanged.

WRITE/XOR SELECT - bit 2

When the WRITE/XOR SELECT bit is clear, the parity byte or decoded byte is written to the destination. When this bit is set, the parity byte or decoded byte is XOR'd with the byte currently stored in the buffer RAM at the destination address. On hardware and master reset this bit is unchanged.

ADDRESS MODE SELECT - bits 1,0

The ADDRESS MODE SELECT bits determine the layout of the data for ECC operations. The ECC processor can use four modes of addressing: row oriented; column oriented; column oriented, XOR 2; and column oriented, XOR 4. For all QIC format standards, column oriented ECC is used. For DAT formats, any mode may be used. The difference between modes is the method used to compute the addresses of the bytes during ECC operations. On hardware and master reset these bits are unchanged.

Table 3-17 below shows the bit settings for each mode with the address computation performed in each. For computations where "±" is indicated, subtraction is done only if the DESTINATION INCREMENT/DECREMENT SELECT bit is set and this is a destination address calculation.

Table 3-17. ECC Address Mode Selection

bit 1,0	Address Mode	Address of Next Byte	Address of Next Row/Column
00	row oriented	**addr** ± 1	**row** + **Byte Inc**
01	column oriented	**addr** ± **Byte Inc**	**col** + 1
10	column oriented, XOR 2	odd byte: (**addr** XOR 2) ± **Byte Inc** even byte: **addr** XOR 2	column # 3, 7, 11, 15, etc.: **col** + 5 other columns: **col** + 1
11	column oriented, XOR 4	odd byte: (**addr** XOR 4) ± **Byte Inc** even byte: **addr** XOR 4	column # 3, 7, 11, 15, etc.: **col** + 5 other columns: **col** + 1

addr = address of current byte
Byte Inc = value in the ECC Byte Increment Register
row = starting address of current row
col = starting address of current column

In row oriented mode, the address of the next byte within the row is the address of the current byte plus 1. The starting address of the next row is the starting address of the current row plus the value in the ECC Byte Increment Register or the Byte Increment Register.

In column oriented mode, the address of the next byte within the column is the address of the current byte plus the value in the ECC Byte Increment Register or the Byte Increment Register. The starting address of the next column is the starting address of the current column plus 1.

In column oriented, XOR 2 mode, the address of the next byte within the column depends on whether the next byte is even or odd. Even bytes are the second, fourth, sixth, etc., bytes within a column and odd bytes are the first, third, fifth, etc. If the next byte is even, the address of the next byte is the address of the current byte XOR'd with 2. If the next byte is odd, the address of the next byte is the address of the current byte XOR'd with 2 plus the value in the ECC Byte Increment Register or the Byte Increment Register. The starting address of the next column in the frame is the starting address of the current column plus 1 or 5. Five is added only if the next column is the third, seventh, eleventh, fifteenth, etc. Refer to *ECC Examples* for examples of the addressing in various modes.

In column oriented, XOR 4 mode, the address of the next byte within the column depends upon whether the next byte is even or odd. If the next byte is even, then the next address is the current byte address XOR'd with 4. If the next byte is odd, the address used is the current byte address XOR'd with 4 plus the value in the ECC Byte Increment Register or the Byte Increment Register. The starting address of the next column is the starting address of the current column plus 1 or 5. Five is added only if the next column is the third, seventh, eleventh, fifteenth, etc. Refer to *ECC Examples* for examples of the addressing in various modes.

ECC Source Address Register (RS = 33H, 34H, 35H)

The ECC Source Address Register contains the buffer address for the data to be used in the ECC operations. This register should be loaded with the starting address of the data for the next ECC operation. The value read from this register is the address of the next byte to be used in the operation. At the end of an operation, the value read will be the address of the next row or column (depending on the value of the ADDRESS MODE SELECT bits). This register is prearmable. On hardware and master reset this register is unchanged.

ECC Destination Address Register (RS = 3BH, 3CH, 3DH)

The ECC Destination Address Register contains the buffer address for the results of the ECC calculations. The value written to this register is the starting address where the results of operation are to be stored. The value read from this register is the address where the next result of the ECC operation will be stored. At the end of an operation, the value read will be the address of the next row or column (depending on the value of the ADDRESS MODE SELECT bits). This register is prearmable. On hardware and master reset this register is unchanged.

ECC Row/Column Size Register (RS = 37H)

When the 9802A performs syndrome generation, the ECC Row/Column Size Register contains the total number of bytes (data and parity) in a row or column of an ECC matrix. The value of the ADDRESS MODE SELECT bits in the ECC Command Register determines whether the 9802A is performing row oriented or column oriented ECC. When the 9802A performs parity generation, this register contains the number data bytes in a row or column of an ECC matrix. When the 9802A performs error correction, this register contains the original redundancy value. Note that if a zero is written here, the maximum value will be used. Refer to Table 3-18 below. This register must be initialized for each ECC operation. On hardware and master reset this register is unchanged.

Table 3-18. ECC Row/Column Size Register Values

Format	ECC Row/Column Size Register (decimal)		
	Syndrome Generation	Parity Generation	Error Correction
QIC-112(QIC-24, QIC-120/150)	32	30	2
QIC-525, QIC-2110	16	14	2
QIC-40/80	32	29	3
QIC-100	3	2	1
QIC-1350	32	26	6
Data/DAT (ID=0)	38	32	6
Data/DAT (ID=1)	64	59	5
DDS	46	44	2
8mm (vertical)	24	20	4
8mm (horizontal)	60	52	8

ECC Matrix Size Register (RS = 3EH, 3FH)

The ECC Matrix Size Register contains the number of columns in an ECC matrix when the 9802A performs column oriented ECC (the ADDRESS MODE SELECT bits in the ECC Command Register are 01, 10 or 11) or the number of rows in an ECC matrix when the 9802A performs row oriented ECC (the ADDRESS MODE SELECT bits in the ECC Command Register are 00). Refer to Table 3-19 below. Note that if a zero is written here, the maximum value will be used. This register must be initialized for each ECC operation. On hardware and master reset, the value of this register is unchanged.

Table 3-19. ECC Matrix Sizes

Format	ECC Matrix Size Register (decimal)
QIC-112 (QIC-24, QIC-120/150)	512
QIC-525, QIC-2110	1025
QIC-40/80	1024
QIC-100	4160
QIC-1350	513
Data/DAT (ID=0)	4244
Data/DAT (ID=1)	4044
DDS	2878
8mm (vertical)	60
8mm (horizontal)	20

ECC Byte Increment Register (RS = 07, 08, 09)

When the BYTE INCREMENT REGISTER SELECT bit in the ECC Command Register is set, the ECC Processor uses the ECC Byte Increment Register for ECC operations. The ECC Processor assumes that the ECC Byte Increment Register contains the separation between rows. Note that this address wraps. (See earlier Byte Increment Register description.) On hardware and master reset, the value of this register is unchanged.

Register Summary

The tables in this subsection are a quick reference of the bit assignments, reset status, and interrupt sources for the 9802A control registers, which are presented in greater detail in the preceding subsections.

Table 3-20. Register Bit Summary

Name	RS	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Configuration	00	Master Reset	Buffer Arbitration Priority Select	Buffer Parity Error Interrupt Enable	RAM Size Select		Buffer RAM Cycle Time Select	DRAM Refresh Rate Select	
Interrupt Status	01	Compare Failure Interrupt	ECC Interrupt	Buffer Parity Error Interrupt	DMA 3 Interrupt	DMA 2 Interrupt	DMA 1 Interrupt	DMA Channel 2/3 Parity Error Interrutp	DMA Channel 1 Parity Error Interrupt
Status	02	Non-Zero	ECC Operation Done	MPU Data Ready	DMA 3 Done	DMA 2 Done	DMA 1 Done	(reserved)	(reserved)
Prearm Status	03	(reserved)	ECC Prearmed	(reserved)	DMA 3 Prearmed	DMA 2 Prearmed	DMA 1 Prearmed	(reserved)	(reserved)
DMA Configuration	04	DACK3 Polarity Select	DREQ3 Polarity Select	DACK2 Polarity Select	DREQ2 Polarity Select	DACK1 Polarity Select	DREQ1 Polarity Select	Transfer Length Allocation Select	
DMA Handshake Configuration	05	DACK3 Length Select		DACK2 Length Select	DACK1 Length Select		DACK3 Enable	DACK2 Enable	DACK1 Enable
DMA Compare/Link	06	DMA Linked	(reserved)	DMA Parity Error Interrupt Enable	TC1/TOE Select	Compare IRQ Enable	DMA 3 Compare Status	DMA 2 Compare Status	DMA 1 Compare Status
ECC Byte Increment	07 08 09	High byte Value to increment or decrement an address... Low byte							
Byte Increment	0A 0B 0C	High byte Value to increment or decrement an address between bytes (3 bytes) Low byte							
Row Increment	0D 0E 0F	High byte Value to increment or decrement an address between rows in matrix mode (3 bytes) Low byte							
DMA 1 Command	12	Address Increment Amount Select	Halt	DMA Interrupt Enable	Compare Mode Enable	Direction Select	Linear/ Matrix Mode Select	IRQ Time Select	
DMA 1 Address	13 14 15	High byte Buffer address for DMA access (3 bytes) Low byte							
DMA 1 Transfer Length	16 17	High byte Number of bytes to be transferred (2 bytes) Low byte							
DMA 2 Command	1A	Address Increment Amount Select	Halt	DMA Interrupt Enable	Compare Mode Enable	Direction Select	Linear/ Matrix Mode Select	IRQ Time Select	

(continued)

Table 3-20. Register Bit Summary (cont'd)

Name	RS	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DMA 2 Address	1B 1C 1D	High byte Low byte	Buffer address of the next DMA access (3 bytes)						
DMA 2 Transfer Length	1E 1F	High byte Low byte	Number of bytes to be transferred (2 bytes)						
DMA 3 Command	22	Address Increment Amount Select	Halt	DMA Interrupt Enable	Compare Mode Enable	Direction Select	Linear/ Matrix Mode Select	IRQ Time Select	
DMA 3 Address	23 24 25	High byte Low byte	Buffer address for DMA access (3 bytes)						
DMA 3 Transfer Length	26 27	High byte Low byte	Number of bytes to be transferred (2 bytes)						
MPU Buffer Command	2A	Address Increment/ Decrement Amount Select	Halt	(reserved)	Address Increment/ Decrement Select	Direction Select	(reserved)	Continue Mode Enable	(reserved)
MPU Buffer Address	2B 2C 2D	High byte Low byte	Buffer address to read data from or write data to (3 bytes)						
MPU Buffer Data	30	Data to be written or data just read							
ECC Polynomial Coefficient Stack	31	Multipliers for encode or decode operations							
ECC Command	32	Byte Increment Register Select	Halt	ECC Interrupt Enable	Destination Increment/ Decrement Select	ECC Operation Select	Write/XOR Select	Address Mode Select	
ECC Source Address	33 34 35	High byte Low byte	Buffer address of the encode or decode source (3 bytes)						
ECC Row/Column Size	37	Number of bytes in a row or column							
ECC GF(256) Feedback	38	GF(256) field representation for performing field multiplication							
ECC Redundancy	39	Redundancy for the ECC operations							
ECC Destination Address	3B 3C 3D	High byte Low byte	Buffer address of the encode or decode destination (3 bytes)						
ECC Matrix Size	3E 3F	High byte Low byte	The number of rows or columns in an ECC matrix (2 bytes)						

NOTE: Reserved bits must be written with zeros for compatibility with future Stac products.

Table 3-21. Register Values on Reset

This table shows the register values on both hardware and master reset. Asserting RESET causes hardware reset. Setting the MASTER RESET bit in the Configuration Register causes master reset. Note that hardware reset causes the MASTER RESET bit to be set. Each character in the Reset Value column represents a bit. The most significant bit is on the left. A "-" indicates that the bit is unchanged.

Register	RS	Reset Value	
	(hex)	Hardware	Master
Configuration	00	1-0--100	--------
Interrupt Status	01	00000000	00000000
Status	02	00000000	00000000
Prearm Status	03	00000000	00000000
DMA Configuration	04	111111--	--------
DMA Handshake Configuration	05	-----000	--------
DMA Compare/Link	06	-0000000	----0000
ECC Byte Increment	07 - 09	------------------------	------------------------
Byte Increment	0A - 0C	------------------------	------------------------
Row Increment	0D - 0F	------------------------	------------------------
DMA 1 Command	12	-1--1---	-1------
DMA 1 Address	13 - 15	------------------------	------------------------
DMA 1 Transfer Length	16, 17	----------------	----------------
DMA 2 Command	1A	-1--1---	-1------
DMA 2 Address	1B - 1D	------------------------	------------------------
DMA 2 Transfer Length	1E, 1F	----------------	----------------
DMA 3 Command	22	-1--1---	-1------
DMA 3 Address	23 - 25	------------------------	------------------------
DMA 3 Transfer Length	26, 27	----------------	----------------
MPU Buffer Command	2A	-1--1---	-1------
MPU Buffer Address	2B - 2D	------------------------	------------------------
MPU Buffer Data	30	--------	--------
ECC Polynomial Coefficient Stack	31	--------	--------
ECC Command	32	-1------	-1------
ECC Source Address	33 - 35	------------------------	------------------------
ECC Row/Column Size	37	--------	--------
ECC GF(256) Feedback	38	--------	--------
ECC Redundancy	39	--------	--------
ECC Destination Address	3B - 3D	------------------------	------------------------
ECC Matrix Size	3E, 3F	----------------	----------------

Table 3-22. Summary of Interrupt Sources

Interrupt Source	How Enabled	Bits set by 9802A to assert$\overline{\text{IRQ}}$	How Cleared
Buffer RAM parity error detected	MPU sets the BUFFER PARITY ERROR INTERRUPT ENABLE bit	BUFFER PARITY ERROR INTERRUPT	Write a one to BUFFER PARITY ERROR INTERRUPT bit
Parity error on DMA Channel 1 data bus detected	MPU sets the DMA PARITY ERROR INTERRUPT ENABLE bit	DMA CHANNEL 1 PARITY ERROR INTERRUPT	Write a one to DMA PARITY ERROR INTERRUPT bit
Parity error on DMA Channel 2/3 data bus detected	MPU sets the DMA PARITY ERROR INTERRUPT ENABLE bit	DMA CHANNEL 2 PARITY ERROR INTERRUPT	Write a one to DMA PARITY ERROR INTERRUPT bit
Compare failure on a DMA transfer	MPU sets the COMPARE MODE ENABLE bit, COMPARE IRQ ENABLE bit, and the appropriate DMA COMPARE STATUS bits	COMPARE FAILURE INTERRUPT (If the TC1/TOE SELECT bit was set by the MPU, $\overline{\text{TOE}}$ will also be asserted)	MPU writes a one to all DMA COMPARE STATUS bits
ECC operation completed	MPU sets the ECC INTERRUPT ENABLE bit	ECC INTERRUPT	Write a one to ECC INTERRUPT bit
DMA transfer completed	MPU sets the DMA INTERRUPT ENABLE bit and the appropriate IRQ TIME SELECT bits	The appropriate DMA INTERRUPT bit	Write a one to DMA INTERRUPT bit

Interfacing with the 9802A

Peripheral Access through DMA Channel 1

The 9802A allows access to single-bus peripherals over DMA channel 1. This feature makes it possible to use a single-bus peripheral chip, such as a SCSI chip, in a dual-bus system. Dual-bus systems offer higher transfer rates than single-bus systems because they use separate busses for MPU communication with peripheral control chips and data transfers to and from tape. The diagram below shows a tape controller system. The connection between the 9802A and the SCSI chip uses the peripheral access capabilities of the 9802A. (See Figure 4-1 below.)

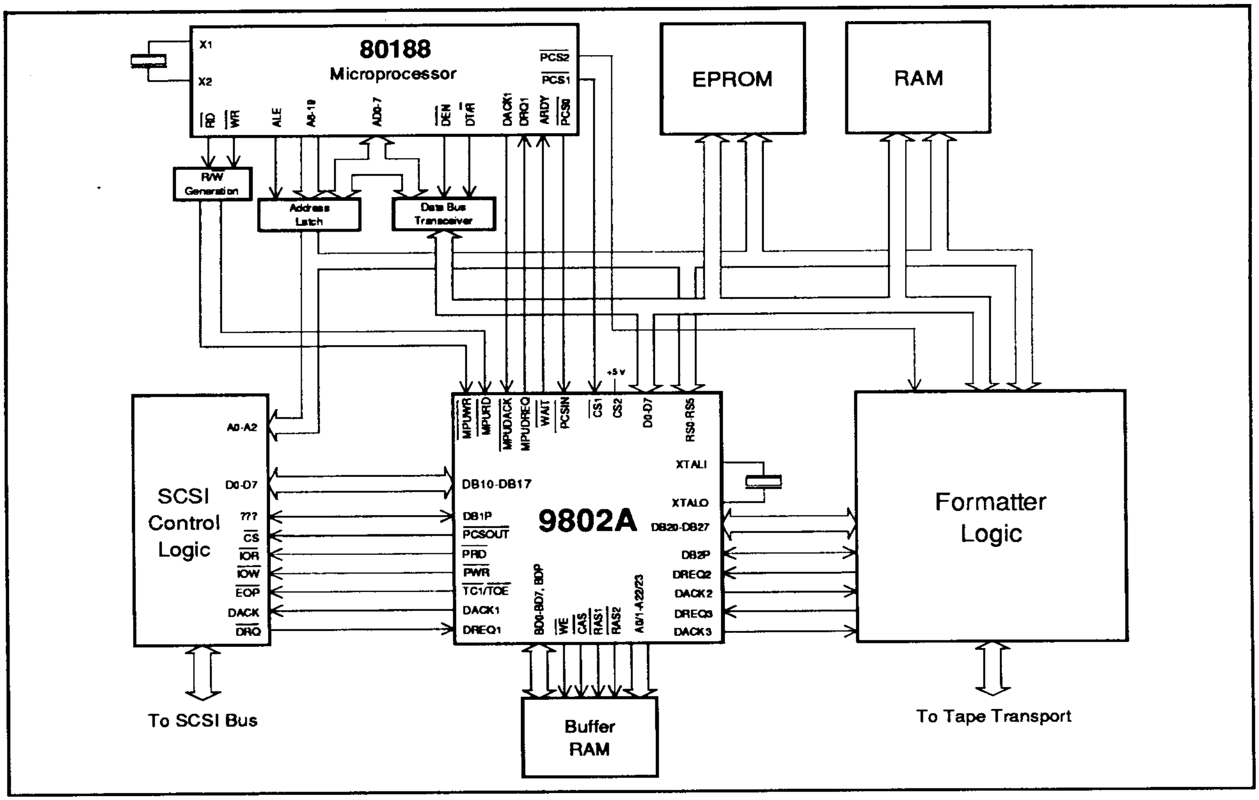


Figure 4-1. Peripheral Access with the 9802A

During a peripheral access, the 9802A passes data between the MPU data bus and the SCSI chip. First the MPU requests a transfer to the SCSI chip by asserting $\overline{\mathrm{PCSIN}}$ while CS2 is active. Once $\overline{\mathrm{PCSIN}}$ and CS2 become active, the 9802A immediately asserts $\overline{\mathrm{WAIT}}$. When DACK1 and DREQ1 become inactive, access to the peripheral is granted. The 9802A will assert $\overline{\mathrm{PCSOUT}}$ and then deassert $\overline{\mathrm{WAIT}}$. The 9802A will transfer data between the MPU and the SCSI chip via DMA channel 1, asserting $\overline{\mathrm{PRD}}$ or $\overline{\mathrm{PWR}}$ as determined by $\overline{\mathrm{MPURD}}$ and $\overline{\mathrm{MPUWR}}$. When the MPU has completed its transfer to the SCSI chip, it will deassert $\overline{\mathrm{PCSIN}}$.

