MEMTECH mmT

7225
CONTROLLER FOR 4-MEGABIT BPK 74 BUBBLE MEMORY SUBSYSTEM

D7225-1	0°C to +70°C
D7225-6	−40°C to +85°C

- **Provides Interface between Host Microprocessor or Microcontroller and 4-Megabit Bubble Memory Subsystems**
- **Interfaces to Intel 80 Series and Other Standard Microprocessors and Micro-Controllers**
- **Controls up to Eight Bubble Memory Subsystems**
- **128 Byte FIFO for Data Buffering**
- **Three Modes of Data Transfer**
  - **— DMA**
  - **— Polled**
  - **— Enhanced Interrupt**
- **Transfer of Single (64 bytes) or Multiple Pages of Data**
- **System Power Fail Protection Internally Packaged**

The 7225 is a complete 4-Megabit Bubble Memory Controller (BMC) that provides the interface between the microprocessor host and the 4-Megabit Bubble Memory Subsystem. All communication between the host processor and the bubble memory is performed through the controller.

The BMC interfaces easily to any Intel microprocessor or other standard microprocessors. The user has 18 easy-to-use commands available. Information such as the starting page location, the number of pages to be transferred, and a read or write command is passed to the BMC before the read or write operation is initiated.

The 18 commands of the 4-Megabit BMC are a superset of the 1-Megabit BMC's 16 commands, providing an easy up-grade of software from the 1-Megabit to the 4-Megabit system.

The design engineer writes a bubble memory software driver to integrate the bubble memory into his system. This interfacing with the BMC is similar to interfacing a disk drive controller. Application notes and manuals describe the details of interfacing to the BMC.

The BMC can transfer data in DMA, interrupt, or polled mode. Data is transferred in and out of the bubble memory subsystem to the controller in single or multiple pages. A page size may vary from 64 bytes in a single bubble system up to 512 bytes in an eight bubble system.

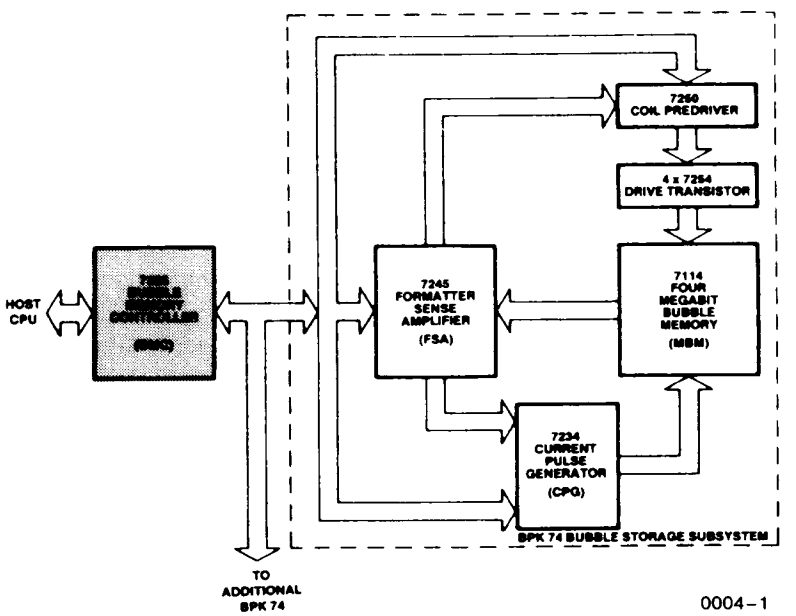


Figure 1. Block Diagram of a 512K Byte Bubble Storage System

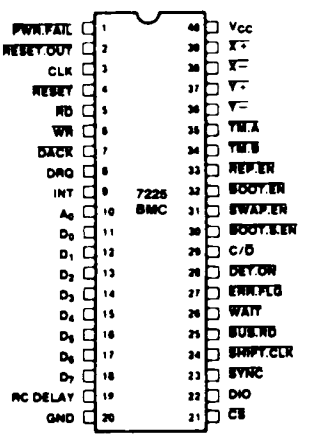


Figure 2. Pin Configuration

October 1987
Order Number: 152625-001

The BMC utilizes an eight bit data bus.

The BMC generates all the timing and control signals to the BPK 74 subsystem.

One BMC can control up to eight BPK 74 subsystems. This provides an easy expansion path to expand any 4-Megabit (512K byte) system up to 32-Megabit (4-Megabyte) by adding on additional BPK 74 subsystems.

The BMC is manufactured using Intel's high performance HMOS III process and is packaged in a standard 40-pin dual-in-line package. All inputs are directly TTL compatible and the device uses a single +5V supply.

HARDWARE DESCRIPTION

The 7225 Bubble Memory Controller is packaged in a 40-pin Dual In-Line Package (DIP). The following lists the individual pins and describes their function.

Table 1. Pin Description

Signal Name	Pin No.	I/O	Source/Destination	Description
$\overline{\text{PWR.FAIL}}$	1	I	7234 CPG	A low forces a controlled stop sequence and holds BMC in an IDLE state. Pulls $\overline{\text{RESET.OUT}}$ low. Requires a pull-up resistor (5.6 KΩ).
$\overline{\text{RESET.OUT}}$	2	O	7250 CPD/7245 FSA 7234 Reference Current Switch	An active low, open-drain signal to disable external logic initiated by $\overline{\text{PWR.FAIL}}$ signal, but not active until a stopping point in a field rotation is reached (if the BMC is causing the bubble memory drive field to rotate).
CLK	3	I	Host Bus	4 MHz ±0.1%, TTL-level clock. 50% ±5% Duty Cycle.
$\overline{\text{RESET}}$	4	I	Host Bus	A low on this pin interrupts any BMC activity, forces a controlled shutdown, and initiates a reset sequence.
$\overline{\text{RD}}$	5	I	Host Bus	A low on this pin enables the BMC output data to be transferred to the host data bus (D_0–D_7). $\overline{\text{CS}}$ or $\overline{\text{DACK}}$ must also be active.
$\overline{\text{WR}}$	6	I	Host Bus	A low on this pin enables the contents of the host data bus (D_0–D_7) to be transferred to the BMC. $\overline{\text{CS}}$ or $\overline{\text{DACK}}$ must also be active.
$\overline{\text{DACK}}$	7	I	Host Bus	A low signal is a DMA acknowledge. This notifies the BMC that the next bus cycle is available to transfer data. This line should be active only when DMA transfer is desired and the DMA ENABLE bit has been set. $\overline{\text{CS}}$ should not be active during DMA transfers except to read status. $\overline{\text{DACK}}$ requires an external pullup to V_{CC} (5.6 KΩ).
DRQ	8	O	Host Bus	A high on this pin indicates that a data transfer between the BMC and the host memory is being requested.
INT	9	O	Host Bus	A rising edge on this pin indicates that the BMC has a new status and requires servicing when enabled by the host CPU.
A_0	10	I	Host Bus	A high on this pin selects the command/status registers. A low on this pin selects the data register.
D_0–D_7	11–18	I/O	Host Bus	Host CPU data bus. An eight-bit bidirectional port which can be read or written by using the $\overline{\text{RD}}$ and $\overline{\text{WR}}$ strobes. D_0 is the LSB.
RC Delay	19	I	Host Bus	Connected to an RC Network with a 85 ms or greater time constant for proper power up. R = 5.6K, C = 15 μF is satisfactory in most typical cases. (V_{RC} < 0.7V for >2 ms after V_{CC} ≥ 4.70V)
GND	20	I		Ground

Table 1. Pin Description (Continued)