For processors with one R/$\overline{\mathrm{W}}$ signal, the peripheral access cycle will be initiated when $\overline{\mathrm{PCSIN}}$ and CS2 are active. For processors with separate $\overline{\mathrm{RD}}$ and $\overline{\mathrm{WR}}$ signals, the peripheral access cycle will be initiated when $\overline{\mathrm{PCSIN}}$ and CS2 are active and either $\overline{\mathrm{MPURD}}$ or $\overline{\mathrm{MPUWR}}$ is active.

Interfacing to the DRAM Buffer

Figure 4-2 is one example of a 9802A/DRAM buffer interconnection. Refer to the DRAM Interface section of the *Product Overview* for more interfacing information, and refer to *Electrical Specifications* for details regarding timing requirements.

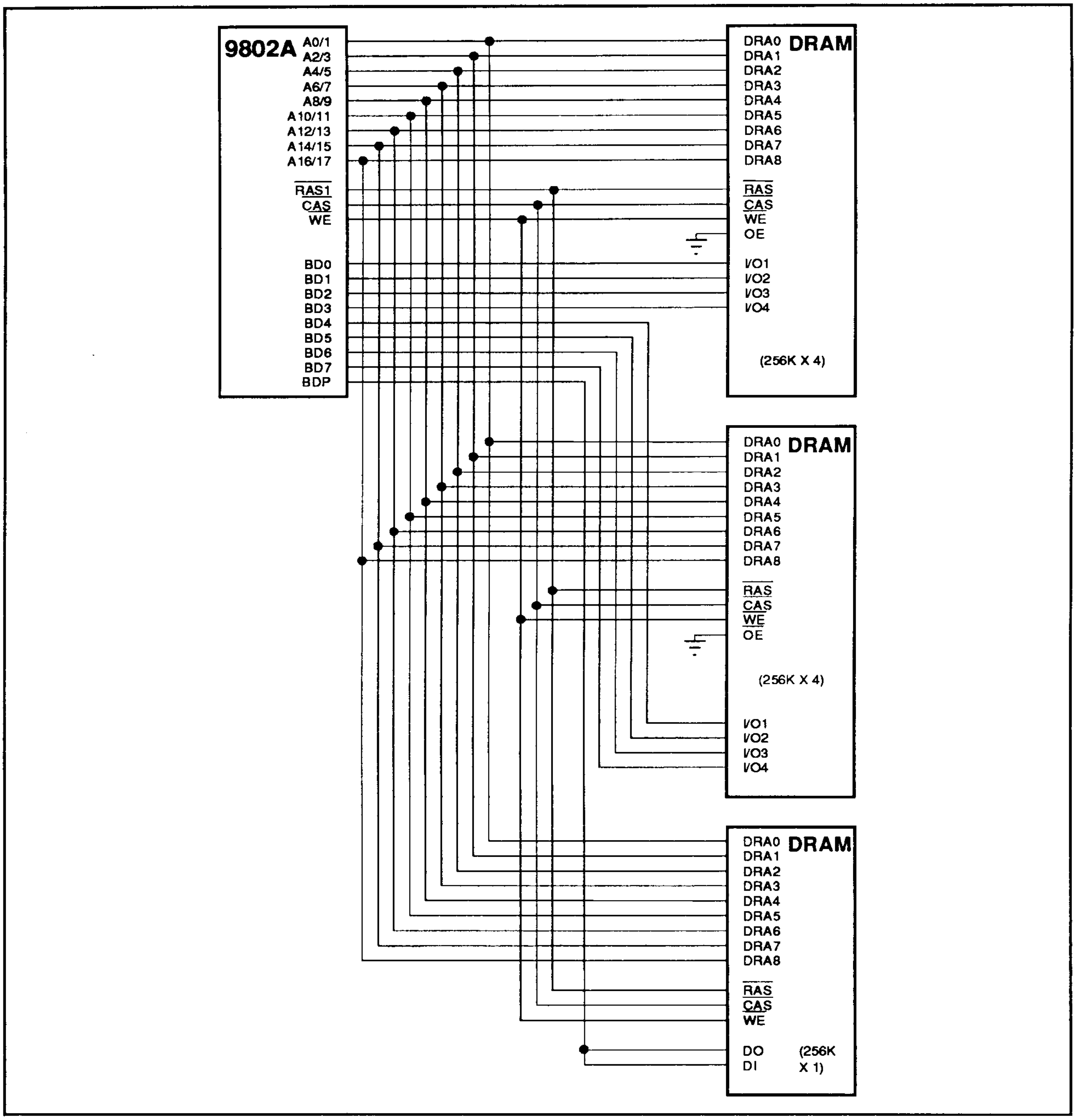


Figure 4-2. Interfacing to the DRAM Buffer

Internal Oscillator

A crystal or an external signal can be used to control the internal oscillator. If an external signal is used, it must be connected to XTALI, and XTALO should not be connected. Refer to Figure 4-3 below.

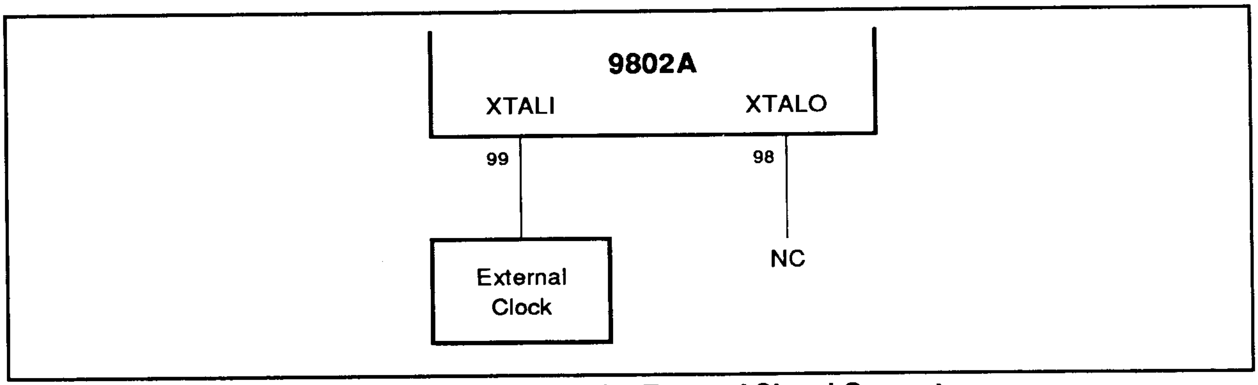


Figure 4-3. Connection for External Signal Generator

If a crystal is used, the external components should be oriented on the printed circuit board to minimize the length of the traces in the circuit. Figure 4-4 is the recommended circuit configuration when using a crystal. Note that this configuration requires a fundamental mode crystal. Typical values are shown. They will vary slightly with frequency and the distributed capacitance of the circuit board. Some adjustment may be necessary to find the optimum values. Resistance is given in ohms.

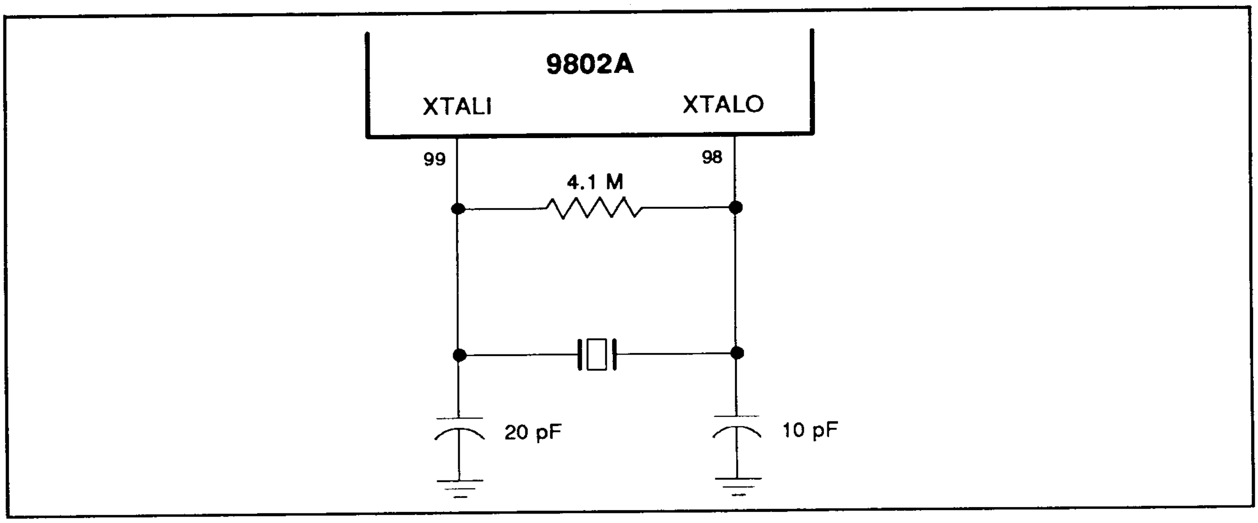


Figure 4-4. Connection for Crystal Oscillator

—This Page Intentionally Left Blank—

ECC Examples

The following examples demonstrate the various ECC modes of the 9802A. The memory configuration shown in Table 5-1 below is used for Examples 1 and 2 in this section. The addresses shown at left are the starting addresses of the following row of data in the buffer (in hex). The addresses increase from left to right. For example, refer to the first line of the table: 00 is at address 010000, 23 is at address 010001, 18 is at address 010002, and so forth. C7 is at address 01000C, and the last entry is at address 01000F.

Table 5-1. Memory Configuration — ECC Examples 1 and 2

Address	Data															
010000	00	23	18	CC	E9	62	7B	87	08	09	35	36	C7	..	..	..
010010	12	23	34	45	56	67	78	89	9A	AB	BC	CD	..	..	..	..
010020	01	23	45	67	89	AB	CD	EF	ED	CB	A9	87	..	..	..	..
010030	11	22	33	44	55	66	77	88	99	AA	BB	CC	..	..	..	..
0101F0	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	0F
0103F0	9A	9B	9C	9D	9E	9F	A0	A1	A2	A3	A4	A5	..	0D	..	21
010800	10	11	12	13	14	37	16	97	18	..	1A	..	1C	..	..	..
0109F0	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	1F
010BF0	..	..	..	..	..	..	..	..	..	..	..	1B	..	1D	..	9F
011000	20	21	22	23	24	12	26	A7	28	..	2A	..	2C	..	..	..
0111F0	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	2F
0113F0	..	..	..	..	..	..	..	..	..	..	..	2B	..	2D	..	AF
011800	30	31	32	33	34	24	36	B7	38	..	3A	..	3C	..	..	..
0119F0	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	3F
011BF0	..	..	..	..	..	..	..	..	..	..	..	3B	..	3D	..	BF
012000	40	41	42	43	44	C1	46	C7	48	..	4A	..	4C	..	..	..
0121F0	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	4F
012800	50	51	52	53	54	D5	56	D7	58	..	5A	..	5C	..	..	..
0129F0	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	5F
013000	60	61	62	63	64	A5	66	E7	68	..	6A	..	6C	..	..	..
0131F0	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	6F
013800	F0	F1	F2	F3	F4	F5	F6	F7	F8	..	FA	..	FC	..	..	..
0139F0	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	FF
10F800	12	34	56	78	9A	BC	DE	F0	..	..	..	..	..	..	..	..

ECC Example 1 — Parity Generation
(all addressing modes)

The following example demonstrates parity generation using the ECC processor for an ECC matrix that is 8 x 512 with each row beginning on a 2048 byte boundary. The polynomial x^2+3x+2 is used to generate the parity. All four addressing modes of the ECC unit are demonstrated and the registers are written with the values indicated in Table 5-2 below. In this table, "b" means that the value of this bit is listed in Table 5-3. The memory configuration for Example 1 is shown in Table 5-1 above.

Table 5-2. ECC Register Values — Example 1

Register	Value	Base
ECC GF(256) Feedback	87	hex
ECC Redundancy Register	02	hex
ECC Polynomial Stack Register	03 02 00 00 00 00 00 00	hex
ECC Row/Column Size Register	08	hex
ECC Matrix Size Register	0200	hex
ECC Source Address Register	010000	hex
ECC Destination Address Register	830000	hex
Byte Increment Register	000800	hex
ECC Command Register: DESTINATION INCREMENT/DECREMENT SELECT (bit 4) ECC OPERATION SELECT (bit 3) WRITE/XOR SELECT (bit 2) ADDRESS MODE SELECT (bits 1, 0)	00X010bb 0 1 0 as described below	binary

The ADDRESS MODE bits of the ECC Command Register determine the addressing mode used by the ECC unit. Refer to Table 5-3 below.

Table 5-3. ECC Unit Address Mode Selection — Example 1

ADDRESS MODE	Description
00	row oriented
01	column oriented
10	column oriented, XOR 2
11	column oriented, XOR 4

Table 5-4 describes the sequence of buffer memory accesses which the 9802A uses to carry out parity generation. The first column gives the operation to be performed on the memory: read or write. The next four columns describe the buffer memory accesses for each of the four possible addressing modes. The double horizontal lines indicate the boundaries between one column or row in the matrix and another column or row. In each case, 8 bytes are read (as determined by the ECC Row/Column Size Register) and 2 bytes are written (as determined by the ECC Redundancy Register).

Table 5-4. Buffer Memory Access Sequence — Example 1

Operation	Data Format			
	row oriented address/data	column oriented address/data	column oriented, XOR 2 address/data	column oriented, XOR 4 address/data
Read	010000/00	010000/00	010000/00	010000/00
Read	010001/23	010800/10	010002/18	010004/E9
Read	010002/18	011000/20	010800/10	010800/10
Read	010003/CC	011800/30	010802/12	010804/14
Read	010004/E9	012000/40	011000/20	011000/20
Read	010005/62	012800/50	011002/22	011004/24
Read	010006/7B	013000/60	011800/30	011800/30
Read	010007/87	013800/F0	011802/32	011804/34
Write	830000/B5	830000/C0	830000/1F	830000/BD
Write	830001/35	830800/40	830002/05	830004/50
Read	010800/10	010001/23	010001/23	010001/23
Read	010801/11	010801/11	010003/CC	010005/62
Read	010802/12	011001/21	010801/11	010801/11
Read	010803/13	011801/31	010803/13	010805/37
Read	010804/14	012001/41	011001/21	011001/21
Read	010805/37	012801/51	011003/23	011005/12
Read	010806/16	013001/61	011801/31	011801/31
Read	010807/97	013801/F1	011803/33	011805/24
Write	830800/E5	830001/D5	830001/FB	830001/12
Write	830801/47	830801/77	830003/16	830005/53
Read	011000/20	010002/18	010006/7B	010006/7B
Read	011001/21	010802/12	010004/E9	010002/18
Read	011002/22	011002/22	010806/16	010806/16
Read	011003/23	011802/32	010804/14	010802/12
Read	011004/24	012002/42	011006/26	011006/26
Read	011005/12	012802/52	011004/24	011002/22
Read	011006/26	013002/62	011806/36	011806/36
Read	011007/A7	013802/F2	011804/34	011802/32
Write	831000/30	830002/C4	830006/6F	830006/CD
Write	831001/87	830802/5E	830004/FF	830002/AA

(continued)

Table 5-4. Buffer Memory Access Sequence — Example 1 (continued)

Operation	Data Format			
	row oriented address/data	column oriented address/data	column oriented, XOR 2 address/data	column oriented, XOR 4 address/data
Read	011800/30	010003/CC	010007/87	010007/87
Read	011801/31	010803/13	010005/62	010003/CC
Read	011802/32	011003/23	010807/97	010807/97
Read	011803/33	011803/33	010805/37	010803/13
Read	011804/34	012003/43	011007/A7	011007/A7
Read	011805/24	012803/53	011005/12	011003/23
Read	011806/36	013003/63	011807/B7	011807/B7
Read	011807/B7	013803/F3	011805/24	011803/33
Write	831800/FA	830003/A7	830007/20	830007/C9
Write	831801/6B	830803/E8	830005/43	830003/06
Read	012000/40	010004/E9	010008/08	010008/08
Read	012001/41	010804/14	01000A/35	01000C/C7
Read	012002/42	011004/24	010808/18	010808/18
Read	012003/43	011804/34	01080A/1A	01080C/1C
Read	012004/44	012004/44	011008/28	011008/28
Read	012005/C1	012804/54	01100A/2A	01100C/2C
Read	012006/46	013004/64	011808/38	011808/38
Read	012007/C7	013804/F4	01180A/3A	01180C/3C
Write	832000/81	830004/D1	830008/9D	830008/B9
Write	832001/85	830804/BC	83000A/A2	83000C/72
...				
Read	10F800/12	0101FF/0F	0103FF/21	0103FF/21
Read	10F801/34	0109FF/1F	0103FD/0D	0103FB/A5
Read	10F802/56	0111FF/2F	010BFF/9F	010BFF/9F
Read	10F803/78	0119FF/3F	010BFD/1D	010BFB/1B
Read	10F804/9A	0121FF/4F	0113FF/AF	0113FF/AF
Read	10F805/BC	0129FF/5F	0113FD/2D	0113FB/2B
Read	10F806/DE	0131FF/6F	011BFF/BF	011BFF/BF
Read	10F807/F0	0139FF/FF	011BFD/3D	011BFB/3B
Write	92F800/D6	8301FF/F4	8303FF/F6	8303FF/18
Write	92F801/D6	8309FF/74	8303FD/58	8303FB/18

ECC Example 2 — Parity Generation
(row oriented mode)

The following example demonstrates parity generation using row oriented ECC addressing mode for an ECC matrix that is 64 x 12 with each row beginning on a 16 byte boundary. Since each row is 12 bytes long and on a 16 byte boundary, there is a 4 byte gap between rows. Parity bytes are written in this gap. When the ECC operation is complete, there will be a 1 Kbyte frame of data in memory consisting of 16 byte blocks, each arranged as 12 bytes of data followed by 4 bytes of parity. The polynomial $x^4 + 196x^3 + 206x^2 + 15x + 4$ is used to generate the parity, and the registers are written with the values indicated in Table 5-5 below. The memory configuration for this example is given in Table 5-1 above.