Signal Name	Pin No.	I/O	Source/Destination	Description
$\overline{CS}$	21	I	Host Bus	Chip Select Input. A high on this pin disables the device to all but DMA transfers (i.e., it ignores bus activity and goes into a high impedance state).
DIO	22	I/O	7245 FSA	A bidirectional active high data line is used for serial communications with 7245 FSA devices.
$\overline{SYNC}$	23	O	7245 FSA	An active low output utilized to create time division multiplexing slots in a 7245 FSA chain. It also indicates the beginning of a data or command transfer between BMC and 7245 FSA.
$\overline{SHIFT.CLK}$	24	O	7245 FSA	A controller generated clock that initiates data transfer between selected FSAs and their corresponding bubble memory devices. The timing of $\overline{SHIFT.CLK}$ varies depending upon whether data is being read or written to the bubble memory.
$\overline{BUS.RD}$	25	O	To User External Circuit	An active low signal that indicates that the DIO line is in the output mode, i.e., BMC is sending data to FSA. It may be used to allow off-board expansion of 7245 FSA devices.
$\overline{WAIT}$	26	I/O	To Alternate Controller(s) When User System Uses More Than One Controller.	A bidirectional pin that must be tied to the $\overline{WAIT}$ pin on other BMCs when operated in parallel. It indicates that an interrupt has been generated and that the other BMCs should halt in synchronization with the interrupting BMC. $\overline{WAIT}$ is an open collector active low signal. Requires an external pullup resistor to V_{CC} (5.6 KΩ).
$\overline{ERR.FLG}$	27	I	7245 FSA	An active low input generated externally by 7245 FSA indicating that an error condition exists. It is an open collector input that requires an external pullup resistor (5.6 KΩ).
$\overline{DET.ON}$	28	O	To User External Circuit	An active low signal that indicates the system is in the read mode and may be detecting. It is useful for power saving in the MBM.
C/$\overline{D}$	29	O	7245 FSA	A high on this line indicates that the BMC is beginning an FSA command sequence. A low on this line indicates that the BMC is beginning a data transmit or receive sequence.
$\overline{BOOT.SW.EN}$	30	O	7234 CPG	An active low signal is used for enabling the BOOT.SWAP output of the 7234 CPG.
$\overline{SWAP.EN}$	31	O	7234 CPG	An active low signal used to create the swap function in external circuits.
$\overline{BOOT.EN}$	32	O	7234 CPG	An active low signal enabling the bootstrap loop replicate function in external circuitry.
$\overline{REP.EN}$	33	O	7234 CPG	An active low signal used to enable the replicate function in external circuitry.
$\overline{TM.B}$	34	O	7234 CPG	An active low timing signal generated by the decoder logic for determining TRANSFER pulse width.
$\overline{TM.A}$	35	O	7234 CPG	An active low timing signal generated by the decoder logic for determining CUT pulse width.
$\overline{Y-}$, $\overline{Y+}$, $\overline{X-}$, $\overline{X+}$	36–39	O	7250 CPD	Four active low timing signals generated by the decoding logic and used to create coil drive currents in the bubble memory device.
V_{CC}	40	I		+5 VDC Supply

FUNCTIONAL DESCRIPTION

The 7225 Bubble Memory Controller provides the user interface to the bubble memory system. The BMC generates all memory system timing and control, maintains memory address information, interprets and executes user request for data transfers, and provides a Microprocessor-Bus compatible interface for the magnetic bubble memory system.

Figure 3 is a block diagram of the 7225 Bubble Memory Controller (BMC). The following paragraphs describe the functions of the individual functional sections of the BMC.

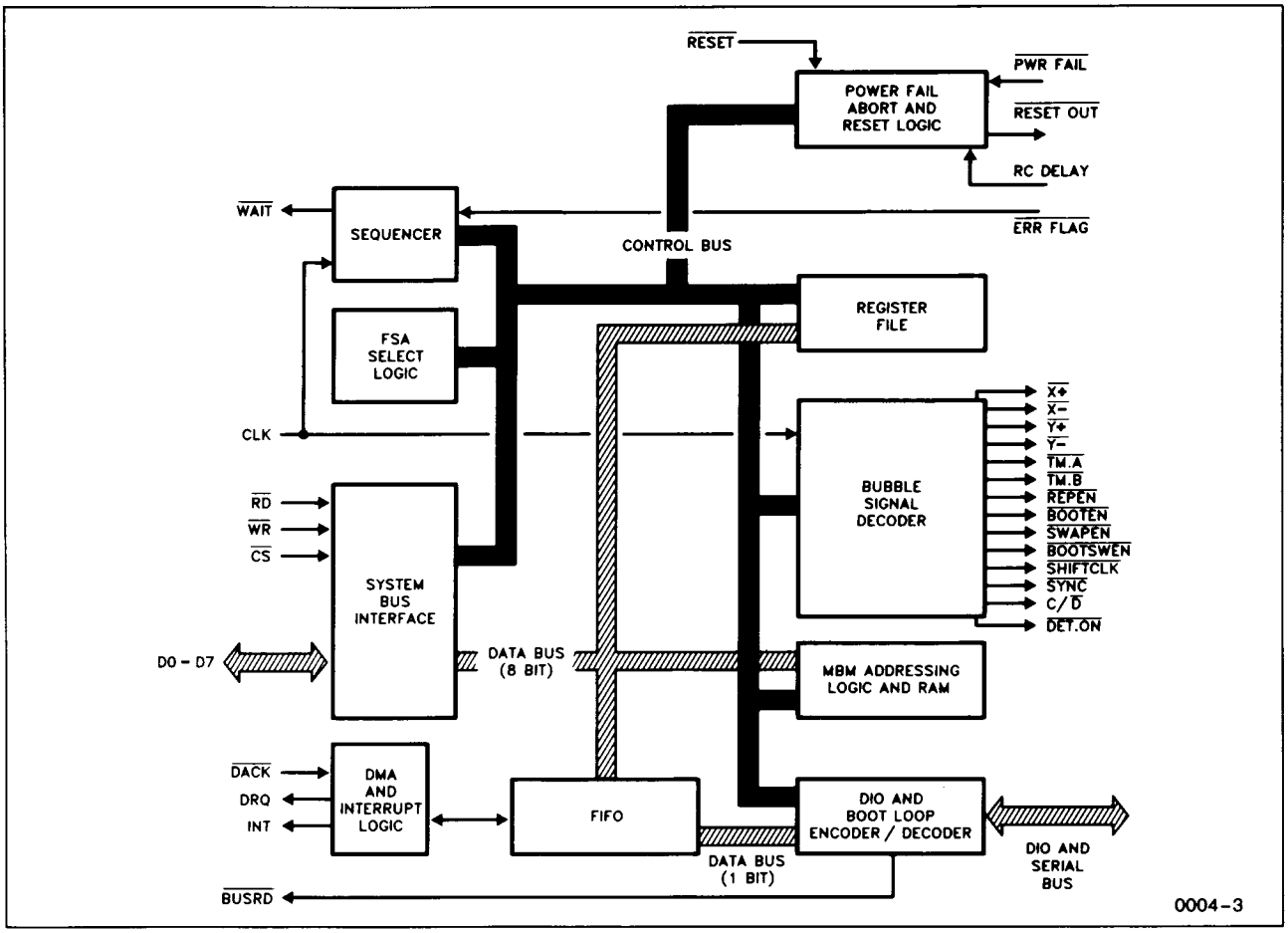


Figure 3. 7225 Bubble Memory Controller (BMC), Block Diagram

System Bus Interface—The System Bus Interface (SBI) logic contains the timing and control logic required to interface the BMC to a non-multiplexed bus. The interface has input data, output data, and status data latches. With a 4 MHz clock, it is capable of sustaining a 1 Mbyte per second transfer rate while data is available in the BMC FIFO.

FIFO—The FIFO consists of a 128 × 8 bit FIFO RAM for data storage. The FIFO block also contains input and output data latches, providing double data buffering, to improve the R/W cycle times seen at the system bus interface. The FIFO may be used as a general purpose FIFO when a command is not being executed by the BMC Sequencer. In this mode, the FIFO READY status bit becomes a FIFO not-empty indicator indicating that the RAM and input/output latches have at least one byte of data.

DMA and Interrupt Logic—The DRQ and INT pins operate as follows:

If the DMA, enable bit in the enable register is set, the DRQ pin, in conjunction with the $\overline{DACK}$ pin, provides a standard DMA transfer capability; i.e., it has the ability to handshake with an 8257 or 9517/8237 DMA controller chip.

The INT pin performs the task of interrupting the host system when the BMC is no longer busy processing a command (Operation Complete or Operation Fail). This can be used by the host system for an interrupt mode of command execution.

Register File—The register file contains 3 user accessible registers, 5 parametric registers, and a data FIFO. These are explained in the register section of the data sheet.

MBM Address Logic and RAM—The MBM address logic consists of the block length counter, starting address counter, adder, and MBM Address RAM. The MBM Address RAM is used to store the next available page address for each of up to 8 dual FSAs. The address maintained is the read address; the write address is generated, when needed, by adding a constant to the stored read address.

The block length counter enables multiple page transfers of up to 2048 pages in length.

The starting address counter is used as a register to hold the desired start address. Once the start address is reached, the counter is incremented on each subsequent page transfer so that its value is equal to the present read address. There are 8192 possible starting addresses.

DIO Bootloop Decoder/Encoder—Generates the $\overline{\text{BUS.RD}}$ signal, which indicates the direction of data transfer on the DIO line (this is useful in situations which require external buffering on the DIO line). This block also contains the circuitry which decodes the bootloop data during a Read Bootloop or Initialize operation, and encodes the bootloop data during a Write Bootloop operation.

Sequencer—Controls the execution of commands by decoding the contents of its own internal ROM in which the BMC firmware is located. This block also sets and resets flags and status bits, and controls actions in other parts of the BMC.

Power Fail and Reset—Provides a means of resetting the bubble systems in an orderly manner, when activated by the $\overline{\text{PWR.FAIL}}$ signal, the $\overline{\text{RESET}}$ signal, or the ABORT command. An RC circuit is connected to Pin 19 on the 7225 to allow a 2 ms delay to guarantee proper power up. An external diode-resistor circuit is connected to Pin 2 on the 7225 to guarantee proper power-up characteristics of $\overline{\text{RESET.OUT}}$.

FSA Select Logic block contains the logic which controls the timing of the interaction between the BMC and the FSAs. The FSA selection is determined by the four high-order bits in the BLR and the three high-order bits in the AR, both set by the user.

Bubble Signal Decoder block contains the logic for creating all the MBM timing signals. The BMC to bubble memory interface consists of active low timing signals. The starting and stopping point of each signal is determined by the decoder logic. Each signal may occur every field rotation or only once in a number of field rotations. The field rotation in which a timing pulse occurs is controlled by the sequencer logic.

Table 2. 7225 BMC Timing (Degrees)**

Signal	Start	Width
$\overline{\text{X+}}$	270°	108°
$\overline{\text{Y+}}$	0°	108°
$\overline{\text{X−}}$	90°	108°
$\overline{\text{Y−}}$	180°	108°
$\overline{\text{TM.A}}$(LATE)	279°	9°
$\overline{\text{TM.A}}$(EARLY)	99°	9°
$\overline{\text{TM.B}}$(LATE)	279°	90°
$\overline{\text{TM.B}}$(EARLY)	99°	90°
$\overline{\text{BOOT.EN}}$	261°	126°
$\overline{\text{REP.EN}}$	261°	126°
$\overline{\text{SWAP.EN}}$	180°	432°
$\overline{\text{BOOT.SW.EN}}$	180°	DC*
$\overline{\text{SHIFTCLK (RD)}}$ LATE	207°	112.5°
EARLY	18°	121.5°
$\overline{\text{SHIFTCLK (WR)}}$ LATE	261°	108°
EARLY	81°	108°

*Stays low for 8211 field rotation periods when writing the MBM Bootloop.
**All phases relative to $\overline{\text{Y+}}$ start phase.

Figure 4 and Table 2 illustrate the typical timing signals for the BMC. These signals are described in the following paragraphs.

$\overline{\text{X+}}$, $\overline{\text{X−}}$, $\overline{\text{Y+}}$, and $\overline{\text{Y−}}$ go to the 7250 CPDs, and are used to enable the coil drive currents in the MBMs.

$\overline{\text{TM.A}}$ and $\overline{\text{TM.B}}$ go to the 7234 CPGs, and are used to determine, respectively, the pulse widths for the CUT and TRANSFER functions used in replicating and generating the bubbles.

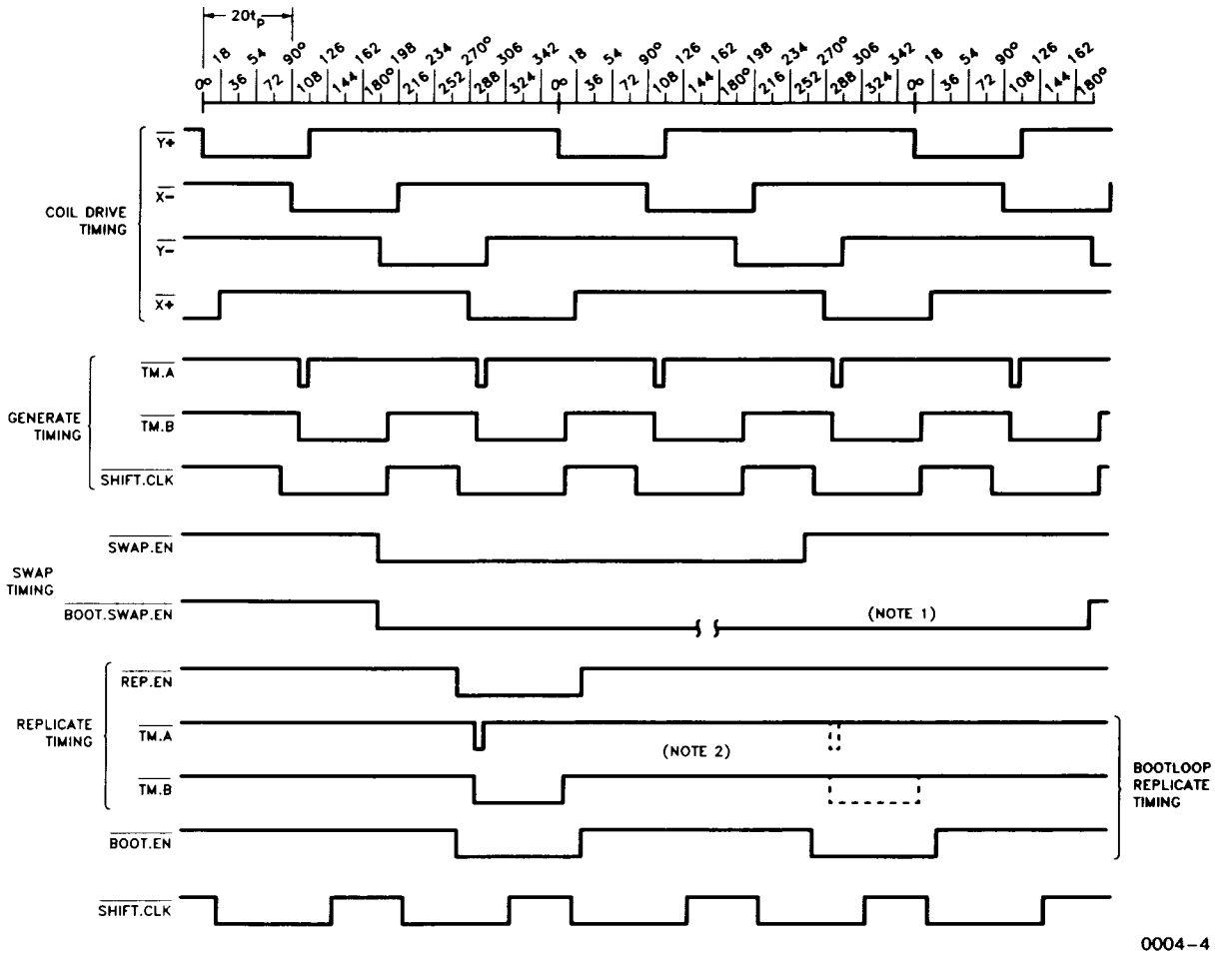
$\overline{\text{SWAP.EN}}$, $\overline{\text{REP.EN}}$ $\overline{\text{BOOT.SW.EN}}$, and $\overline{\text{BOOT.EN}}$ all go to the 7234 CPG. They are used to enable, respectively, the data swap, data replicate, boot swap, and boot replicate functions within the MBMs.