Table 5-5. ECC Register Values — Example 2

Register	Value	Base
ECC Redundancy Register	04	hex
ECC Polynomial Stack Register	C4 CE 0F 04 00 00 00 00	hex
ECC Row/Column Size Register	0C	hex
ECC Matrix Size Register	0040	hex
ECC Source Address Register	010000	hex
ECC Destination Address Register	01000C	hex
Byte Increment Register	000010	hex
ECC Command Register: DESTINATION INCREMENT/DECREMENT SELECT (bit 4) ECC OPERATION SELECT (bit 3) WRITE/XOR SELECT (bit 2) ADDRESS MODE SELECT (bits 1, 0)	00X01000 0 1 0 00	binary

Table 5-6 describes the sequence of buffer memory accesses which the 9802A uses to carry out parity generation. The first column gives the operation to be performed on the memory: read or write. The next four columns describe the buffer memory accesses for each of the four possible addressing modes. The double horizontal lines indicate the boundaries between one column or row in the matrix and another column or row. In each case, 12 bytes are read (as determined by the ECC Row/Column Size Register) and 4 bytes are written (as determined by the ECC Redundancy Register).

Table 5-6. Buffer Memory Access Sequence — Example 2

Operation	Address/Data
Read	010000/00
Read	010001/23
Read	010002/18
Read	010003/CC
Read	010004/E9
Read	010005/62
Read	010006/7B
Read	010007/87
Read	010008/08
Read	010009/09
Read	01000A/35
Read	01000B/36
Write	01000C/AE
Write	01000D/EC
Write	01000E/A7
Write	01000F/67
Read	010010/12
Read	010011/23
Read	010012/34
Read	010013/45
Read	010014/56
Read	010015/67
Read	010016/78
Read	010017/89
Read	010018/9A
Read	010019/AB
Read	01001A/BC
Read	01001B/CD
Write	01001C/90
Write	01001D/29
Write	01001E/F3
Write	01001F/8A
Read	010020/01
Read	010021/23
Read	010022/45
Read	010023/67
Read	010024/89
Read	010025/AB
Read	010026/CD
Read	010027/EF
Read	010028/ED
Read	010029/CB
Read	01002A/A9
Read	01002B/87
Write	01002C/FD
Write	01002D/E0
Write	01002E/F6
Write	01002F/E3
Read	010030/11
Read	010031/22
Read	010032/33
Read	010033/44
Read	010034/55
Read	010035/66
Read	010036/77
Read	010037/88
Read	010038/99
Read	010039/AA
Read	01003A/BB
Read	01003B/CC
Write	01003C/D1
Write	01003D/71
Write	01003E/4D
Write	01003F/21
...	...
Read	0103F0/9A
Read	0103F1/9B
Read	0103F2/9C
Read	0103F3/9D
Read	0103F4/9E
Read	0103F5/9F
Read	0103F6/A0
Read	0103F7/A1
Read	0103F8/A2
Read	0103F9/A3
Read	0103FA/A4
Read	0103FB/A5
Write	0103FC/E7
Write	0103FD/C2
Write	0103FE/AE
Write	0103FF/8B

ECC Example 3 — Syndrome Generation and Error Correction

(column oriented mode)

The following is an example of column oriented ECC syndrome generation and error correction. For this example, assume that the frame from the ECC Example 1 using column oriented addressing is read back with the fifth row being in error. When the frame was read back it was put in memory via DMA (linear mode) at addresses 020000 to 0213FF (there are 10 rows: 8 data rows and 2 parity rows). The resulting buffer memory is shown in Table 5-7 below. Note that the bytes at 020800 to 0209FF are different from those originally "written". (They were: 40 41 42 43 44 ... 4F.)

Table 5-7. Memory Configuration — Examples 3 and 4

Address	Data															
020000	00	23	18	CC	E9	..	..	..	..	..	..	..	..	..	..	..
0201F0	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	0F
020200	10	11	12	13	14	..	..	..	..	..	..	..	..	..	..	..
0203F0	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	1F
020400	20	21	22	23	24	..	..	..	..	..	..	..	..	..	..	..
0205F0	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	2F
020600	30	31	32	33	34	..	..	..	..	..	..	..	..	..	..	..
0207F0	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	3F
020800	***00***	***00***	***42***	***FF***	***FF***	***..***	***..***	***..***	***..***	***..***	***..***	***..***	***..***	***..***	***..***	***..***
0209F0	***..***	***..***	***..***	***..***	***..***	***..***	***..***	***..***	***..***	***..***	***..***	***..***	***..***	***..***	***..***	***FF***
020A00	50	51	52	53	54	..	..	..	..	..	..	..	..	..	..	..
020BF0	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	5F
020C00	60	61	62	63	64	..	..	..	..	..	..	..	..	..	..	..
020DF0	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	6F
020E00	F0	F1	F2	F3	F4	..	..	..	..	..	..	..	..	..	..	..
020FF0	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	FF
021000	C0	D5	C4	A7	D1	..	..	..	..	..	..	..	..	..	..	..
0211F0	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	F4
021200	40	77	5E	E8	BC	..	..	..	..	..	..	..	..	..	..	..
0213F0	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	74

To check and correct data, use the register settings in Table 5-8 to compute the syndromes for the block read. Note that the values for the GF(256) Feedback Register, ECC Redundancy Register, ECC Polynomial Stack Register, ECC Row/Column Size Register and ECC Matrix Size Register are the same as they were when the parity was generated (in Example 1).

Table 5-8. ECC Register Values — Example 3

Register	Value	Base
GF(256) Feedback Register	87	hex
ECC Redundancy Register	02	hex
ECC Polynomial Stack Register	03 02 00 00 00 00 00 00	hex
ECC Row/Column Size Register	08	hex
ECC Matrix Size Register	0200	hex
ECC Source Address Register	020000	hex
ECC Destination Address Register	021000	hex
Byte Increment Register	000200	hex
ECC Command Register: DESTINATION INCREMENT/DECREMENT SELECT (bit 4) ECC OPERATION SELECT (bit 3) WRITE/XOR SELECT (bit 2) ADDRESS MODE SELECT (bit 1, 0)	00X01101 0 1 1 (XOR into the parity) 01	binary

The syndromes are computed in the same manner as parity generation except the computed ECC results are XOR'd with the data in memory. The following memory data results at the destination (Table 5-9).

Table 5-9. Resulting Memory Configuration — Example 3

Address	Data															
021000	5B	44	00	AB	F6	..	..	..	..	..	..	..	..	..	..	..
0211F0	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	2F
021200	1B	05	00	17	4D	..	..	..	..	..	..	..	..	..	..	..
0213F0	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	9F

Since the ECC processor wrote non-zero values to memory, the NON-ZERO bit in the Status Register was set. This indicates that the syndromes are non-zero so there must be an error. Typically, the CRC would also be in error on a bad row and this information can be used to generate the error correction vector to correct the data. In this example the only error is in the fifth row of a ten row frame and the original redundancy was 2. Using this information and the appropriate seed tables or equations, the error correction vector can be computed to be: 15 and 00 (hex). To correct the row in error, the registers are set up as shown in Table 5-10.

Table 5-10. ECC Register Values for Correction — Example 3

Register	Value	Base
ECC Redundancy Register	01 (always 1 for correction)	hex
ECC Polynomial Stack Register	15 00 00 00 00 00 00 00	hex
ECC Row/Column Size Register	02 (the original redundancy)	hex
ECC Matrix Size Register	0200	hex
ECC Source Address Register	021000 (starting syndrome address)	hex
ECC Destination Address Register	020800 (row in error)	hex
Byte Increment Register	000200	hex
ECC Command Register: DESTINATION INCREMENT/DECREMENT SELECT (bit 4) ECC OPERATION SELECT (bit 3) WRITE/XOR SELECT (bit 2) ADDRESS MODE SELECT (bit 1, 0)	00X00101 0 0 (error correction operation) 1 (XOR to get corrected data) 01	binary

Table 5-11 describes the sequence of buffer memory accesses which the 9802A uses to carry out parity generation. The first column gives the operation to be performed on the memory: read or write. The next four columns describe the buffer memory accesses for each of the four possible addressing modes. The double horizontal lines indicate the boundaries between byte corrections. Note that for each corrected byte 2 syndrome bytes are read (2 being the original redundancy), the original data is read, and then the corrected data is written.

Table 5-11. Buffer Memory Access Sequence — Example 3

Operation	Address/Data (hex)
Read	021000/5B
Read	021200/1B
Read	020800/00
Write	020800/40
Read	021001/44
Read	021201/05
Read	020801/00
Write	020801/41
Read	021002/00
Read	021202/00
Read	020802/42
Write	020802/42
Read	021003/AB
Read	021203/17
Read	020803/FF
Write	020803/43
Read	021004/F6
Read	021204/4D
Read	020804/FF
Write	020804/44
...	...
Read	0211FF/2F
Read	0213FF/9F
Read	0209FF/FF
Write	0209FF/4F

Now the row starting at hex 020800 contains the correct values.

ECC Example 4 — Syndrome Generation and Error Correction (Alternate Method) (column oriented mode)

The following is an example of column oriented ECC syndrome generation and error correction. This example is similar to the previous example except that the computed ECC results are written elsewhere in memory instead of being XOR'd with the parity rows in memory. For this example, use the frame from ECC Example 1 that was read back with the fifth row in error. (See Table 5-7 in ECC Example 3).

To check and correct data, use the register settings in Table 5-12 to compute the syndromes for the block read. Note that the values for the GF(256) Feedback Register, ECC Redundancy Register, ECC Polynomial Stack Register and ECC Matrix Size Register are the same as they were when the parity was generated (in Example 1).

Table 5-12. ECC Register Values — Example 4

Register	Value	Base
GF(256) Feedback Register	87	hex
ECC Redundancy Register	02	hex
ECC Polynomial Stack Register	03 02 00 00 00 00 00 00	hex
ECC Row/Column Size Register	0A	hex
ECC Matrix Size Register	0200	hex
ECC Source Address Register	020000	hex
ECC Destination Address Register	028000	hex
Byte Increment Register	000200	hex
ECC Command Register: DESTINATION INCREMENT/DECREMENT SELECT (bit 4) ECC OPERATION SELECT (bit 3) WRITE/XOR SELECT (bit 2) ADDRESS MODE SELECT (bit 1, 0)	00X01101 0 1 0 (Write result to destination) 01	binary

The syndromes are computed in the same manner as parity generation except that the two parity rows are treated as data and the computed ECC results are written elsewhere (to 028000). The following memory data results at the destination (Table 5-13).

Table 5-13. Resulting Memory Configuration — Example 4

Address	Data															
028000	2B	54	00	66	9C	..	..	..	..	..	..	..	..	..	..	..
0281F0	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	EB
028200	6B	15	00	DA	27	..	..	..	..	..	..	..	..	..	..	..
0283F0	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	5B

Since the ECC processor wrote non-zero values to memory, the NON-ZERO bit in the Status Register was set. This indicates that the syndromes are non-zero so there must be an error. Typically, the CRC would also be in error on a bad row and this information can be used to generate the error correction vector to correct the data. In this example the only error is in the fifth row of a ten row frame and the original redundancy was 2. Using this information and the appropriate seed tables or equations, the error correction vector can be computed to be: 00 and A3 (hex). To correct the row in error, the registers are set up as shown in Table 5-14.

Table 5-14. ECC Register Values for Correction — Example 4

Register	Value	Base
ECC Redundancy Register	01 (always 1 for correction)	hex
ECC Polynomial Stack Register	00 A3 00 00 00 00 00 00	hex
ECC Row/Column Size Register	02 (the original redundancy)	hex
ECC Matrix Size Register	0200	hex
ECC Source Address Register	028000 (starting syndrome address)	hex
ECC Destination Address Register	020800 (row in error)	hex
Byte Increment Register	000200	hex
ECC Command Register: DESTINATION INCREMENT/DECREMENT SELECT (bit 4) ECC OPERATION SELECT (bit 3) WRITE/XOR SELECT (bit 2) ADDRESS MODE SELECT (bit 1, 0)	00X00101 0 0 (error correction operation) 1 (XOR to get corrected data) 01	binary

Table 5-15 describes the sequence of buffer memory accesses which the 9802A uses to carry out parity generation. The first column gives the operation to be performed on the memory: read or write. The next four columns describe the buffer memory accesses for each of the four possible addressing modes. The double horizontal lines indicate the boundaries between byte corrections. Note that for each corrected byte 2 syndrome bytes are read (2 being the original redundancy), the original data is read, and then the corrected data is written.

Table 5-15. Buffer Memory Access Sequence — Example 4

Operation	Address/Data (hex)
Read	028000/2B
Read	028200/6B
Read	020800/00
Write	020800/40
Read	028001/54
Read	028201/15
Read	020801/00
Write	020801/41
Read	028002/00
Read	028202/00
Read	020802/42
Write	020802/42
Read	028003/66
Read	028203/DA
Read	020803/FF
Write	020803/43
Read	028004/9C
Read	028204/27
Read	020804/FF
Write	020804/44
...	...
Read	0281FF/EB
Read	0283FF/5B
Read	0209FF/FF
Write	0209FF/4F

Now the row starting at hex 020800 contains the correct values.

ECC Example 5 — Using the ECC Unit to Copy Buffer Data

(Without using the Byte Increment Register)

This procedure will copy a block of buffer data by using the ECC Unit and not using the Byte Increment Register. The values given in Table 5-16 will copy a data block of 100 (hex) bytes at addresses 020000–0200FF (hex) to addresses 030000–0300FF (hex). Use this method if the Byte Increment Register is unavailable because it is already programmed for a DMA operation. (The Byte Increment Register is not saved.)

Table 5-16. Register Values for Block Moves —Example 5

Register	Value	Base
ECC Redundancy Register	01	hex
ECC Polynomial Stack Register	01 00 00 00 00 00 00 00	hex
ECC Row/Column Size Register	01	hex
ECC Matrix Size Register	0100	hex
ECC Source Address Register	020000	hex
ECC Destination Address Register	030000	hex
Byte Increment Register	not used	hex
ECC Command Register: DESTINATION INCREMENT/DECREMENT SELECT (bit 4) ECC OPERATION SELECT (bit 3) WRITE/XOR SELECT (bit 2) ADDRESS MODE SELECT (bit 1, 0)	00X01001 0 1 0 01	binary

Table 5-17 gives the sequence of memory operations the 9802A will perform to copy the block of data.

Table 5-17. Buffer Memory Access Sequence — Example 5

Operation	Address (hex)
Read	020000
Write	030000
Read	020001
Write	030001
Read	020002
Write	030002
...	...
Read	0200FE
Write	0300FE
Read	0200FF
Write	0300FF

ECC Example 6 — Using the ECC Unit to Copy Buffer Data

(Using the Byte Increment Register or ECC Byte Increment Register)

Example 6 is similar to the previous example. It executes much more quickly than Example 5, but since it uses the Byte Increment Register, it is possible only if no DMA operation using the Byte Increment Register is scheduled. It moves 8 bytes at a time, and the number of bytes moved must be a multiple of 8. This example will copy a data block of 100 (hex) bytes at addresses 020000–0200FF (hex) to addresses 030000–0300FF (hex).

Table 5-18. Register Values for Block Moves — Example 6

Register	Value	Base
ECC Redundancy Register	08	hex
ECC Polynomial Stack Register	00 00 00 00 00 00 00 01	hex
ECC Row/Column Size Register	08	hex
ECC Matrix Size Register	0020	hex
ECC Source Address Register	020000	hex
ECC Destination Address Register	030000	hex
Byte Increment Register or ECC Byte Incr Reg	08	hex
ECC Command Register (using Byte Incr Reg) ECC Command Register (using ECC Byte Incr Reg)	00X01000 10X01000	binary binary
ECC Command Register bit definitions: DESTINATION INCREMENT/DECREMENT SELECT (bit 4) ECC OPERATION SELECT (bit 3) WRITE/XOR SELECT (bit 2) ADDRESS MODE SELECT (bit 1, 0)	 0 1 0 00	binary

Table 5-19 gives the sequence of buffer accesses the 9802A will perform to copy the block of data.