$\overline{\text{SHIFT.CLK}}$ goes to the FSAs. It is used to control the timing of events at the interface between each FSA and its corresponding MBM. (Refer to 7245 FSA Specification for a description of the BMC/FSA interface.)

$\overline{\text{SYNC}}$ and C/$\overline{\text{D}}$ control the serial communications between the BMC and the FSAs (on the DIO line).

USER-ACCESSIBLE REGISTERS

The user operates the bubble memory system by reading from or writing to specific registers within the bubble memory controller (BMC). The following paragraphs identify these registers and gives brief functional descriptions, including bit configurations and address assignments.



NOTES:
1. BOOT.SWAP.EN stays low for 8211 field rotation periods when writing MBM bootloop.
2. One TM.A, TM.B and REP.EN pulse for data reads; 7694 to 15886 (approx.) TM.A, TM.B and BOOT.EN pulses for bootloop reads.

Figure 4. 7225 BMC Timing Diagram

Register Addressing

Selection of the user-accessible registers depends on register address information sent from the user to the BMC. This address information is sent via a single address line (designated A_0) and data bus lines D_0 through D_5.

The Command Register (CMDR) is an 8-bit register which is loaded from D_0–D_7. Register Address Counter (RAC) is a 4-bit register which is loaded from D_0–D_3. The status register is selected and read by a single read request. The command register is selected and loaded by a single write request. The remaining registers are accessible indirectly, and the desired register is first selected by placing its address in the RAC, and then read or written with a subsequent read or write request.

Table 3 gives a complete listing of the address assignments for the user-accessible registers. The registers are listed in two groups. The first group (STR, CMDR, RAC) consists of the user-accessible registers that are selected and accessed in one operation. The second group (BLR, ER, AR, FIFO) consists of those registers that are addressed indirectly by the contents of RAC.

Table 3. Address Assignments for the User-Accessible Registers

A0	D7	D6	D5	D4	D3	D2	D1	D0	Symbol	Name of Register	Read/Write
1	0	0	C	1	C	C	C	C	CMDR	Command Register	Write Only
1	0	0	M	0	B	B	B	B	RAC	Register Address Counter	Write Only
1	S	S	S	S	S	S	S	S	STR	Status Register	Read Only

Table 3. Address Assignments for the User-Accessible Registers (Continued)

RAC Address A0	B3	B2	B1	B0	Symbol	Name of Register	Read/Write
0	1	0	1	1	BLR LSB	Block Length Register LSB	Write Only
0	1	1	0	0	BLR MSB	Block Length Register MSB	Write Only
0	1	1	0	1	ER	Enable Register	Read or Write
0	1	1	1	0	AR LSB	Address Register LSB	Read or Write
0	1	1	1	1	AR MSB	Address Register MSB	Read or Write
0	0	0	0	0	FIFO	FIFO Data Buffer	Read or Write

SSSSSSSS = 8-bit status information returned to the user from the STR
C1CCCC = 6-bit command code sent to the CMDR by the user.
BBBB = 4-bit register address sent to the RAC by the user.
B3B2B1B0 = 4-bit contents of RAC at the time the user makes a read or write request with A0 = 0.
LSB = Least Significant Byte
MSB = Most Significant Byte
M = Modifier (When written high will clear any pending interrupt from 7225 without destroying any data present in the FIFO and its associated latches.

The register file contains the registers with address 1011 through 1111. These registers are also called parametric registers because they contain flags and parameters that determine exactly how the BMC will respond to commands written to the CMDR.

To facilitate programming, the BMC automatically increments the RAC by one count after each transfer of data to or from a parametric register.

The RAC increments from the initially loaded value through address 1111 and then on to 0000 (the FIFO address). When it has reached 0000, it no longer increments. All subsequent data transfers (with A0 = 0) will be to or from the FIFO until the RAC is loaded with a different register address.

REGISTER DESCRIPTIONS

Command Register (CMDR) 5 Bits, Write Only

The user issues a command to the BMC by writing a 5-bit command code to the CMDR.

Table 4 lists the 5-bit command codes used to issue the eighteen commands recognized by the BMC.

Table 7 lists the commands and describes their functions.

Table 4. Command Code Definitions

D5	D3	D2	D1	D0	Command Name
0	0	0	0	1	Initialize
0	0	0	1	0	Read Bubble Data
0	0	0	1	1	Write Bubble Data
0	0	1	0	0	Read Seek
0	0	1	0	1	Read Bootloop Register
0	0	1	1	0	Write Bootloop Register
0	0	1	1	1	Write Bootloop
0	1	0	0	0	Read FSA Status
0	1	0	0	1	Abort
0	1	0	1	0	Write Seek
0	1	0	1	1	Read Bootloop
0	1	1	0	0	Read Corrected Data
0	1	1	0	1	Reset FIFO
0	1	1	1	0	MBM Purge
0	1	1	1	1	Software Reset
1	0	0	0	0	Write Bootloop Register Masked
1	0	0	0	1	Zero Access Read Seek
1	0	0	1	0	Zero Access Read Bubble Data

The most commonly used commands in normal operation are:

Initialize
Read Bubble Data
Write Bubble Data
Reset FIFO
Read Seek
Write Seek
Abort
Read FSA Status
Zero Access Read Seek
Zero Access Read Bubble Data

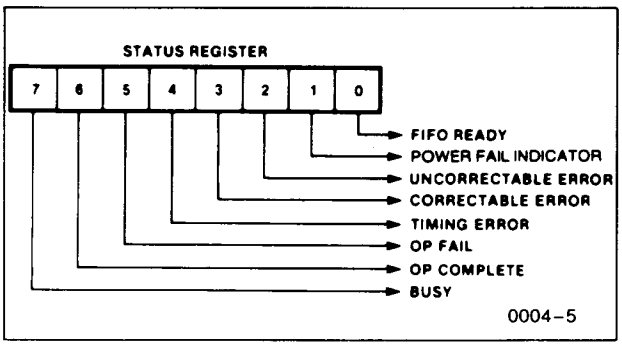
Commands relating to the bootloop, and used only for diagnostic purposes, are:

Read Bootloop Register
Write Bootloop Register
Write Bootloop Register Masked
Read Bootloop
Write Bootloop
Read Corrected Data
Software Reset
MBM Purge

Status Register (STR) 8 Bits, Read Only

The user reads the BMC status register in response to an interrupt signal or as part of the polling process in a polled data transfer mode. The status register provides information about error conditions, completion or termination of commands, and the BMC's

readiness to transfer data or accept new commands. The individual bit descriptions are as follows:



BUSY (when = 1) indicates that the BMC is in the process of executing a command. When equal to 0, BUSY indicates that the BMC is ready to receive a new command.

OP COMPLETE (when = 1) indicates the completion of a microcode command sequence.

OP FAIL (when = 1) indicates that the BMC is unable to complete a microcode command sequence. The TIMING, CORRECTABLE ERROR, and/or UNCORRECTABLE ERROR bits indicate the cause of the OP FAIL.

TIMING ERROR (when = 1) indicates that a FSA has reported a timing error to the BMC, or that the host system has failed to keep up with the BMC, thereby causing the BMC FIFO to overflow or to underflow. TIMING ERROR is also set if no bootloop sync word is found during initialization, or if a Write Bootloop command is issued when the WRITE BOOTLOOP ENABLE bit is equal to zero in the enable register, or the Write Bootloop Register Masked command is sent without an adequate number of 1's present in data pattern.

CORRECTABLE ERROR (when = 1) indicates that an FSA has reported to the BMC that a correctable error has been detected in the last data block transferred.

UNCORRECTABLE ERROR (when = 1) indicates that at least one FSA has reported to the BMC that an uncorrectable error has been detected in the last data block transferred.

POWER FAIL INDICATOR (when = 1) indicates that the BMC has experienced a power fail or power up. Only the Abort command can reset this bit.

FIFO READY has two functions, which are as follows:

NOTE: If RAC ≠ FIFO, FIFO READY = 1

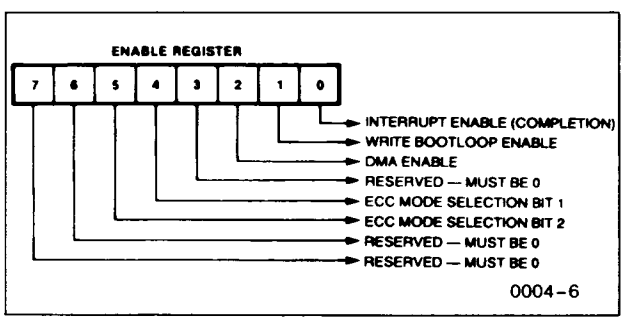
Status Bits		Read	Write
FIFO READY	BUSY		
1	1	data in FIFO	space in FIFO
0	1	no data	no space
1	0	— data in FIFO —	
0	0	—FIFO empty—	

Although status word can be read at any time, the status information, bit 1 through 6, is not valid until the BUSY bit is low.

STR Bits 1 through 6 are reset when a new command is issued. They may also be reset by writing to the RAC (A0 = 1 and D4 = 0) with the Modifier Bit set to a logic one (D5 = 1). This operation also resets the "INT" pin to a logic zero. Note: A byte of FIFO data can be lost when using this procedure if the RAC is written to with the address of any parametric register other than the FIFO when data is still present in the FIFO.

Enable Register (ER) 8 Bits, Write Only

The user sets various bits of the enable register to enable or disable various functions within the BMC or the FSAs. The individual bit descriptions are as follows:



INTERRUPT ENABLE COMPLETION enables the BMC to interrupt the host system (VIA the INT Line) when a command execution has been completed (BUSY = 0.)

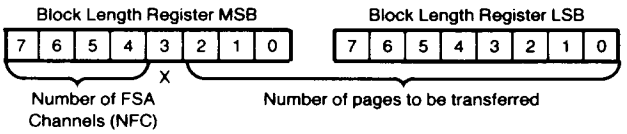
WRITE BOOTLOOP ENABLE (when = 1) enables the bootloop to be written. If this bit is equal to zero and a Write Bootloop command is received by the BMC, the command is aborted and the TIMING ERROR bit is set in the STR.

DMA ENABLE (when = 1) enables the BMC to operate in DMA data transfer mode, using the DRQ and $\overline{\text{DACK}}$ signals with an external DMA controller.

ECC Mode Selection Bits (1 and 2) are used to enable various options of the internal error correction scheme. These options are discussed in the software section of this data sheet.

Block Length Register (BLR) 16 Bits, Write Only

The contents of the block length register determine the system page size and also the number of pages to be transferred in response to a single bubble data read or write command. The bit configuration is as follows:

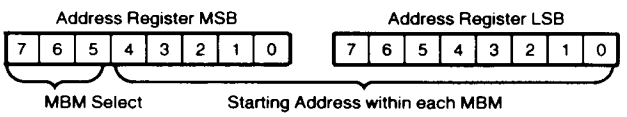


The system page size is proportional to the number of magnetic bubble memory modules (MBMs) operating in parallel during the data read or write operation. Each MBM requires two FSA channels. Bits 4 through 7 of BLR MSB actually specify the number of FSA channels to be accessed.

The BLR LSB, together with the 3 least significant bits of the BLR MSB, specify the number of pages to be transferred. Up to 2048 pages can be transferred in response to a single bubble data read or write command, hence the requirement for 11 bits. All 11 bits equal to zero specifies a 2048 page transfer.

Address Register (AR) 16 Bits, Read or Write

The contents of the address register determine which MBM group is to be accessed, and, within that group, what starting address location shall be used in a data read or write operation. The bit configuration is as follows:



Within each MBM there 8192 possible starting address locations for a data read or write operation, hence the requirement for 13 bits in the starting address field.

The selection of the MBMs to be read or written is specified by AR MSB Bits 5–7. The BMCs interpretation of these bits depends on the number of MBMs in a group, which is specified by BLR MSB Bits 4–7.

Table 6 shows which MBM groups are selected in response to given values for BLR MSB Bits 4–7 and AR MSB Bits 5–7. A 4-megabyte system (8 MBMs) is represented, with the FSA channels numbered 0 through F.

Table 6. Selection of FSA Channels

AR MSB Bits (7,6,5)	BLR MSB Bits (7,6,5,4)				
	0000	0001	0010	0100	1000
000	0	0,1	0,1,2,3	0 to 7	0 to F
001	1	2,3	4,5,6,7	8 to F	
010	2	4,5	8,9,A,B,		
011	3	6,7	C,D,E,F		
100	4	8,9			
101	5	A,B			
110	6	C,D			
111	7	E,F			

FIFO Data Buffer (FIFO) 128 x 8 Bits, Read or Write

The BMC FIFO is a 128-byte buffer through which data passes on its way from the FSAs to the user, or from the user to the FSAs. The FIFO allows the data transfer to proceed in an asynchronous and flexible manner, and relaxes timing constraints, both to the FSAs and also to the user's equipment. The user's system must, however, meet the data rate requirements. When the BMC is busy (executing a command) the FIFO functions as a data buffer. When the BMC is not busy, the FIFO is available to the user as a general purpose FIFO.