Table 5-19. Buffer Memory Access Sequence — Example 6

Operation	Address (hex)
Read	020000
Read	020001
Read	020002
Read	020003
Read	020004
Read	020005
Read	020006
Read	020007
Write	030000
Write	030001
Write	030002
Write	030003
Write	030004
Write	030005
Write	030006
Write	030007
Read	020008
Read	020009
Read	02000A
...	...
Write	0300FE
Write	0300FF

DMA Examples

Below are four examples of DMA operation. For all examples, assume that data is being transferred via DMA from the DMA device and that the bytes being transferred in are ordered from 00 to FF. The registers are written with the values given in Table 6-1 below. The resulting buffer memory for each example is given. Note: the letter "b" in Table 6-1 refers to the values given in following tables.

Table 6-1. Register Values — DMA Examples 1–4

Register	Value	Base
DMA Address Register	100000	hex
DMA Transfer Length Register	as described below	
Byte Increment Register	000004	hex
Row Increment Register	as described below	
DMA Configuration Register ROW/COLUMN COUNT SIZE SELECT (bits 1, 0)	XXXXXX00 00	binary
DMA Command Register ADDRESS INCREMENT AMOUNT SELECT (bit 7) COMPARE MODE ENABLE (bit 4) DIRECTION SELECT (bit 3) LINEAR/MATRIX MODE SELECT (bit 2)	b0X00bXX as described below 0 0 as described below	binary

Example 1 - Linear Mode (increment address by 1)

In this example, the DMA data is written to memory linearly with an address increment of one between bytes. The values of the registers not specified in Table 6-1 are given in Table 6-2 below. The contents of memory where the DMA data is written is given in Table 6-3 below.

Table 6-2. Register Values — Example 1

Register	Value	Base
DMA Transfer Length Register	0100	hex
Row Increment Register	not used	
DMA Command Register ADDRESS INCREMENT AMOUNT SELECT (bit 7) LINEAR/MATRIX MODE SELECT (bit 2)	00X000XX 0 0	binary

Table 6-3. Memory Contents — Example 1

Address	Data															
100000	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
100010	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F
100020	20	21	22	23	24	25	26	27	28	29	2A	2B	2C	2D	2E	2F
100030	30	31	32	33	34	35	36	37	38	39	3A	3B	3C	3D	3E	3F
100040	40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F
100050	50	51	52	53	54	55	56	57	58	59	5A	5B	5C	5D	5E	5F
100060	60	61	62	63	64	65	66	67	68	69	6A	6B	6C	6D	6E	6F
100070	70	71	72	73	74	75	76	77	78	79	7A	7B	7C	7D	7E	7F
100080	80	81	82	83	84	85	86	87	88	89	8A	8B	8C	8D	8E	8F
100090	90	91	92	93	94	95	96	97	98	99	9A	9B	9C	9D	9E	9F
1000A0	A0	A1	A2	A3	A4	A5	A6	A7	A8	A9	AA	AB	AC	AD	AE	AF
1000B0	B0	B1	B2	B3	B4	B5	B6	B7	B8	B9	BA	BB	BC	BD	BE	BF
1000C0	C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	CA	CB	CC	CD	CE	CF
1000D0	D0	D1	D2	D3	D4	D5	D6	D7	D8	D9	DA	DB	DC	DD	DE	DF
1000E0	E0	E1	E2	E3	E4	E5	E6	E7	E8	E9	EA	EB	EC	ED	EE	EF
1000F0	F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	FA	FB	FC	FD	FE	FF

Example 2 - Linear Mode (increment address by Byte Increment Register)

In this example, DMA data is transferred linearly using the value in the Byte Increment Register to increment the address between bytes. The values of the registers not specified in Table 6-1 are given in Table 6-4 below. The contents of memory where the DMA data is written is given in Table 6-5.

Table 6-4. Register Values — Example 2

Register	Value	Base
DMA Transfer Length Register	0100	hex
Row Increment Register	not used	
DMA Command Register ADDRESS INCREMENT AMOUNT SELECT (bit 7) LINEAR/MATRIX MODE SELECT (bit 2)	10X000XX 1 0	binary

Table 6-5. Memory Contents — Example 2

Address	Data
100000	00 .. .. .. 01 .. .. .. 02 .. .. .. 03 .. .. ..
100010	04 .. .. .. 05 .. .. .. 06 .. .. .. 07 .. .. ..
100020	08 .. .. .. 09 .. .. .. 0A .. .. .. 0B .. .. ..
100030	0C .. .. .. 0D .. .. .. 0E .. .. .. 0F .. .. ..
100040	10 .. .. .. 11 .. .. .. 12 .. .. .. 13 .. .. ..
100050	14 .. .. .. 15 .. .. .. 16 .. .. .. 17 .. .. ..
100060	18 .. .. .. 19 .. .. .. 1A .. .. .. 1B .. .. ..
100070	1C .. .. .. 1D .. .. .. 1E .. .. .. 1F .. .. ..
100080	20 .. .. .. 21 .. .. .. 22 .. .. .. 23 .. .. ..
100090	24 .. .. .. 25 .. .. .. 26 .. .. .. 27 .. .. ..
1000A0	28 .. .. .. 29 .. .. .. 2A .. .. .. 2B .. .. ..
1000B0	2C .. .. .. 2D .. .. .. 2E .. .. .. 2F .. .. ..
1000C0	30 .. .. .. 31 .. .. .. 32 .. .. .. 33 .. .. ..
1000D0	34 .. .. .. 35 .. .. .. 36 .. .. .. 37 .. .. ..
1000E0	38 .. .. .. 39 .. .. .. 3A .. .. .. 3B .. .. ..
1000F0	3C .. .. .. 3D .. .. .. 3E .. .. .. 3F .. .. ..
100100	40 .. .. .. 41 .. .. .. 42 .. .. .. 43 .. .. ..
100110	44 .. .. .. 45 .. .. .. 46 .. .. .. 47 .. .. ..
100120	48 .. .. .. 49 .. .. .. 4A .. .. .. 4B .. .. ..
100130	4C .. .. .. 4D .. .. .. 4E .. .. .. 4F .. .. ..
100140	50 .. .. .. 51 .. .. .. 52 .. .. .. 53 .. .. ..
100150	54 .. .. .. 55 .. .. .. 56 .. .. .. 57 .. .. ..
100160	58 .. .. .. 59 .. .. .. 5A .. .. .. 5B .. .. ..
100170	5C .. .. .. 5D .. .. .. 5E .. .. .. 5F .. .. ..
100180	60 .. .. .. 61 .. .. .. 62 .. .. .. 63 .. .. ..
100190	64 .. .. .. 65 .. .. .. 66 .. .. .. 67 .. .. ..
1001A0	68 .. .. .. 69 .. .. .. 6A .. .. .. 6B .. .. ..
1001B0	6C .. .. .. 6D .. .. .. 6E .. .. .. 6F .. .. ..
1001C0	70 .. .. .. 71 .. .. .. 72 .. .. .. 73 .. .. ..
1001D0	74 .. .. .. 75 .. .. .. 76 .. .. .. 77 .. .. ..
1001E0	78 .. .. .. 79 .. .. .. 7A .. .. .. 7B .. .. ..
1001F0	7C .. .. .. 7D .. .. .. 7E .. .. .. 7F .. .. ..
100200	80 .. .. .. 81 .. .. .. 82 .. .. .. 83 .. .. ..
100210	84 .. .. .. 85 .. .. .. 86 .. .. .. 87 .. .. ..
100220	88 .. .. .. 89 .. .. .. 8A .. .. .. 8B .. .. ..
100230	8C .. .. .. 8D .. .. .. 8E .. .. .. 8F .. .. ..
100240	90 .. .. .. 91 .. .. .. 92 .. .. .. 93 .. .. ..
100250	94 .. .. .. 95 .. .. .. 96 .. .. .. 97 .. .. ..
100260	98 .. .. .. 99 .. .. .. 9A .. .. .. 9B .. .. ..
100270	9C .. .. .. 9D .. .. .. 9E .. .. .. 9F .. .. ..
100280	A0 .. .. .. A1 .. .. .. A2 .. .. .. A3 .. .. ..
100290	A4 .. .. .. A5 .. .. .. A6 .. .. .. A7 .. .. ..
1002A0	A8 .. .. .. A9 .. .. .. AA .. .. .. AB .. .. ..
1002B0	AC .. .. .. AD .. .. .. AE .. .. .. AF .. .. ..
1002C0	B0 .. .. .. B1 .. .. .. B2 .. .. .. B3 .. .. ..
1002D0	B4 .. .. .. B5 .. .. .. B6 .. .. .. B7 .. .. ..
1002E0	B8 .. .. .. B9 .. .. .. BA .. .. .. BB .. .. ..
1002F0	BC .. .. .. BD .. .. .. BE .. .. .. BF .. .. ..
100300	C0 .. .. .. C1 .. .. .. C2 .. .. .. C3 .. .. ..
100310	C4 .. .. .. C5 .. .. .. C6 .. .. .. C7 .. .. ..
100320	C8 .. .. .. C9 .. .. .. CA .. .. .. CB .. .. ..
100330	CC .. .. .. CD .. .. .. CE .. .. .. CF .. .. ..
100340	D0 .. .. .. D1 .. .. .. D2 .. .. .. D3 .. .. ..
100350	D4 .. .. .. D5 .. .. .. D6 .. .. .. D7 .. .. ..
100360	D8 .. .. .. D9 .. .. .. DA .. .. .. DB .. .. ..
100370	DC .. .. .. DD .. .. .. DE .. .. .. DF .. .. ..
100380	E0 .. .. .. E1 .. .. .. E2 .. .. .. E3 .. .. ..
100390	E4 .. .. .. E5 .. .. .. E6 .. .. .. E7 .. .. ..
1003A0	E8 .. .. .. E9 .. .. .. EA .. .. .. EB .. .. ..
1003B0	EC .. .. .. ED .. .. .. EE .. .. .. EF .. .. ..
1003C0	F0 .. .. .. F1 .. .. .. F2 .. .. .. F3 .. .. ..
1003D0	F4 .. .. .. F5 .. .. .. F6 .. .. .. F7 .. .. ..
1003E0	F8 .. .. .. F9 .. .. .. FA .. .. .. FB .. .. ..
1003F0	FC .. .. .. FD .. .. .. FE .. .. .. FF .. .. ..

Example 3 - Matrix Mode (increment address by 1)

In this example, DMA data is written to memory using matrix mode with a column to column increment of 1. A row of bytes (specified in the Transfer Length Register) is written to memory sequentially, then *r* memory locations are skipped (where *r* is the value in the Row Increment Register) and the next row is started. The column size is specified in the Transfer Length Register. The values of the registers not specified in Table 6-1 are given in Table 6-6 below. The contents of memory where that DMA data is written is given in Table 6-7.

Table 6-6. Register Values — Example 3

Register	Value	Base
DMA Transfer Length Register	0220 (8 rows of 32 columns)	hex
Row Increment Register	03FFE1	hex
DMA Command Register ADDRESS INCREMENT AMOUNT SELECT (bit 7) LINEAR/MATRIX MODE SELECT (bit 2)	00X001XX 0 1	binary

Table 6-7. Memory Contents — Example 3

Address	Data															
100000	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
100010	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F
140000	20	21	22	23	24	25	26	27	28	29	2A	2B	2C	2D	2E	2F
140010	30	31	32	33	34	35	36	37	38	39	3A	3B	3C	3D	3E	3F
180000	40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F
180010	50	51	52	53	54	55	56	57	58	59	5A	5B	5C	5D	5E	5F
1C0000	60	61	62	63	64	65	66	67	68	69	6A	6B	6C	6D	6E	6F
1C0010	70	71	72	73	74	75	76	77	78	79	7A	7B	7C	7D	7E	7F
200000	80	81	82	83	84	85	86	87	88	89	8A	8B	8C	8D	8E	8F
200010	90	91	92	93	94	95	96	97	98	99	9A	9B	9C	9D	9E	9F
240000	A0	A1	A2	A3	A4	A5	A6	A7	A8	A9	AA	AB	AC	AD	AE	AF
240010	B0	B1	B2	B3	B4	B5	B6	B7	B8	B9	BA	BB	BC	BD	BE	BF
280000	C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	CA	CB	CC	CD	CE	CF
280010	D0	D1	D2	D3	D4	D5	D6	D7	D8	D9	DA	DB	DC	DD	DE	DF
2C0000	E0	E1	E2	E3	E4	E5	E6	E7	E8	E9	EA	EB	EC	ED	EE	EF
2C0010	F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	FA	FB	FC	FD	FE	FF

Example 4 - Matrix Mode (increment address by Byte Increment Register)

In this example the DMA data is written to memory using matrix mode with a column to column increment of *n*, where *n* is the value in the Byte Increment Register. A row of bytes (specified in the Transfer Length Register) is written to memory in every *nth* location, then *r* memory locations are skipped (where *r* is the value in the Row Increment Register) and a new row is started. The number of rows in the transfer are specified by the column size in the Transfer Length Register. The values of the registers not specified in Table 6-1 are given in Table 6-8 below, The contents of memory where the DMA data is written is given in Table 6-9.

Table 6-8. Register Values — Example 4

Register	Value	Base
DMA Transfer Length Register	0220 (8 rows of 32 columns)	hex
Row Increment Register	03FF84	hex
DMA Command Register ADDRESS INCREMENT AMOUNT SELECT (bit 7) LINEAR/MATRIX MODE SELECT (bit 2)	10X001XX 1 1	binary

Table 6-9. Memory Contents — Example 4

Address	Data
100000	00 .. .. .. 01 .. .. .. 02 .. .. .. 03 .. .. ..
100010	04 .. .. .. 05 .. .. .. 06 .. .. .. 07 .. .. ..
100020	08 .. .. .. 09 .. .. .. 0A .. .. .. 0B .. .. ..
100030	0C .. .. .. 0D .. .. .. 0E .. .. .. 0F .. .. ..
100040	10 .. .. .. 11 .. .. .. 12 .. .. .. 13 .. .. ..
100050	14 .. .. .. 15 .. .. .. 16 .. .. .. 17 .. .. ..
100060	18 .. .. .. 19 .. .. .. 1A .. .. .. 1B .. .. ..
100070	1C .. .. .. 1D .. .. .. 1E .. .. .. 1F .. .. ..
140000	20 .. .. .. 21 .. .. .. 22 .. .. .. 23 .. .. ..
140010	24 .. .. .. 25 .. .. .. 26 .. .. .. 27 .. .. ..
140020	28 .. .. .. 29 .. .. .. 2A .. .. .. 2B .. .. ..
140030	2C .. .. .. 2D .. .. .. 2E .. .. .. 2F .. .. ..
140040	30 .. .. .. 31 .. .. .. 32 .. .. .. 33 .. .. ..
140050	34 .. .. .. 35 .. .. .. 36 .. .. .. 37 .. .. ..
140060	38 .. .. .. 39 .. .. .. 3A .. .. .. 3B .. .. ..
140070	3C .. .. .. 3D .. .. .. 3E .. .. .. 3F .. .. ..
180000	40 .. .. .. 41 .. .. .. 42 .. .. .. 43 .. .. ..
180010	44 .. .. .. 45 .. .. .. 46 .. .. .. 47 .. .. ..
180020	48 .. .. .. 49 .. .. .. 4A .. .. .. 4B .. .. ..
180030	4C .. .. .. 4D .. .. .. 4E .. .. .. 4F .. .. ..
180040	50 .. .. .. 51 .. .. .. 52 .. .. .. 53 .. .. ..
180050	54 .. .. .. 55 .. .. .. 56 .. .. .. 57 .. .. ..
180060	58 .. .. .. 59 .. .. .. 5A .. .. .. 5B .. .. ..
180070	5C .. .. .. 5D .. .. .. 5E .. .. .. 5F .. .. ..
1C0000	60 .. .. .. 61 .. .. .. 62 .. .. .. 63 .. .. ..
1C0010	64 .. .. .. 65 .. .. .. 66 .. .. .. 67 .. .. ..
1C0020	68 .. .. .. 69 .. .. .. 6A .. .. .. 6B .. .. ..
1C0030	6C .. .. .. 6D .. .. .. 6E .. .. .. 6F .. .. ..
1C0040	70 .. .. .. 71 .. .. .. 72 .. .. .. 73 .. .. ..
1C0050	74 .. .. .. 75 .. .. .. 76 .. .. .. 77 .. .. ..
1C0060	78 .. .. .. 79 .. .. .. 7A .. .. .. 7B .. .. ..
1C0070	7C .. .. .. 7D .. .. .. 7E .. .. .. 7F .. .. ..
200000	80 .. .. .. 81 .. .. .. 82 .. .. .. 83 .. .. ..
200010	84 .. .. .. 85 .. .. .. 86 .. .. .. 87 .. .. ..
200020	88 .. .. .. 89 .. .. .. 8A .. .. .. 8B .. .. ..
200030	8C .. .. .. 8D .. .. .. 8E .. .. .. 8F .. .. ..
200040	90 .. .. .. 91 .. .. .. 92 .. .. .. 93 .. .. ..
200050	94 .. .. .. 95 .. .. .. 96 .. .. .. 97 .. .. ..
200060	98 .. .. .. 99 .. .. .. 9A .. .. .. 9B .. .. ..
200070	9C .. .. .. 9D .. .. .. 9E .. .. .. 9F .. .. ..
240000	A0 .. .. .. A1 .. .. .. A2 .. .. .. A3 .. .. ..
240010	A4 .. .. .. A5 .. .. .. A6 .. .. .. A7 .. .. ..
240020	A8 .. .. .. A9 .. .. .. AA .. .. .. AB .. .. ..
240030	AC .. .. .. AD .. .. .. AE .. .. .. AF .. .. ..
240040	B0 .. .. .. B1 .. .. .. B2 .. .. .. B3 .. .. ..
240050	B4 .. .. .. B5 .. .. .. B6 .. .. .. B7 .. .. ..
240060	B8 .. .. .. B9 .. .. .. BA .. .. .. BB .. .. ..
240070	BC .. .. .. BD .. .. .. BE .. .. .. BF .. .. ..
280000	C0 .. .. .. C1 .. .. .. C2 .. .. .. C3 .. .. ..
280010	C4 .. .. .. C5 .. .. .. C6 .. .. .. C7 .. .. ..
280020	C8 .. .. .. C9 .. .. .. CA .. .. .. CB .. .. ..
280030	CC .. .. .. CD .. .. .. CE .. .. .. CF .. .. ..
280040	D0 .. .. .. D1 .. .. .. D2 .. .. .. D3 .. .. ..
280050	D4 .. .. .. D5 .. .. .. D6 .. .. .. D7 .. .. ..
280060	D8 .. .. .. D9 .. .. .. DA .. .. .. DB .. .. ..
280070	DC .. .. .. DD .. .. .. DE .. .. .. DF .. .. ..
2C0000	E0 .. .. .. E1 .. .. .. E2 .. .. .. E3 .. .. ..
2C0010	E4 .. .. .. E5 .. .. .. E6 .. .. .. E7 .. .. ..
2C0020	E8 .. .. .. E9 .. .. .. EA .. .. .. EB .. .. ..
2C0030	EC .. .. .. ED .. .. .. EE .. .. .. EF .. .. ..
2C0040	F0 .. .. .. F1 .. .. .. F2 .. .. .. F3 .. .. ..
2C0050	F4 .. .. .. F5 .. .. .. F6 .. .. .. F7 .. .. ..
2C0060	F8 .. .. .. F9 .. .. .. FA .. .. .. FB .. .. ..
2C0070	FC .. .. .. FD .. .. .. FE .. .. .. FF .. .. ..

—This Page Intentionally Left Blank—

Electrical Specifications

Table 7-1. DC Electrical Characteristics

Specified at: V_{DD} = 5 V ±5%, V_{SS} = 0 V, 0°C to +70°C

Symbol	Parameter	Conditions	Min	Typ	Max	Units
V_{IL}	Low Level Input Voltage					
	TTL				0.8	V
	CMOS				1.5	V
	Schmitt				1.0	V
V_{IH}	High Level Input Voltage					
	TTL		2.2			V
	CMOS		3.5			V
	Schmitt		4.0			V
V_H	Schmitt Hysteresis			1.4		V
I_{IL}	Low Level Input Current	$V_{IN} = V_{SS}$	-10		+10	µA
	Input buffer with pull-up	V_{DD} = 5.25 V	–200		–10	
I_{IH}	High Level Input Current	$V_{IN} = V_{DD}$	–10		10	µA
	Input buffer with pull-down	V_{DD} = 5.25 V	10		200	
V_{OL}	Low Level Output Voltage	$V_{IN} = V_{DD}/V_{SS}$ V_{DD} = 4.75 V I_{OL} = 4mA			V_{SS} + 0.05 0.4	V V
V_{OH}	High Level Output Voltage	$V_{IN} = V_{DD}/V_{SS}$ V_{DD} = 4.75 V I_{OL} = -4mA	V_{DD} – 0.05 2.4			V V
I_{OZ}	High Impedance Leakage Current	$V_O = V_{SS}$	–10		10	µA
	Output buffer with pull-up	V_{DD} = 5.25 V	–200		–10	
I_{DD}	Quiescent Supply Current			1		mA
C_{IN}	Input Capacitance			2		pF
C_{OUT}	Output Capacitance			4		pF
P_D	Total Power Dissipation				0.8	W

Table 7-2. Absolute Maximum Ratings

DC Supply Voltage (V_{DD})	- 0.3 V to + 7 V
Input Voltage	- 0.3 V to VDD + 0.3 V
DC Input Current	±10 mA
Storage Temperature	- 40°C to + 125°C

Caution: Stresses above those listed under *Absolute Maximum Ratings* may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions may affect device reliability.

Table 7-3. Recommended Operating Conditions

DC Supply Voltage	+ 4.75 V to + 5.25 V
Operating Temperature Range	0°C to + 70°C

—This Page Intentionally Left Blank—

Timing Specifications

Table 8-1. External Oscillator Timing Requirements

Specified at V_{DD} = 5 V ±5%, V_{SS} = 0 V, 0°C to +70°C
The numbers below correspond to the numbers in the figures throughout this section.

Numbers	Name	Description	Min	Typ	Max	Unit
	f_{OSC}	Oscillator frequency		40		MHz
2	t_{CYC}	Clock period	25			ns
5	t_{WHIGH}	Clock width high	10			ns
6	t_{RISE}	Clock rise time			3	ns
7	t_{FALL}	Clock fall time			3	ns
8	t_{WLOW}	Clock width low	10			ns

Table 8-2. Clock Output Timing

Specified at V_{DD} = 5 V ±5%, V_{SS} = 0 V, 0°C to +70°C, f_{osc} (50% duty cycle).
The numbers below correspond to the numbers in the figures throughout this section.

Numbers	Name	Description	Min	Typ	Max	Unit
2	t_{CYC}	CLKOUT1 period		25		ns
5	t_{WHIGH}	CLKOUT1 width high	13.5		15.5	ns
8	t_{WLOW}	CLKOUT1 width low	9.5		11.5	ns
2	t_{CYC}	CLKOUT2 period		50		ns
5	t_{WHIGH}	CLKOUT2 width high	25.5		28	ns
8	t_{WLOW}	CLKOUT2 width low	22		24.5	ns
2	t_{CYC}	CLKOUT3 period		75		ns
5	t_{WHIGH}	CLKOUT3 width high	38		40	ns
8	t_{WLOW}	CLKOUT3 width low	35		37	ns
2	t_{CYC}	CLKOUT4 period		100		ns
5	t_{WHIGH}	CLKOUT4 width high	50.5		52.5	ns
8	t_{WLOW}	CLKOUT4 width low	47.5		49.5	ns
2	t_{CYC}	CLKOUT8 period		200		ns
5	t_{WHIGH}	CLKOUT8 width high	100.5		102	ns
8	t_{WLOW}	CLKOUT8 width low	98		99.5	ns

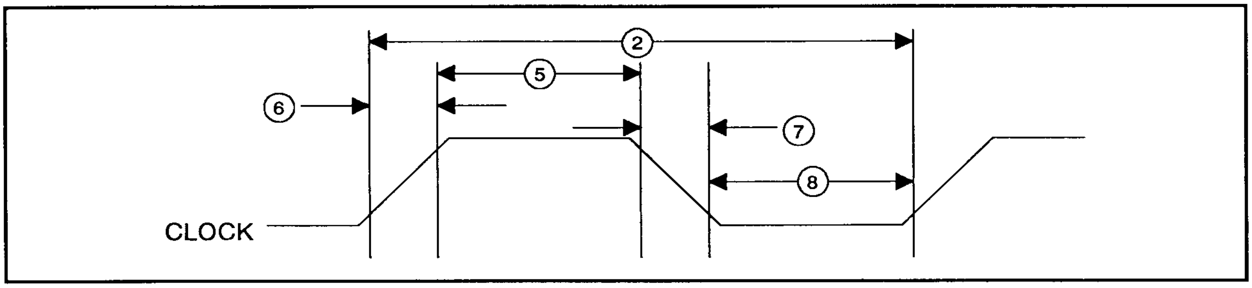


Figure 8-1. Oscillator and Clock Timing

Refer to *Electrical Specifications* for voltage and current information.

Table 8-3. Reset Timing

Specified at: V_{DD} = 5 V ±5%, V_{SS} = 0 V, 0°C to +70°C

The numbers below correspond to the numbers in the figures throughout this section.

Numbers	Name	Description	Min	Max	Unit
1	t_{PRST}	Power on RESET	3	-	ms
4	t_{RESET}	$\overline{\text{RESET}}$ width	10	-	t_{CYC}

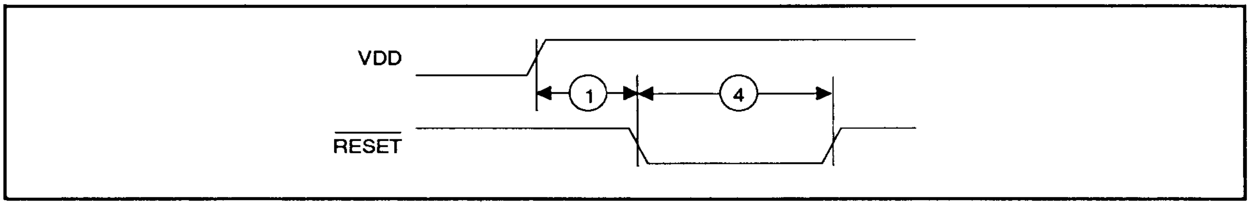


Figure 8-2. Power-on Reset Timing

MPU Interface Timing

The 9802A interfaces to a variety of microprocessors through a bidirectional data bus (D0-D7), six address pins (RS0-RS3), two chip select pins ($\overline{\text{CS1}}$ and CS2), a read pin ($\overline{\text{MPURD}}$), and a write pin ($\overline{\text{MPUWR}}$). To simplify the description of the MPU interface timing, the $\overline{\text{CS}}$ signal shown in Figures 8-3 – 8-7 below represents a composite signal which is active (low) when both chip select signals are active.

Read Cycle Timing

Processors with one read/write signal will use $\overline{\text{MPUWR}}$ to select a read cycle. $\overline{\text{MPURD}}$ does not need to be tied to ground, but it is recommended. During the read cycle, $\overline{\text{MPUWR}}$ must remain high during the entire time the chip is selected. (See Figure 8-3.) Processors with separate read and write signals will use $\overline{\text{MPURD}}$ to select a read cycle. $\overline{\text{MPURD}}$ and $\overline{\text{CS}}$ must remain low during the entire cycle, and $\overline{\text{MPUWR}}$ must remain high. (See Figure 8-5.)

During read cycles, the 9802A drives the data bus, D0-D7, while the 9802A is selected. The register select pins, RS0-RS5, must be stable before the start of the cycle and during the entire cycle.

Write Cycle Timing

During a write cycle, the register select pins, RS0-RS5, must be stable during the entire cycle. For processors with one read/write signal, a write cycle begins when $\overline{\text{MPUWR}}$ and $\overline{\text{CS}}$ are low. $\overline{\text{MPURD}}$ does not need to be tied to ground, but it is recommended. Data is latched when $\overline{\text{CS}}$ is deasserted. (See Figure 8-4.) For processors with separate read and write signals, a write cycle begins when $\overline{\text{MPUWR}}$ and $\overline{\text{CS}}$ are low. $\overline{\text{MPURD}}$ must remain high during the entire cycle. Data is latched when either $\overline{\text{CS}}$ or $\overline{\text{MPUWR}}$ is deasserted. (See Figure 8-6.)

DMA Interrupt Timing

An interrupt request can be generated on completion of a DMA transfer. The interrupt time is relative to the number of bytes remaining to be transferred and can be programmed using the IRQ TIME SELECT field in the DMA Command register. The timing of the interrupt request signal relative to DACK is described in Table 8-4 and Figures 8-4 and 8-6 below.

Table 8-4. MPU Interface Timing

Specified at: V_{DD} = 5 V ±5%, V_{SS} = 0 V, 0°C to +70°C
The numbers below correspond to the numbers in the figures throughout this section.

Number	Description	Min	Max	Unit
9	RS setup time before $\overline{CS}$ active	note**		ns
10	$\overline{MPUWR}$ setup time before $\overline{CS}$ active	0		ns
11	$\overline{MPUWR}$ hold time after $\overline{CS}$ inactive	0		ns
12	$\overline{CS}$ active to $\overline{WAIT}$ active*	5	25	
13	Disable time from $\overline{CS}$ inactive to high-Z data bus		20	ns
14	MPUDREQ active to high-Z $\overline{WAIT}$	1*	note*	ns
15	$\overline{MPUDACK}$ active to data valid		20	ns
16	Access time from $\overline{CS}$ active		note***	ns
17	$\overline{MPUDACK}$ inactive to high-Z data bus	2	15	ns
18	Enable time from $\overline{CS}$ active to driven data bus		25	ns
19	RS hold time after $\overline{MPUWR}$ high or $\overline{CS}$ inactive	0		ns
20	Write pulse width ($\overline{MPUWR}$ low and $\overline{CS}$ active)	t_{CYC} + 15		ns
21	Write Recovery Time	$5t_{CYC}$		ns
22	Data hold time after $\overline{MPUWR}$ high or $\overline{CS}$ inactive	0		ns
23	$\overline{MPUDACK}$ active to MPUDREQ inactive	3	20	ns
24	Data hold time after $\overline{MPUDACK}$ inactive	5		ns
25	Data setup time before $\overline{MPUDACK}$ inactive	5		ns
26	Data setup time before $\overline{MPUWR}$ high or $\overline{CS}$ inactive	10		ns
27	DACK inactive to $\overline{IRQ}$ active		t_{CYC}	ns
28	$\overline{CS}$ setup time before $\overline{MPURD}$ or $\overline{MPUWR}$ active	0		ns
29	$\overline{CS}$ hold time after $\overline{MPURD}$ or $\overline{MPUWR}$ inactive	0		ns

* $\overline{WAIT}$ will only be asserted during MPU reads and writes of the MPU Buffer Data Register and during MPU reads of RAM-based registers (all registers with addresses, in hex, 12–17, 1A–1F, 22–27, 2A–2F, 32–37, 3A–3F). For reads and writes of the MPU Buffer Data Register, the minimum time from MPUDREQ active to high-Z $\overline{WAIT}$ is 1 and the maximum time is 10. For MPU reads of RAM-based registers, WAIT will go to the high impedance state 5 ns (max) after the 9802A writes data.

** RS must be valid on or prior to the last assertion of $\overline{CS}$, $\overline{MPUWR}$ or $\overline{MPURD}$.

*** For RAM-based registers (all registers with addresses, in hex, 12–17, 1A–1F, 22–27, 2A–2F, 32–37, 3A–3F), maximum time is $5t_{CYC}$ + 15. For non-RAM based registers, maximum time is 40.