Table 5. 4 Mbyte System Page Size, Page Address Range, and Data Transfer Performance Configuration

BLR MSB				NFC	System Page Size	# of Pages	Address Range	MBM Data Transfer Rate
7	6	5	4					
0	0	0	1	2	64 byte	64K	0000—FFFF	25K bytes/sec
0	0	1	0	4	128 byte	32K	0000—7FFF	50K bytes/sec
0	1	0	0	8	256 byte	16K	0000—3FFF	100K bytes/sec
1	0	0	0	16	512 byte	8K	0000—1FFF	200K bytes/sec

FUNCTIONAL OPERATION

The IC components used in the bubble memory systems have been designed with transparency in mind—that is, a maximum number of operations are handled by the hardware and firmware of these components.

Each four megabit bubble memory (MBM) operates in its own domain, and is unaffected by the number of bubble memories in the system. The roles played by the MBM's immediate support circuitry can be described as if the system contained only one MBM module.

Data Flow within the Magnetic Bubble Memory (MBM) System (Single MBM Systems)

During a read operation, data flows as follows: The data from the MBM is input to the Formatter/Sense Amplifier (FSA). Data from each channel (A channel or B channel) of the MBM goes to the corresponding channel of the FSA. In the FSA, the data is paired up with the corresponding bit in the FSA's bootloop register to determine whether it represents data from a 'good' loop. If it does, the data bit is stored in the FSA FIFO. Error detection and correction is applied to each block of 256 data bits.

From the FSA FIFO, data is sent to the bubble memory controller (BMC) in the form of a serial bit stream, via a one-line bidirectional data bus (DIO). The data is multiplexed onto the DIO line, with data bits coming alternately from the A and B channels of the FSA. The BMC outputs a $\overline{\text{SYNC}}$ pulse to the $\overline{\text{SELECT.IN}}$ input of the FSA. The FSA responds by placing a data bit from the A channel FIFO on the DIO line. One clock cycle later, a data bit from the B channel FIFO is placed on the DIO line. The BMC continues to output $\overline{\text{SYNC}}$ pulses, once every 40 clock cycles for command (20 clock cycles for data), each time receiving two data bits in return.

In the BMC, the data undergoes serial-to-parallel conversion and is assembled into bytes, which are then placed in the BMC FIFO, which can hold 128 bytes of data. From this FIFO, the data bytes are read by the host system.

During a write operation, the data flow consists of the corresponding operations in the reverse order.

INTERFACING REQUIREMENTS

All communications between the host microprocessor and the bubble memory is performed through the 7225 BMC. The general principles are described below.

The hardware interfacing requirements will be discussed first, followed by the software interfacing requirements.

Table 7. Detailed Command Descriptions

Initialize	The BMC executes the Initialize command by first interrogating the bubble system to determine how many FSAs are present, then reading and decoding the bootloop from each MBM and storing the results in the corresponding FSA's bootloop register. All the parametric registers must be properly set up before issuing the Initialize command.
Read Bubble Data	The Read Bubble Data command causes data to be read from the MBMs into the BMC FIFO. The selection of the MBMs to be accessed and the starting address for the read operation is specified in the address register (AR). The block length register (BLR) specifies the number of system pages to be read. All the parametric registers must be properly set up before issuing the Read Bubble Data command.
Write Bubble Data	The Write Bubble Data command causes data to be read from the BMC FIFO and written into the MBMs. The selection of the MBMs to be accessed and the starting address for the write operation is specified in the address register (AR). The block length register (BLR) specifies the number of system pages to be written. All the parametric registers must be properly set up before issuing the Write Bubble Data command.

Table 7. Detailed Command Descriptions (Continued)

Read Seek	The Read Seek command rotates the selected MBMs to a designated page address location. No data transfer occurs. The positioning is such that the next data location available to be read is the specified (in AR) page address plus one. The Read Seek command may be used to reduce latency (access time) in cases where information is available for the user to predict the location of an impending read reference to the MBMs.
Write Seek	The Write Seek command rotates the selected MBMs to a designated page address location. No data transfer occurs. The positioning is such that the next data location available to be written is the specified (in AR) page address plus one. The Write Seek command may be used to reduce latency (access time) in cases where information is available for the user to predict the location of an impending write reference to the MBMs.
Abort	The Abort command causes a controlled termination of the command currently being executed by the BMC. The Abort command is the only command accepted by the BMC (and is typically issued) when the BMC is busy.
MBM Purge	The MBM Purge command clears all BMC registers, counters, and the MBM address RAM. Furthermore, it determines how many FSA channels are present in the system and stores this value in the 7225. The "INITIALIZE" command uses this command as a subroutine.
FIFO Reset	The FIFO Reset command clears the BMC FIFO, its input and output latches, and the BMC input latch.
Read Corrected Data	The Read Corrected Data command causes the BMC to read into the BMC FIFO a 256-bit block of data from the FIFO of each selected FSA channel, after an error has been detected. The data cycles through the error correction network of the FSA. After the data has been read, the FSA reports to the BMC whether or not the error was correctable.
Software Reset	The software Reset command clears the RAC pointer. It also causes the BMC to send the Software Reset command to selected FSAs in the system. No reinitialization is needed after this command.
Read FSA Status	The Read FSA Status command causes the BMC to read the 8-bit status register of all FSAs, and to store this information in the BMC FIFO.
Read Bootloop Register	The Read Bootloop Register command causes the BMC to read the bootloop register of the selected FSA channels and to store this information in the BMC FIFO. The 80 bytes in the bootloop registers are transferred.
Write Bootloop Register Masked	Proper operation of the FSAs during data transfer to or from the MBMs requires that the bootloop register contain at least 540 logic 1s for each FSA bootloop register. The user may manually write the 80 byte bootloop code for the MBM into the BMC's FIFO and issue the Write Bootloop Register Masked command. This in turn counts the first 540 logical 1's (270 per side) within the bootloop, masks logic 0's after the 540 logic 1's, and writes this data into the bootloop registers of the selected FSA. This performs the redundancy mapping for the MBM.
Write Bootloop Register	Performs exactly like the Write Bootloop Register Masked command with the exception that it does not mask logic 0's after the first 540 logic 1's (270 per side). Instead, it writes the bootloop code to the FSA's bootloop register exactly as it is written into the BMC FIFO.
Read Bootloop	The Read Bootloop command causes the BMC to read the bootloop from the selected MBM, and to store the decoded bootloop information in the BMC FIFO. The Initialize command uses this command as a subroutine.
Write Bootloop	The Write Bootloop command causes the existing contents of the selected MBM's bootloop to be replaced by new bootloop data based on 80 bytes of information stored in the FIFO. Encoding of the bootloop data is done by the BMC hardware.

Table 7. Detailed Command Descriptions (Continued)

Zero Access Read Bubble Data	The Zero Access Read Bubble Data command functions exactly the same as the Read Bubble Data command except it must be preceeded by the Zero Access Read Seek command. The parametric registers are written prior to the Zero Access Read Seek command and should not be rewritten for the Zero Access Read Bubble Data command.
Zero Access Read Seek	The Zero Access Read Seek command operates similarly to the Read Seek command, but it reads the first page of data from the MBM into the FSA(s) FIFO. This eliminates the first page overhead involved in the Read Bubble Data command. The latency for the first page data is only the time required to read the data from the FSA. The values written into the parametric registers prior to the issuance of the Zero Access Read Seek command are identical to those written for a Read Bubble Data command. The seeking address for the Zero Access Read Seek command is equal to the desired read address.

HARDWARE INTERFACE REQUIREMENTS

User Interface Signals

The source, destination and function of the user interface signals are described in Table 1 in the data sheet.

System Timing

As shown on the timing diagrams in the WAVEFORM section, the typical read/write cycle timing provides sufficient tolerance to allow most currently available microprocessors to be easily adapted to the BMC timing requirements.

User Data Transfer Rate Requirements

The maximum data rate for the user interface is a function of the number of MBMs operated in parallel as outlined in Table 8. The rates listed must be considered in relation to the data transfer mode (polled, interrupt, or DMA) to be implemented in order to be sure that the host system software and hardware are capable of keeping up with the data transfer. In other words, the BMC requires the host CPU to be able to sustain the maximum data rate transfer rate for the minimum data transfer (e.g., for a one bubble system keep up the transfer rate for at least 64 bytes = one page).

Table 8. User Data Transfer Rate Requirements

Number of MBMs Operating in Parallel	Burst Data Transfer Rate between BMC FIFO and the FSAs
1	25 kbytes/second
2	50 kbytes/second
4	100 kbytes/second
8	200 kbytes/second

Hardware Interfacing for Data Transfer

The BMC supports three data transfer modes: DMA, interrupt, and polled.

To support DMA, an external hardware mechanism is required for servicing the BMC's data transfer requests. While several hardware implementations are possible, one common configuration is the Intel 8257 DMA controller.

To support an interrupt-driven system an Intel 8259 Programmable Interrupt Controller is often used.

The polled data transfer mode relies almost exclusively on the software interaction between the host processor and the BMC to control the transfer of data.

Multiple MBM-System

A BMC is capable of processing data and of supplying the required timing and control signals for operating up to eight Bubble Storage Units (BSUs), each of which is capable of storing 512 kbytes of user data. A BSU consists of a 512 kbyte MBM and its seven immediate IC support chips (i.e., a BPK 74 Kit).

SOFTWARE INTERFACE REQUIREMENTS

To use the BMC, the user has to write a "bubble memory software driver".

The bubble driver is responsible for all the system interaction with the bubble memory controller and is intrinsic to the efficient and reliable operation of the bubble system. The driver accepts bubble memory commands and command execution parameters from the application program, controls and monitors command execution, and returns operational status information to the application program at command completion. To perform all of these operations, the bubble driver must support the bit-to-byte level of the bubble memory controller's command and status registers and the parametric registers that define the operating mode, system configuration, and extent of the transfer.

The level of the software driver complexity is a function of the specific application needs. Regardless, a set of basic drivers must be developed that in turn are integrated into a system at the appropriate level. If an application program is small and simple, a basic bubble driver may simply be called from the main program.

At the highest level of driver sophistication, the application program treats the bubble system as a collection of named data areas of files similar to the way in which data is stored and retrieved in disk operating systems. At the file system level, an application program can ignore the mechanics of bubble storage and access and merely present a file name to the driver to open, read, or write, then close the desired bubble file.

Data Organization

From a software viewpoint, data logically is organized into blocks of bytes called pages. During data transfer operations, one or more of these pages are transferred between the bubble(s) and the host microprocessor. A page is the smallest increment of data that can be transferred; single bytes cannot be transferred. Conceptually, the data organization within a bubble memory is analogous to a disk system. Just as disk sector sizes are fixed when a disk is formatted, bubble page sizes are established under software control when the bubble system is initialized.

For a single bubble system, the page size is fixed at 64 bytes. The total number of pages available is 8192. In systems with multiple bubbles, page size can vary from 64 bytes to 512 bytes depending on the number of bubble devices in the system. Page size is directly proportional to the system data rate and also determines the total number of available pages (address field size). The selection of the appropriate page size depends primarily on the data rate supported by the system. The higher the data rate, the faster the microprocessor must respond to the demands of the bubble memory controller.

Buffering

The Bubble Memory Controller includes a FIFO data buffer that reconciles timing differences between the host microprocessor and the bubble system. Being 128 bytes long, the BMC is capable of fully buffering two 64-byte pages between it and the user. For data transfers over 128 bytes (for a single Read or Write operation), it is necessary for the user to meet data rates that will prevent the FIFO from overflowing or underflowing.

Command Execution

Command execution can be performed either in interrupt driven mode or in polled mode irrespective of the data transfer mode (polled, interrupt-driven, or DMA).

Data Transfer Mode

As described earlier in the hardware section, three data transfer modes are available (polled, interrupt-driven or DMA).

System performance and additional hardware and software overhead are all important considerations when choosing the appropriate mode for your application.

Error Correction

The bubble memory system utilizes a built-in 28 bit/page fire code to accomplish the error detection/correction. Four options of error correction are available with the Intel 7225 based bubble memory system. These options are selected using the ECC Mode Selection Bits 1 and 2 in the Enable Register. Table 9 lists the options:

Table 9. ECC Options

ECC MODE SELECTION 2	ECC MODE SELECTION 1	INTERRUPT ACTION	ECC OPTION
0	0	NO INTERRUPT	OPTION 1
1	0	INTERRUPT ON UCE	OPTION 2
0	1	INTERRUPT ON UCE, CE	OPTION 3
1	1	INTERRUPT ON UCE, CE	OPTION 4

OPTION 1 identifies uncorrectable errors in block transfers and responds by setting the UCE bit in the status register after the transfer is complete. Thus, the user will know if erroneous data has been transferred. No interrupt to the host occurs. Correctable errors are completely transparent, being corrected by the error correction network. The CE bit in the status register will be set even though the data was corrected and transferred. Option 1 would typically be used for small block transfers.

OPTION 2 identifies uncorrectable errors with respect to a single page in a block transfer. The BMC will generate an interrupt to the host to inform it of the UCE. The AR and BLR will not be altered so that it is possible to determine where the error occurred. However, the page that contained the UCE will already have been transferred to the BMC's FIFO and read by the host. Thus, a retry of that page should be done, knowing that erroneous data has been transferred to the host system. Correctable errors are again transparent, being corrected by the ECC network and transferred.

OPTION 3 is used mostly for diagnostic purposes. It identifies both UCE and CE to a single page. No data is transferred if either error is detected. Neither the AR nor the BLR is altered so that retries are possible.

OPTION 4 is very similar to Option 3 in that it also identifies both UCE and CE. However, the error will be identified to a specific FSA channel within any given page and thus is useful as a diagnostic for a multiple bubble system. No data is transferred if an error is detected.