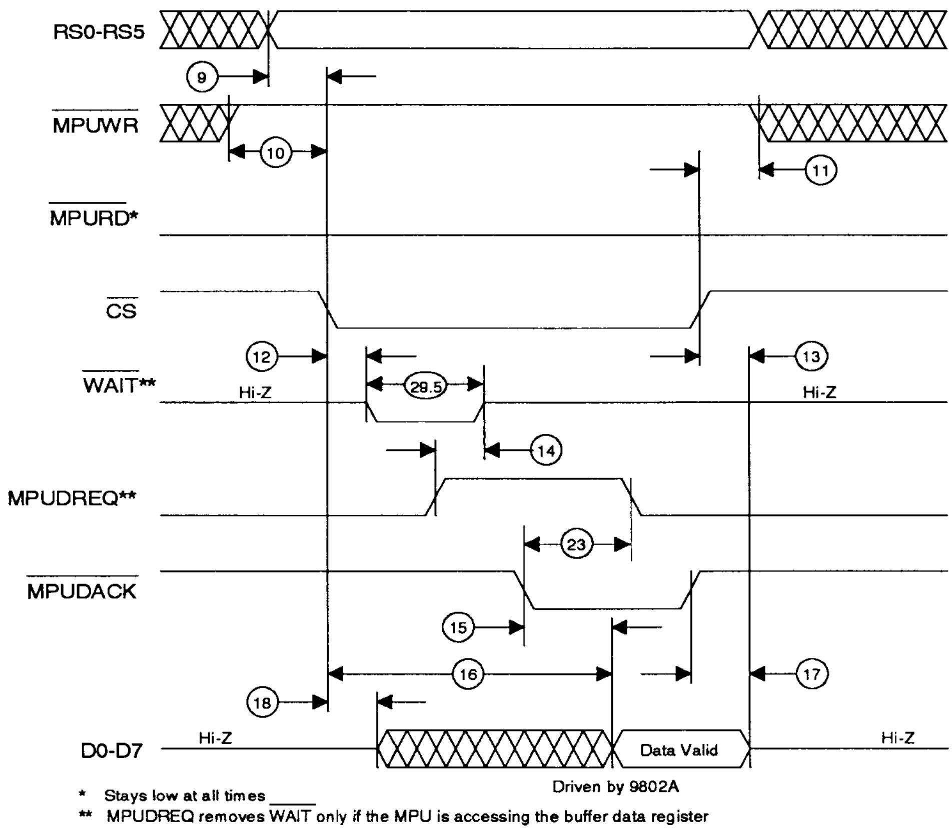


Figure 8-3. MPU Read Cycle Timing - Processors with One Read/Write Signal

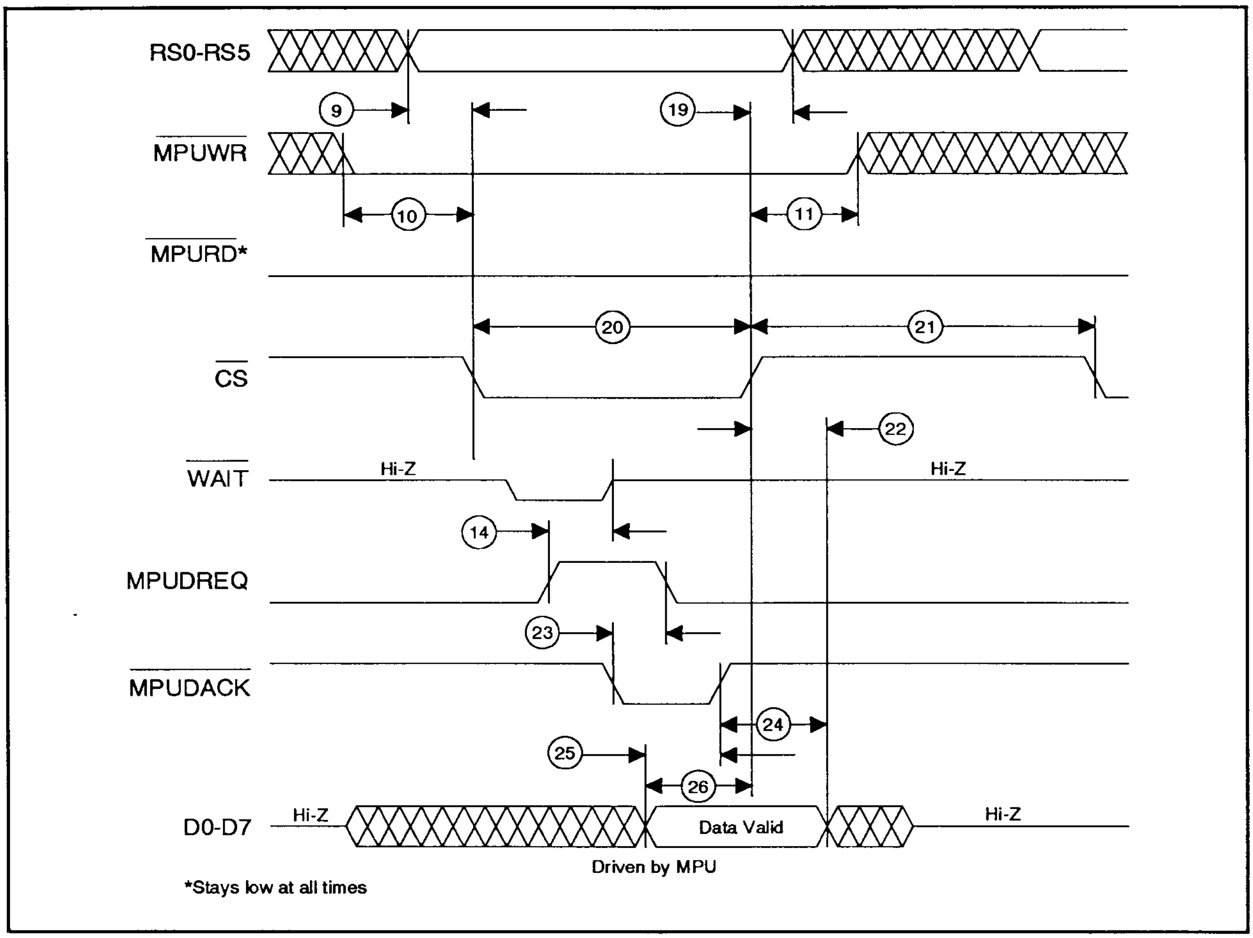


Figure 8-4. MPU Write Cycle Timing – Processors with One Read/Write Signal

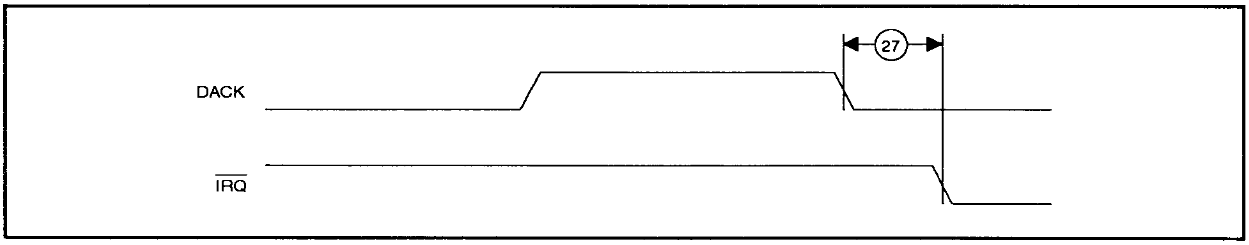


Figure 8-5. DMA Interrupt Timing - Processors with One Read/Write Signal

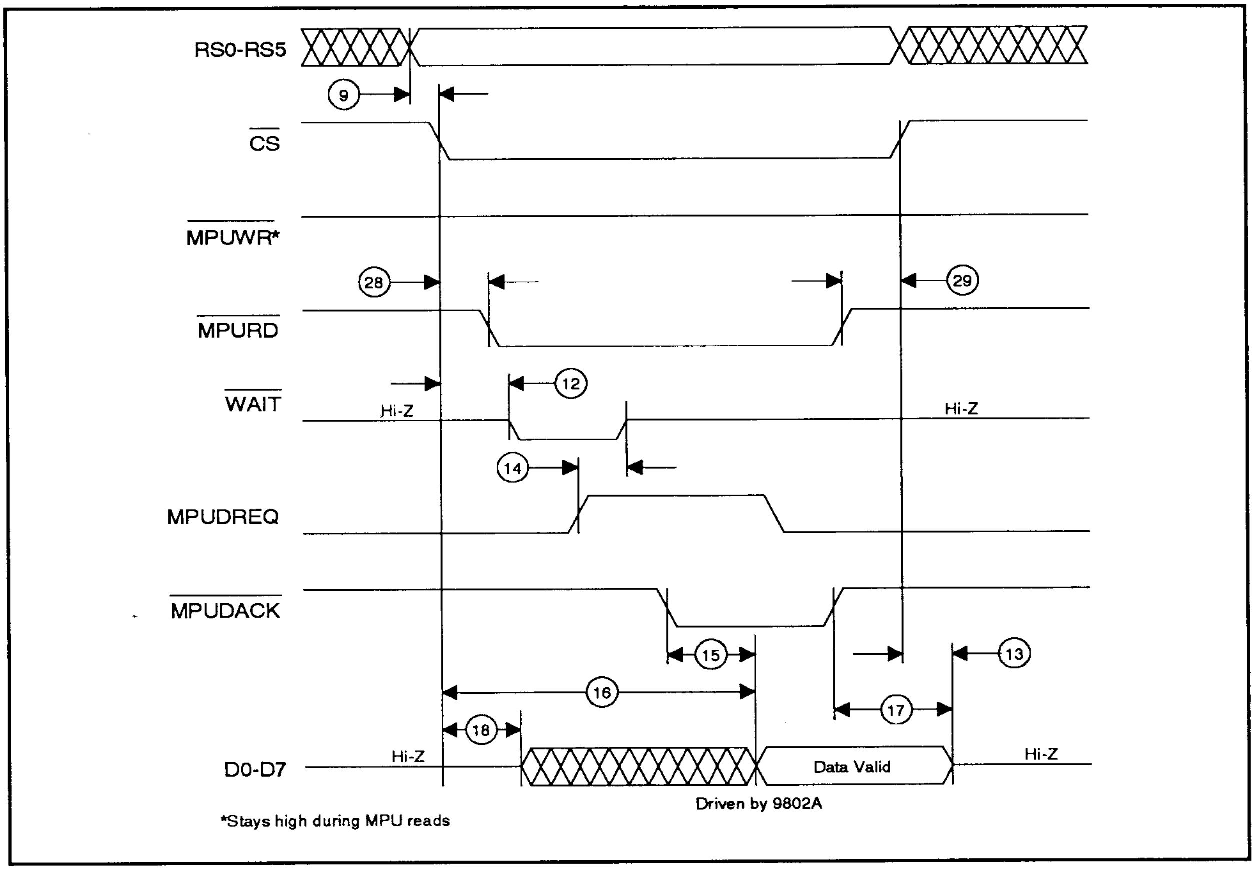


Figure 8-6. MPU Read Cycle Timing – Processors with Separate Read and Write Signals

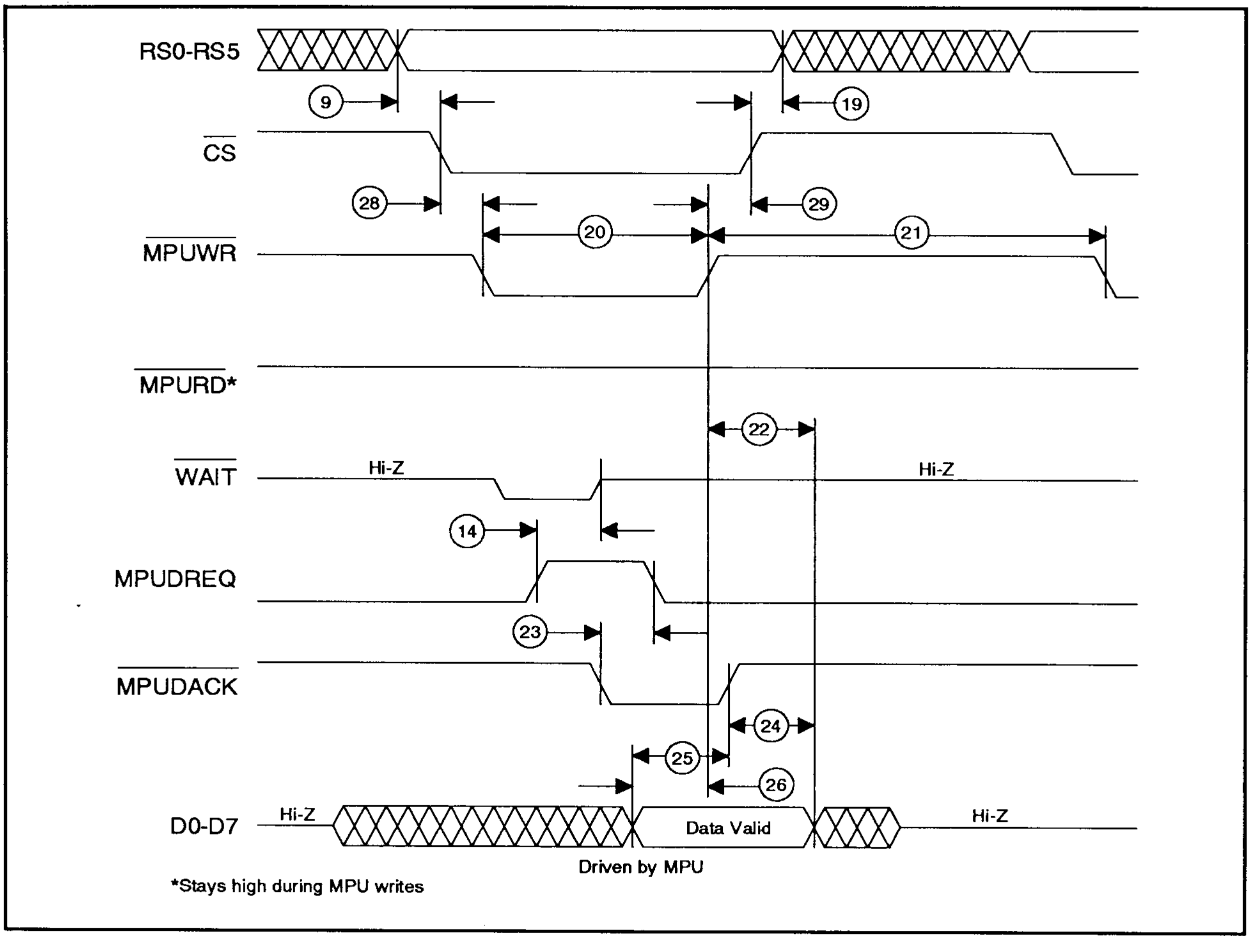


Figure 8-7. MPU Write Cycle Timing – Processors with Separate Read and Write Signals

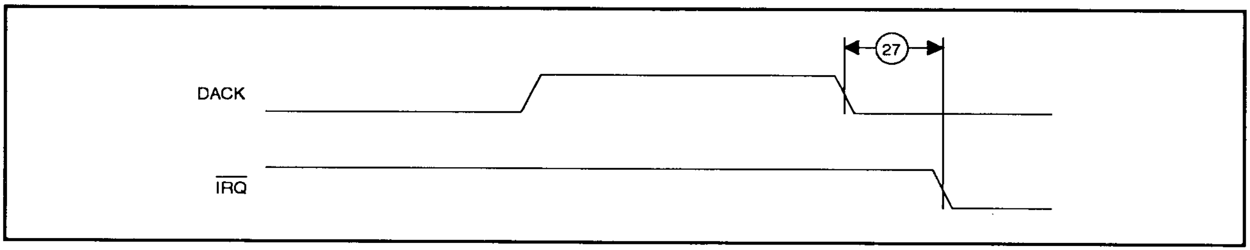


Figure 8-8. DMA Interrupt Timing – Processors with Separate Read and Write Signals

DRAM Interface Timing

The DRAM interface consists of the address pins (A0/1-A22/23), the buffer data bus pins (BD0-BD7, BDP), the write enable pin ($\overline{\text{WE}}$), the column address strobe pin ($\overline{\text{CAS}}$), and the two row address strobe pins ($\overline{\text{RAS1}}$, $\overline{\text{RAS2}}$). Table 8-5 and Figure 8-9 below describe the timing characteristics of the DRAM interface.

Table 8-5. DRAM Interface Timing

Specified at: V_{DD} = 5 V ±5%, V_{SS} = 0 V, 0°C to +70°C
The numbers below correspond to the numbers in the figures throughout this section.

Number	Description	Min	Max	Unit
30	Row address hold after $\overline{\text{RAS}}$ active	t_{CYC} - 10		ns
31	Column address hold after $\overline{\text{CAS}}$ active	$(4t_{CYC})^*$ - 5		ns
32	Row address setup time to $\overline{\text{RAS}}$ active	$(t_{CYC})^*$ - 10		ns
33	Column address setup time before $\overline{\text{CAS}}$ active	t_{CYC} - 20		ns
34	$\overline{\text{RAS}}$ width	$(4t_{CYC})^*$ - 5	$(4t_{CYC})^*$ + 5	ns
35	$\overline{\text{RAS}}$ precharge time	$(3t_{CYC})^*$ - 5		ns
36	$\overline{\text{RAS}}$ hold time	$(2t_{CYC})^*$ - 5		ns
37	$\overline{\text{RAS}}$ inactive to $\overline{\text{CAS}}$ inactive	t_{CYC} - 5	t_{CYC} + 10	ns
38	Read cycle time	$7t_{CYC}$ or $9t_{CYC}$**		ns
39	$\overline{\text{CAS}}$ width	$(3t_{CYC})^*$ - 5	$(3t_{CYC})^*$ + 5	ns
40	Pulse duration, $\overline{\text{CAS}}$ inactive	$(4t_{CYC})$		ns
41	Read command setup before $\overline{\text{CAS}}$ active	t_{CYC} - 10		ns
42	Read command hold after $\overline{\text{CAS}}$ inactive	$(5t_{CYC})^*$ - 5		ns
43	$\overline{\text{CAS}}$ access time	$(2.5t_{CYC})^*$ - 15		ns
44	Data setup time before $\overline{\text{CAS}}$ inactive	$0.5t_{CYC}$ + 10		ns
45	$\overline{\text{RAS}}$ access time	$(4.5t_{CYC})^*$ - 25		ns
46	Data hold time after $\overline{\text{CAS}}$ inactive	0		ns
47	Write cycle time	$7t_{CYC}$ or $9t_{CYC}$**		ns
48	$\overline{\text{RAS}}$ active to $\overline{\text{WE}}$ active		t_{CYC} + 10	ns
49	$\overline{\text{WE}}$ setup before $\overline{\text{CAS}}$ active	t_{CYC} - 10		ns
50	$\overline{\text{WE}}$ width	$(4t_{CYC})^*$ - 10		ns
51	Data setup time before $\overline{\text{CAS}}$ active	t_{CYC} - 10		ns
52	Data hold time after $\overline{\text{CAS}}$ inactive	t_{CYC} - 10		ns
53	Data valid delay from $\overline{\text{RAS}}$ active		t_{CYC} + 15	ns

*Note: If BUFFER RAM CYCLE TIME SELECT bit in Configuration Register is set, add one to coefficient of t_{CYC} in minimum and maximum specifications.

**Note: If BUFFER RAM CYCLE TIME SELECT bit in Configuration Register register is set, minimum time is $9t_{CYC}$. Otherwise, minimum time is $7t_{CYC}$.

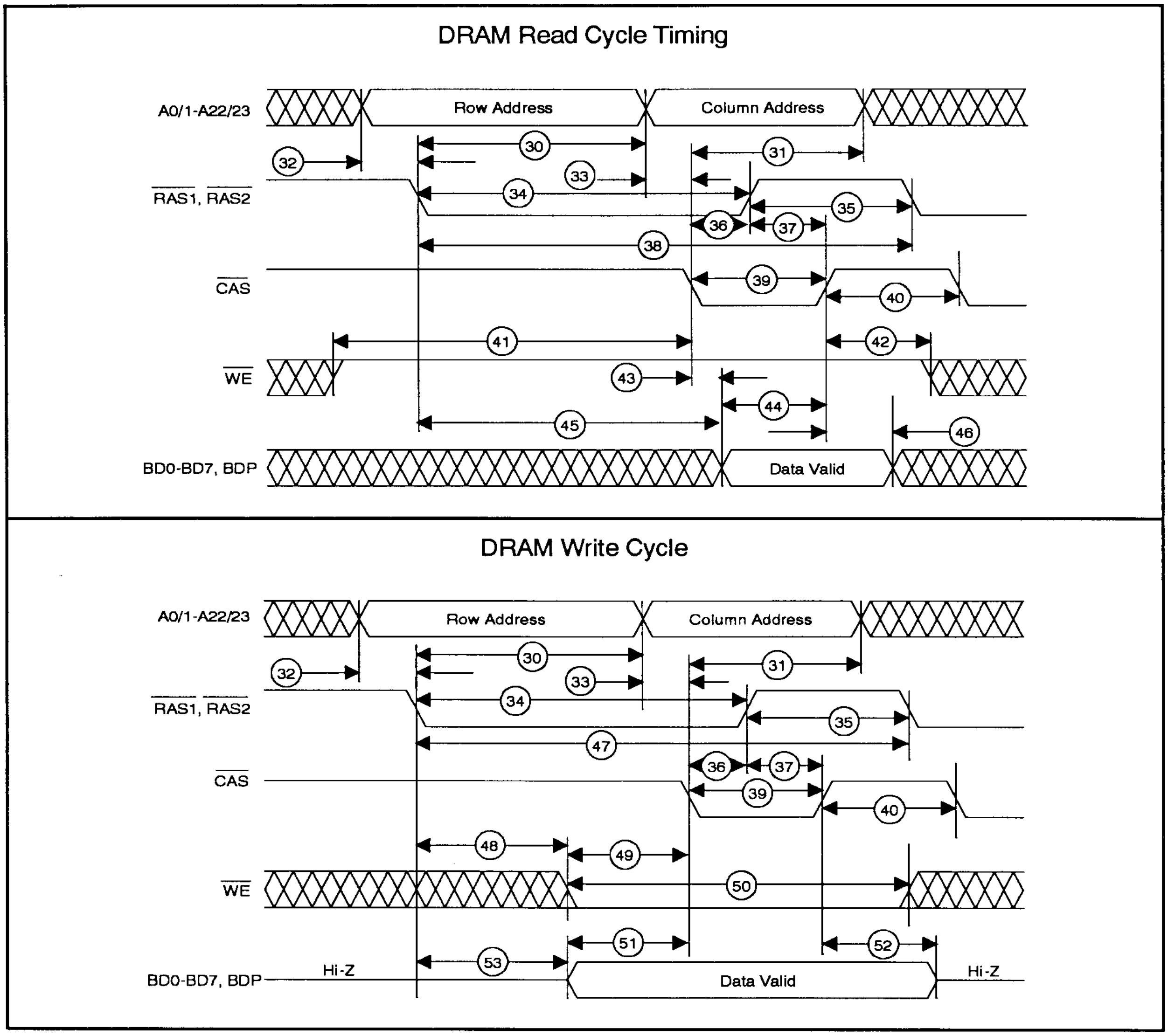


Figure 8-9. DRAM Interface Timing

Table 8-6. DRAM Refresh Timing

Specified at: V_{DD} = 5 V ±5%, V_{SS} = 0 V, 0°C to +70°C

The numbers below correspond to the numbers in the figures throughout this section.

Number	Description	Min	Max	Unit
54	Address setup time before $\overline{\text{RAS}}$ active	t_{CYC}		ns
55	Address hold time after $\overline{\text{RAS}}$ active	$3t_{CYC}$ - 5		ns
56	Refresh cycle time	$8t_{CYC}$ or $10t_{CYC}$**		ns
57	$\overline{\text{RAS}}$ width	$(4t_{CYC})$* - 5	$(4t_{CYC})$* + 5	ns
58	$\overline{\text{RAS}}$ precharge time	$(3t_{CYC})$* - 5		ns
59	$\overline{\text{CAS}}$ to $\overline{\text{RAS}}$ precharge time	$(3t_{CYC})$* - 5		ns
60	$\overline{\text{CAS}}$ hold time after $\overline{\text{RAS}}$ inactive	$(5t_{CYC})$* - 5		ns
61	$\overline{\text{WE}}$ setup time before $\overline{\text{RAS}}$ active	$(3t_{CYC})$* - 10		ns
62	$\overline{\text{WE}}$ hold time after $\overline{\text{RAS}}$ inactive	$(4t_{CYC})$* - 10		ns

*Note: If the BUFFER RAM CYCLE TIME SELECT bit in the Configuration Register is set, add one to the coefficient of t_{CYC} in the minimum and maximum specifications.

**Note: If the BUFFER RAM CYCLE TIME SELECT bit in the Configuration Register register is set, the minimum time is $10t_{CYC}$. Otherwise, the minimum time is $8t_{CYC}$.

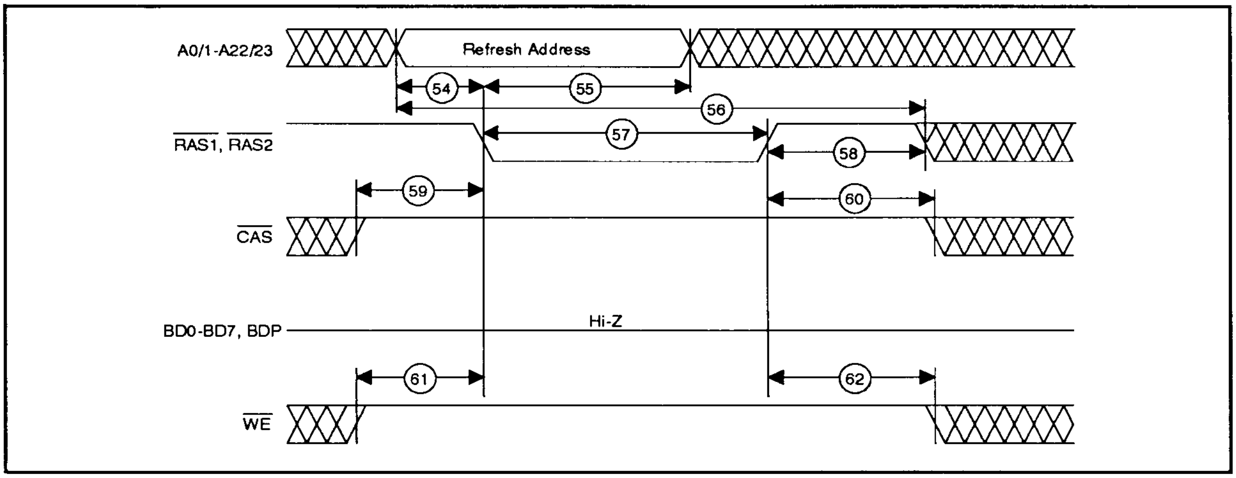


Figure 8-10. DRAM Refresh Cycle Timing

DMA Interface Timing

The LENGTH SELECT field in the DMA Handshake Configuration Register allows the DMA acknowledge to operate in four-cycle handshake mode or in demand mode. In both a four-cycle and a demand mode handshake system, the 9802A receives a DMA request and acknowledges it. In four-cycle handshake mode, DREQ must go inactive before DACK can go inactive. In demand mode, DACK remains active for a programmed number of system clocks without preventing DACK from going inactive. Table 8-7 and Figure 8-11 describe the timing for the two methods of DMA handshaking. For all DMA interface timing, DREQ and DACK polarity are shown active high.

Note that in Table 8-7, "n" is a factor determined from Table 3-7, DACK3 Length Select Interpretation. Available values for n are 3, 5 and 7 system clocks. This length is selected by programming the LENGTH SELECT bits in the DMA Handshake Configuration Register.

Table 8-7. DMA Interface Timing

Specified at: V_{DD} = 5 V ±5%, V_{SS} = 0 V, 0°C to +70°C
The numbers below correspond to the numbers in the figures throughout this section.

Number	Description	Min	Max	Unit
63	DREQ hold after DACK active	0		ns
64	DREQ active to DACK active	$2t_{CYC} + 15$		ns
65	DREQ inactive to DACK inactive	$2t_{CYC} + 15$	$3t_{CYC} + 10$	ns
66	DACK inactive pulse width	$2t_{CYC} + 15$		ns
67	DACK cycle time	$7t_{CYC}$		ns
68	DACK active pulse width	$nt_{CYC} - 15$	$nt_{CYC} + 15$	ns
69	DACK inactive to DREQ active (to avoid delaying the next DACK)		$t_{CYC} - 25$	ns
70	DREQ inactive to end of transfer	0		ns
71	Data setup time before DACK inactive	$t_{CYC} + 15$		ns
72	Data hold time after DACK inactive	0		ns
73	DACK active to $\overline{\text{PRD}}$ active	0		ns
74	$\overline{\text{PRD}}$ inactive to DACK inactive	0		ns
75	DACK inactive to $\overline{\text{TCI}}$ inactive	$t_{CYC} - 5$	$t_{CYC} + 10$	ns
76	DACK active to $\overline{\text{TCI}}$ active	$t_{CYC} - 5$	$t_{CYC} + 10$	ns
77	$\overline{\text{TOE}}$ active after DACK active	36	43	ns
78	$\overline{\text{TOE}}$ active pulse width	$8t_{CYC}$		ns
79	Enable time from DACK active to data valid		$t_{CYC} + 15$	ns
80	Data hold time after DACK inactive	t_{CYC}	$t_{CYC} + 15$	ns
81	DACK active to $\overline{\text{PWR}}$ active	0		ns
82	$\overline{\text{PWR}}$ inactive toDACK inactive	0		ns

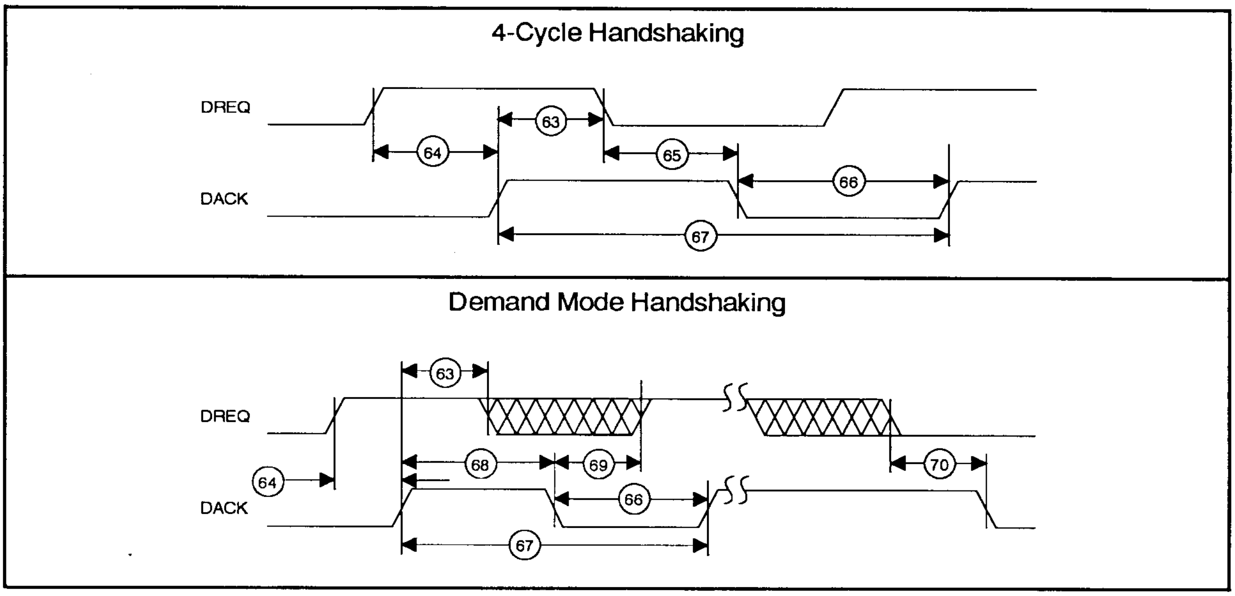
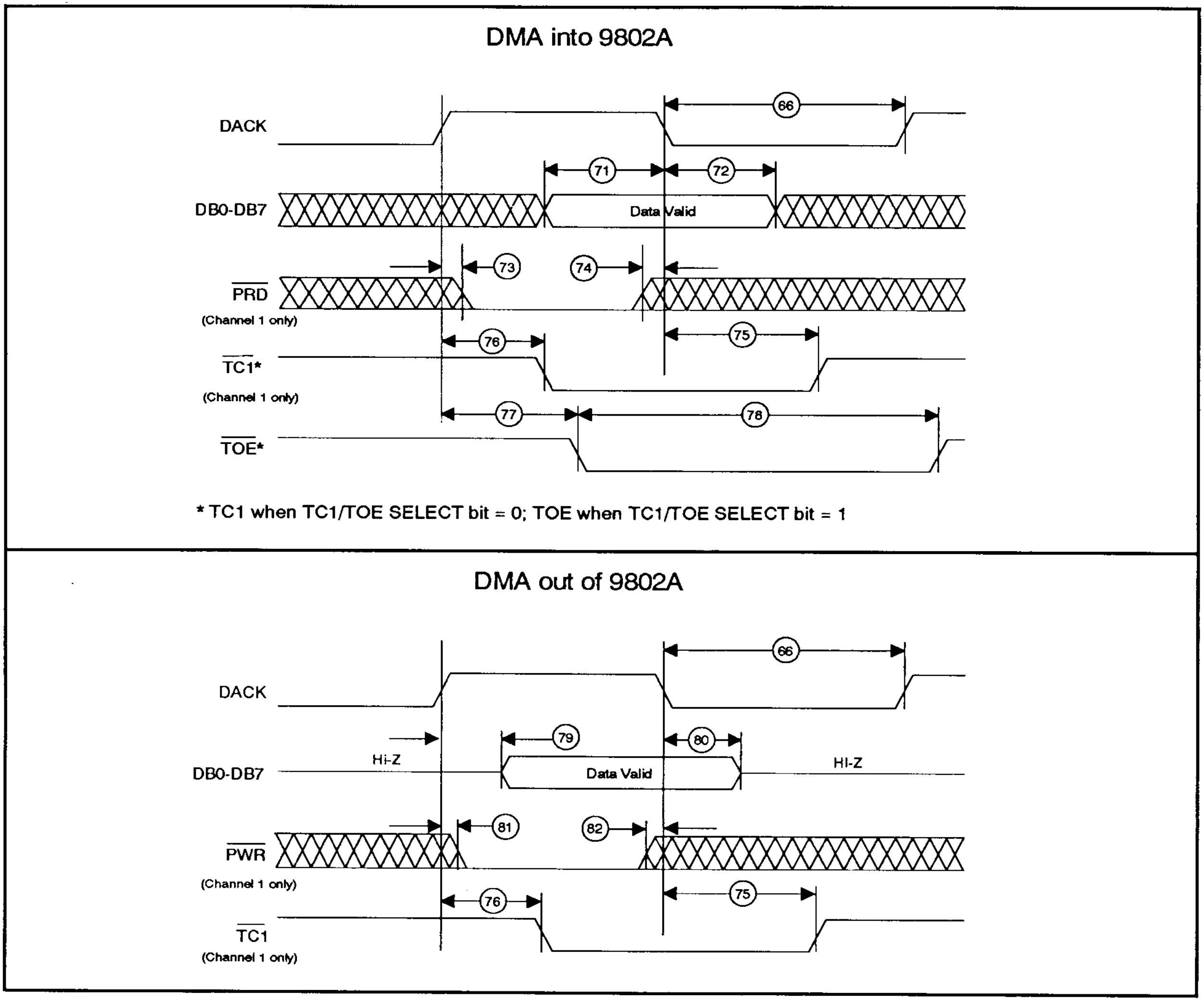


Figure 8-11. DMA Interface Timing



DMA into 9802A

* TC1 when TC1/TOE SELECT bit = 0; TOE when TC1/TOE SELECT bit = 1

DMA out of 9802A

Figure 8-12. DMA Interface Timing (cont.)

Peripheral Access Timing

The 9802A allows access to single-bus peripherals over DMA channel 1. Table 8-8 and Figure 8-13 describe the timing for a peripheral access operation. In the diagram below, the signal labelled "$\overline{\text{PCS}}$" represents a composite signal which is active when $\overline{\text{PCSIN}}$ and CS2 are active. $\overline{\text{WAIT}}$ is always asserted at the beginning of a peripheral access. If DMA Channel 1 is in the middle of a transfer, $\overline{\text{WAIT}}$ will remain active until access to the peripheral is granted (until DACK1 goes inactive).

Read Cycle Timing

During a read cycle, $\overline{\text{MPUWR}}$ must remain high during the entire time of the peripheral access (while $\overline{\text{PCSIN}}$ and CS2 are asserted). The 9802A drives the data bus, D0–D7, with the data presented by the peripheral on DB10-DB17.

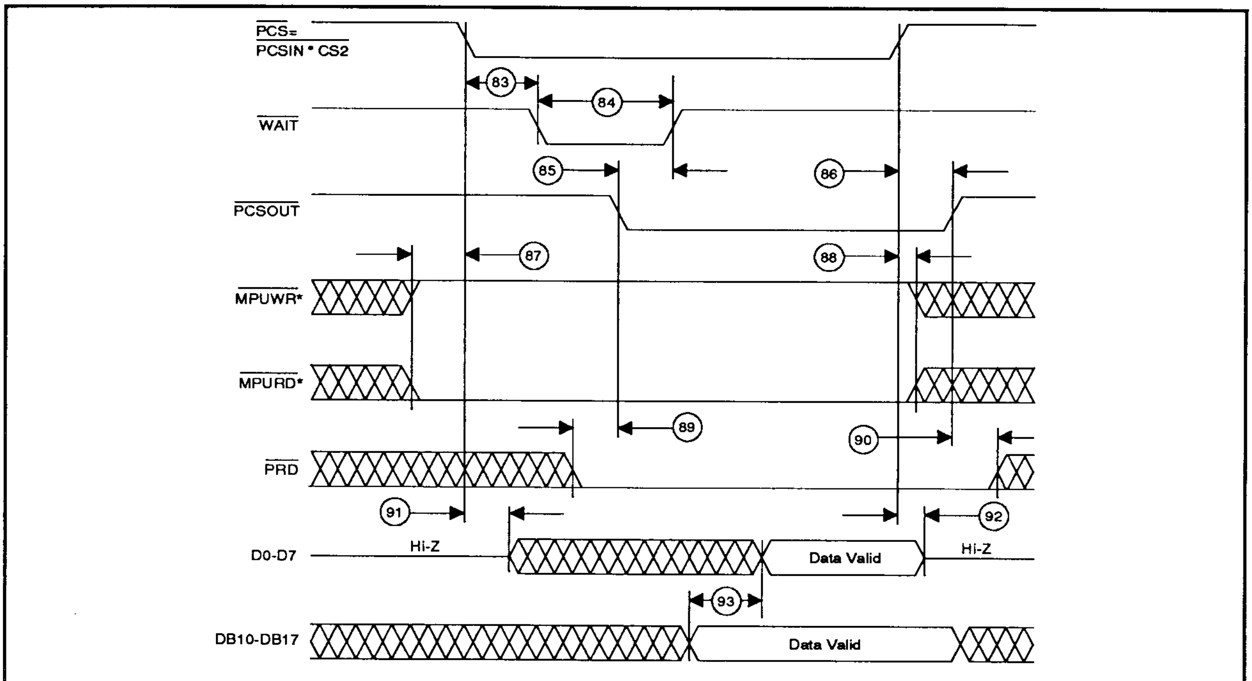
Write Cycle Timing

A write cycle begins when $\overline{\text{MPUWR}}$ is low and the a peripheral access is requested. The MPU data is latched by the 9802A at the end of the MPU cycle (when $\overline{\text{PCSIN}}$ and CS2 go inactive). The 9802A drives the DMA data bus until one clock after $\overline{\text{PCSOUT}}$ is deasserted.

Table 8-8. Peripheral Access Timing

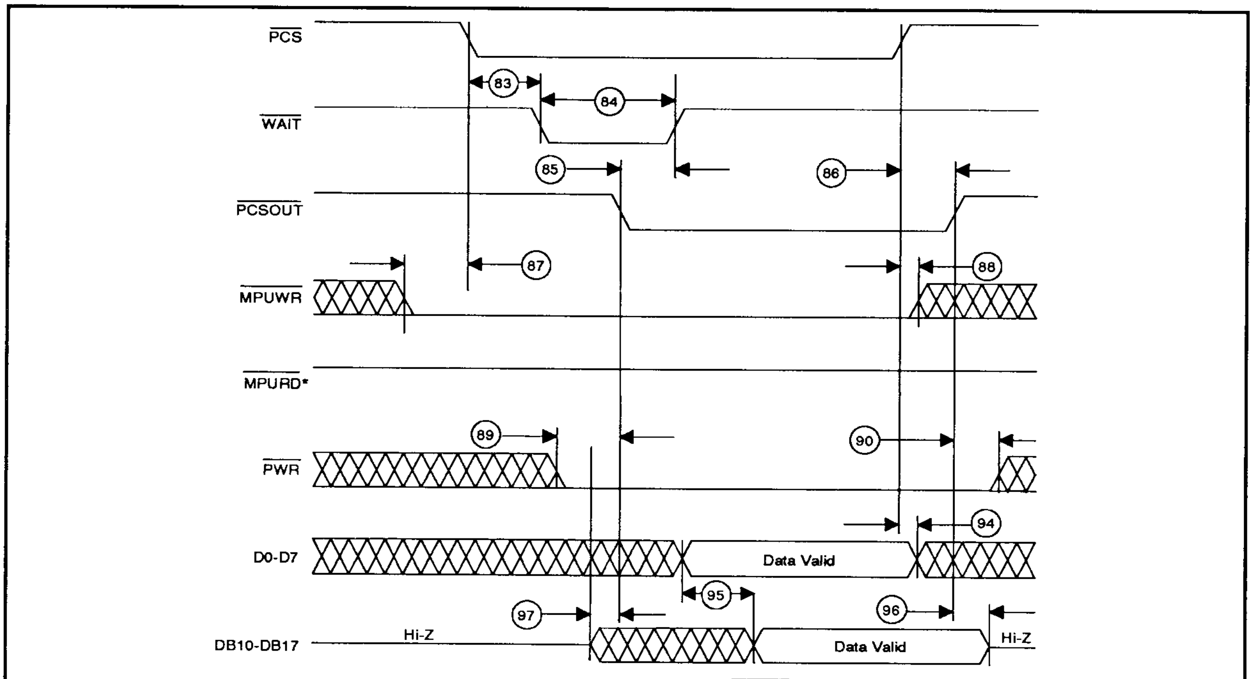
Specified at: V_{DD} = 5 V ±5%, V_{SS} = 0 V, 0°C to +70°C
The numbers below correspond to the numbers in the figures throughout this section.

Number	Description	Min	Max	Unit
83	$\overline{\text{PCS}}$ active to $\overline{\text{WAIT}}$ active	5	20	ns
84	$\overline{\text{WAIT}}$ active pulse width	$4t_{CYC}$		ns
85	$\overline{\text{PCSOUT}}$ active to $\overline{\text{WAIT}}$ inactive	t_{CYC}	$t_{CYC} + 5$	ns
86	$\overline{\text{PCS}}$ inactive to $\overline{\text{PCSOUT}}$ inactive	$3t_{CYC} + 5$	$4t_{CYC} + 5$	ns
87	$\overline{\text{MPUWR}}$ or $\overline{\text{MPURD}}$ set-up time before $\overline{\text{PCS}}$ active	0		ns
88	$\overline{\text{MPUWR}}$ or $\overline{\text{MPURD}}$ hold time after $\overline{\text{PCS}}$ inactive	0		ns
89	$\overline{\text{PRD}}$ or $\overline{\text{PWR}}$ active to $\overline{\text{PCSOUT}}$ active	20	25	ns
90	$\overline{\text{PCSOUT}}$ inactive to $\overline{\text{PRD}}$ or $\overline{\text{PWR}}$ inactive	0		ns
91	MPU data bus enable time from $\overline{\text{PCS}}$ active		25	ns
92	MPU data bus disable time from $\overline{\text{PCS}}$ inactive		20	ns
93	DMA data bus valid to MPU data bus valid		20	ns
94	MPU data bus hold time after $\overline{\text{PCS}}$ inactive	5		ns
95	MPU data bus valid to DMA data bus valid		30	ns
96	DMA data bus disabled from $\overline{\text{PCSOUT}}$ inactive	t_{CYC}	$t_{CYC} + 5$	ns
97	DMA data bus enable time from $\overline{\text{PCSOUT}}$ active	20	30	ns



*Note: Processors with one read/write signal use MPUWR to select a read cycle. MPURD stays low at all times Processors with separate read and write signals use MPURD to select a read cycle. MPUWR stays high during read cycles.

Figure 8-13. Peripheral Access Read Cycle Timing



*Note: When the 9820 is used with a processor with one read/write signal, MPURD stays low at all times. When the 9820 is used with a processor with separate read and write signals, MPURD stays high during write cycles.

Figure 8-14. Peripheral Access Write Cycle Timing

Package Information

100-Pin Plastic Quad Flat Pack (PQFP)

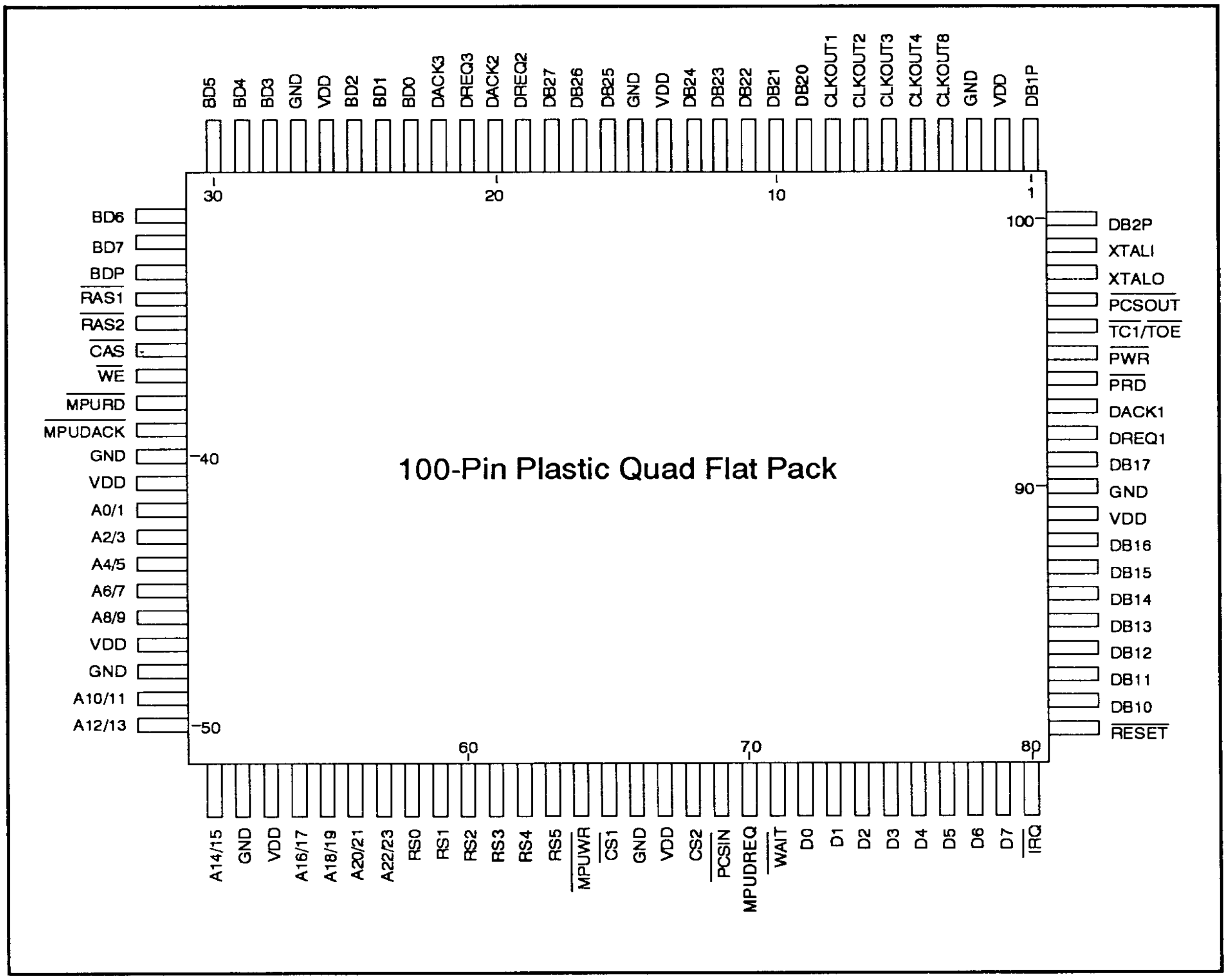


Figure 9-1. 100-Pin Plastic Quad Flat Pack Pinout

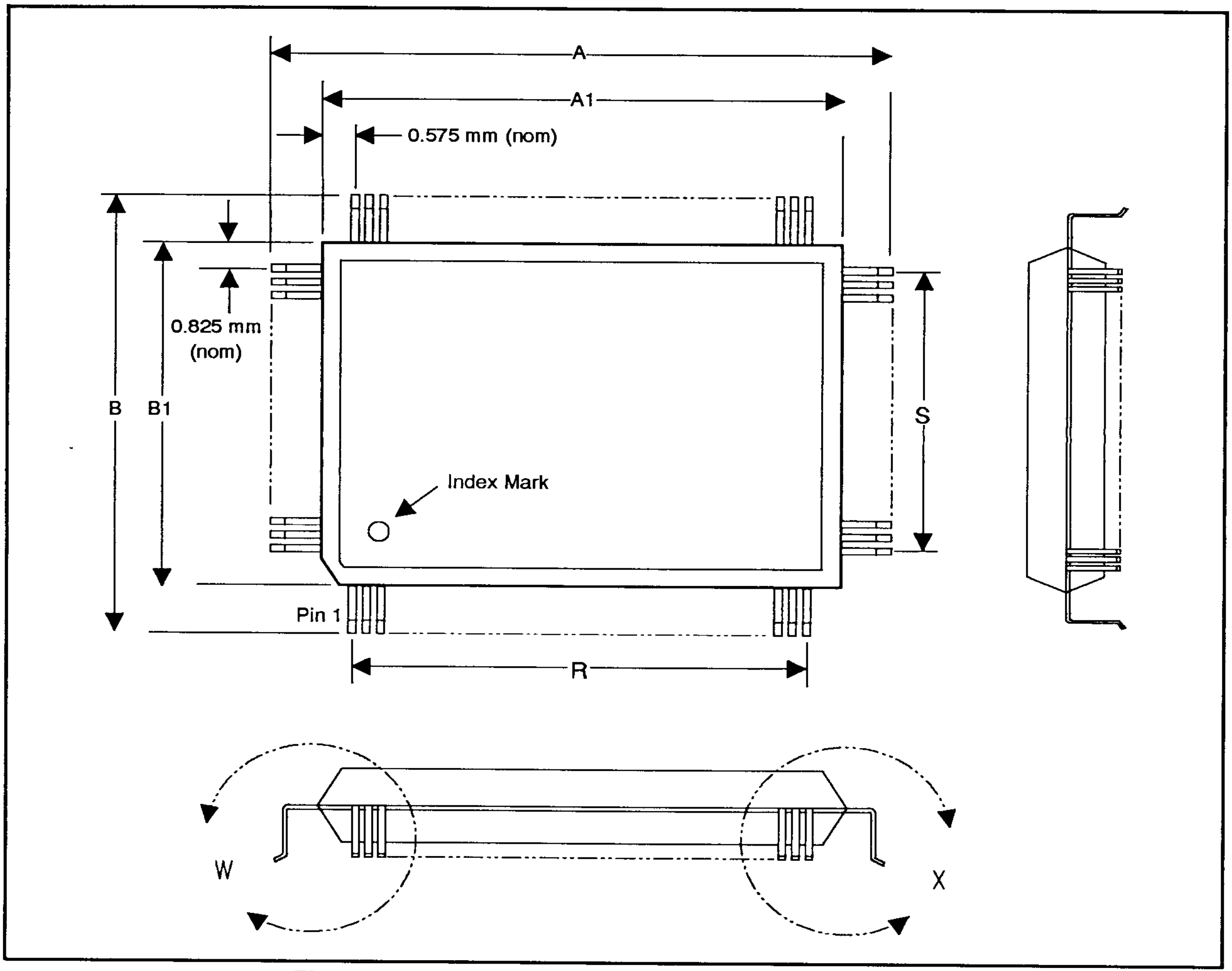


Figure 9-2. 100-Pin Plastic Quad Flat Pack Package

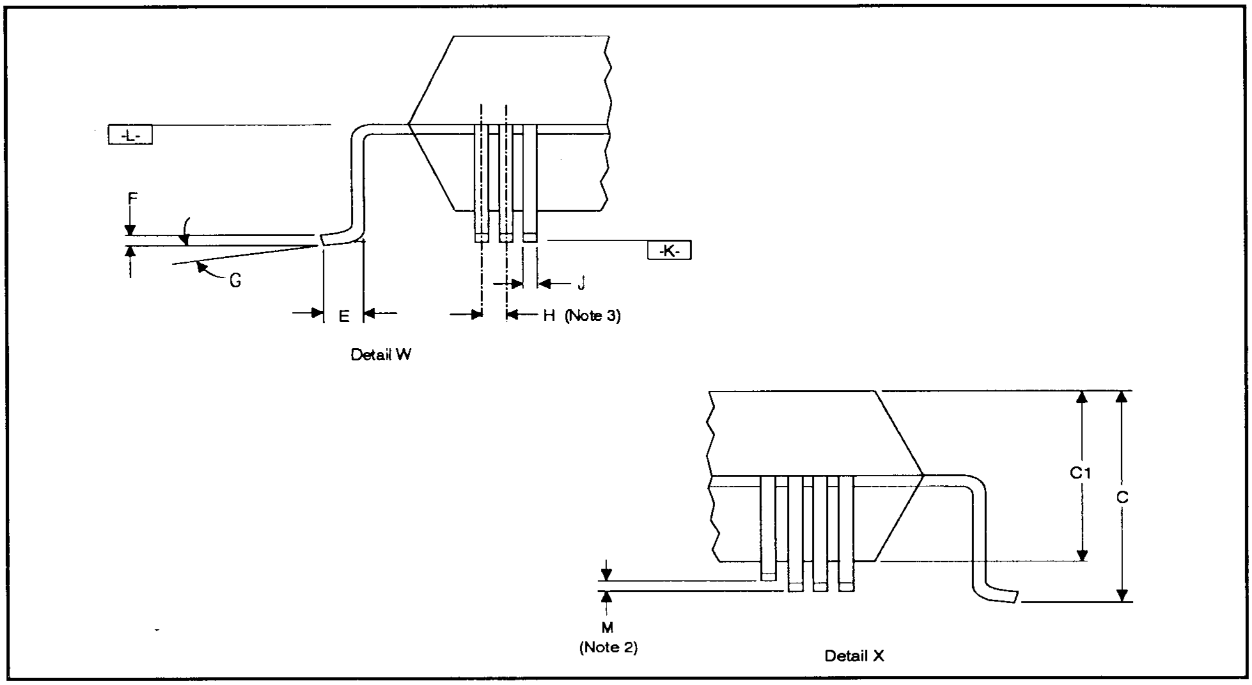


Figure 9-3. 100-Pin Plastic Quad Flat Pack Package

Table 9-1. 100-Pin Plastic Quad Flat Pack Dimensions

millimeters (inches)

Dimensions	Min	Max	Nom
A	23.4 (0.921)	24.2 (0.953)	
A1	19.8 (0.780)	20.2 (0.795)	
B	17.4 (0.685)	18.2 (0.717)	
B1	13.8 (0.543)	14.2 (0.559)	
C		3.15 (0.124)	
C1	2.4 (0.094)	3.0 (0.118)	
E	0.60 (0.024)	1.00 (0.039)	
F	0.10 (0.004)	0.25 (0.010)	
G	0°	10°	
H			0.65 (0.0256)
J	0.20 (0.0079)	0.40 (0.0157)	
M		0.10 (0.004)	
R			18.85 (0.742)
S			12.35 (0.486)

Notes:

1. Controlling dimension: millimeter. Inches rounded to nearest 0.001.
2. Coplanarity of all leads shall be within 0.10 mm (0.004") (difference between highest and lowest lead
   with seating plane -K- as reference).
3. Lead pitch determined at datum -L-.

—This Page Intentionally Left Blank—

Index

Other Stac Products

Part	Description
9704 Data Compression Coprocessor	1.5 Mbytes per second compression throughput 5 Mbytes per second decompression throughput Support for host adapter or embedded controller applications Optional high-speed 16 K FIFO buffer
9707 ECC Coprocessor	Reed-Solomon encoding/decoding and error correction ECC for QIC-112, QIC-40/80 and QIC 525 Programmable block size, frame size, and redundancy Support for nonsequential block numbering due to rewrites
9703 Data Compression Coprocessor	1 Mbyte per second compression throughput 5 Mbytes per second decompression throughput Support for host adapter or embedded controller applications Optional high-speed 16 K FIFO buffer
9820 QIC Formatter	Read/write support for QIC-1350, -525, -380, -120/150, -2110, -100, -24, -11 Read compatibility for QIC-40/80 and other MFM tape formats Programmable all digital phase-lock loop Up to 24 bits of I/O; on-chip oscillator, 8-bit down counter 16-bit event timer (1 µs resolution)

Information furnished by Stac Electronics is believed to be accurate and reliable. However, no responsibility is assumed by Stac Electronics for its use, nor for any infringements of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise under any patent right of Stac Electronics.

5993 AVENIDA ENCINAS • CARLSBAD, CALIFORNIA 92008
Tel (619) 431-7474 • Fax (619) 431-0880