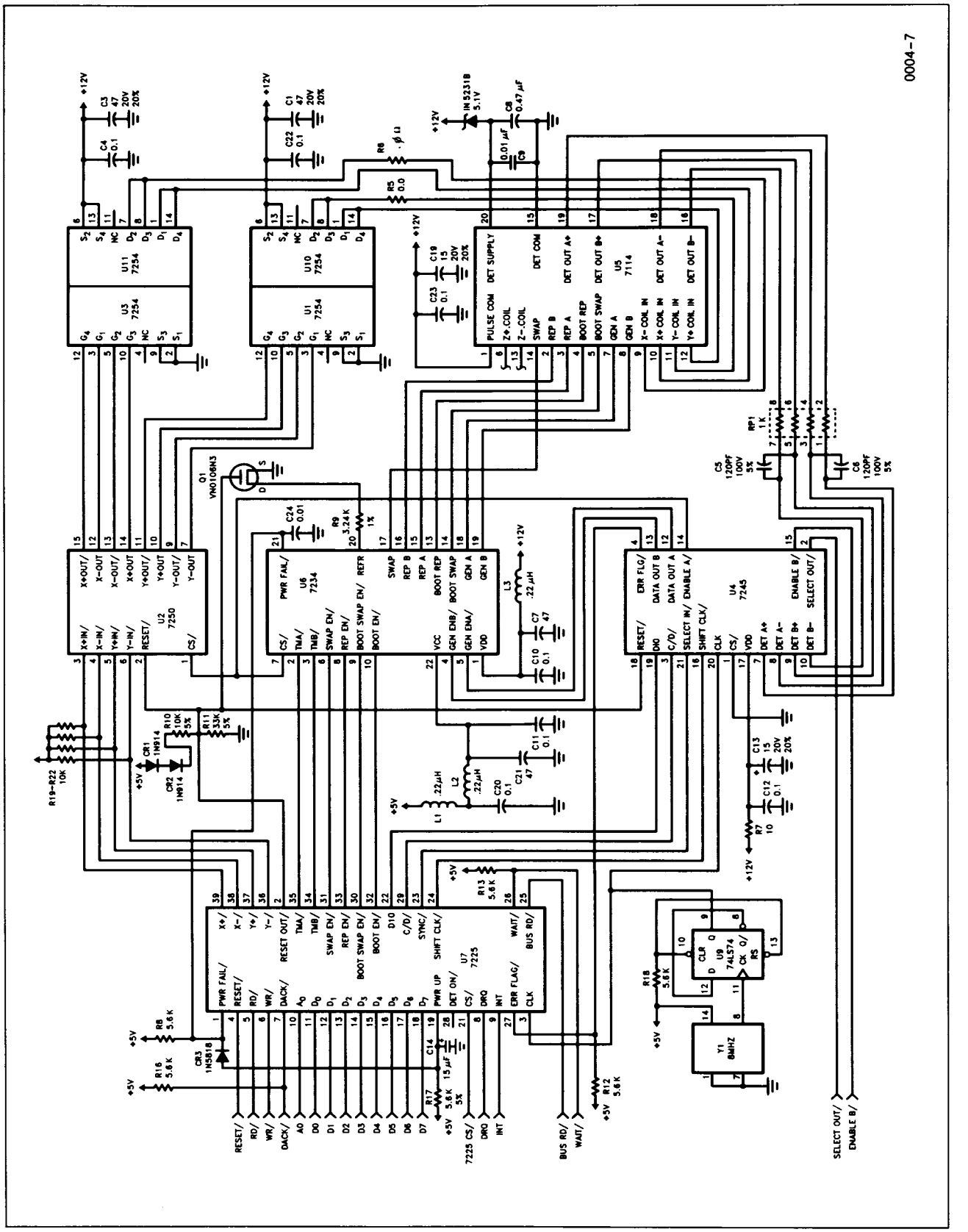
Communication with the BMC

All communications between the host and the bubble memory actually are performed through the BMC. The BMC has two input/output (I/O) ports, an eight-bit bidirectional data port, and an eight-bit command/status port. Conceptually, a bubble memory system can be thought of as a disk system in that data in the bubble memory is organized into blocks called pages that are similar to disk sectors. Information such as starting page location, direction of transfer, and the number of pages to be transferred is passed to the BMC before the desired read or write operation is initiated.

The general procedure for communicating with the BMC is:

1. Set-up the BMC for data transfer communication by loading specific parameters in user-accessible registers.
2. Send the desired command.
3. Read the status register to determine if command is accepted.
4. If applicable, transfer (i.e., read or write) data.
5. Read the status register until BMC is not busy (or under some conditions "INT" pin).
6. Examine the status register to determine whether the operation was successful.

4 MEGABIT SYSTEM SCHEMATIC



ABSOLUTE MAXIMUM RATINGS*

Temperature under bias ..... −40°C to +100°C
Storage Temperature ..... −65°C to +150°C
All Input or Output Voltages
V_{CC} Supply Voltage ..... −0.5 to 7V

**Notice: 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 for extended periods may affect device reliability.*

D.C. CHARACTERISTICS

D7225-1 $T_A = 0°C$ to $+70°C$, $V_{CC} = 5.0V +5\%, -10\%$
D7225-6 $T_A = -40°C$ to $+85°C$, $V_{CC} = 5.0V +5\%, -10\%$

Symbol	Parameter	Min	Max	Units	Test Condition
V_{IL}	Input Low Voltage		0.8	V	
$V_{IH(1)}$	Input High Voltage (all but $\overline{PWR.FAIL}$, RC Delay, and $\overline{DACK}$	2.0	V_{CC} + 0.5	V	
$V_{IH(2)}$	Input High Voltage ($\overline{PWR.FAIL}$) and RC Delay	2.5	V_{CC} + 0.5	V	
V_{IH}	$\overline{DACK}$	2.0	V_{CC} + 0.5		Not Tested
$V_{OL(1)}$	Output Low Voltage (All outputs except $\overline{DET.ON}$, $\overline{BUS.RD}$, $\overline{SHIFT.CLK}$, $\overline{SYNC}$, and $\overline{WAIT}$		0.45	V	I_{OL} = 3.2 mA V_{CC} = +0.25V
$V_{OL(2)}$	Output Low Voltage $\overline{DET.ON}$, $\overline{BUS.RD}$, $\overline{SHIFT.CLK}$, $\overline{SYNC}$		0.45	V	I_{OL} = 1.6 mA V_{CC} = +0.25V
V_{OL}	$\overline{WAIT}$		0.8		Not Tested
V_{OH}	Output High Voltage	2.4		V	I_{OH} = 400 µA
I_{IN}	Input Leakage Current		10	µA	$0 \le V_{IN} \le V_{CC}$
$\lvert I_{OFL} \rvert$	Output Float Leakage DIO, D_0–D_7, RC Delay		10	µA	$0.45 \le V_{OUT} \le V_{CC}$
I_{CC}	Power Supply Current from V_{CC}		200	mA	

A.C. CHARACTERISTICS

D7225-1 $T_A = 0°C$ to $+70°C$, $V_{CC} = 5.0V +5\%, -10\%$; $C_L = 150$ pF; unless otherwise noted
D7225-6 $T_A = -40°C$ to $+85°C$, $V_{CC} = 5.0V +5\%, -10\%$; $C_L = 150$ pF; unless otherwise noted

Symbol	Parameter	Min	Max	Units	Test Condition
t_P	Clock Period	249.75	250.25	ns	
$t_Ø$	Clock Phase Width (High Time)	0.45 t_P	0.55 t_P		
t_r–t_F	Input Signal Rise and Fall Time		30	ns	Not Tested

Read Cycle (Host Interface)

Symbol	Parameter	Min	Max	Units	Test Condition
t_{AC}	Select Setup to $\overline{RD}$ ↓	0		ns	
t_{CA}	Select Hold from $\overline{RD}$ ↑	0		ns	
t_{RR}	$\overline{RD}$ Pulse Width	200		ns	
t_{AD}	Data Delay from Address		150	ns	
t_{RD}	Data Delay from $\overline{RD}$ ↓		150	ns	
t_{DF}	Output Hold Time	10	100	ns	Under Pin Loads*
	Output Float Time		150		
t_{DC}	DACK Setup to $\overline{RD}$ ↓	0		ns	
t_{CD}	DACK Hold from $\overline{RD}$ ↑	0		ns	
t_{KD}	Data Delay from $\overline{DACK}$ ↓		150	ns	
t_{CYCR}	"Read" Cycle Time	(DMA Mode) $4t_P - t_{RR}$			In non DMA mode t_{CYCR} Min. = 6 t_P
t_{DEADR}	Inactive Time between $\overline{RD}$ ↑ and $\overline{RD}$ ↓	150		ns	Not Tested

A.C. CHARACTERISTICS (Continued)

D7225-1 $T_A = 0°C$ to $+70°C$, $V_{CC} = 5.0V +5\%, -10\%$; $C_L = 150$ pF; unless otherwise noted

D7225-6 $T_A = -40°C$ to $+85°C$, $V_{CC} = 5.0V +5\%, -10\%$; $C_L = 150$ pF; unless otherwise noted

Write Cycle (Host Interface)

Symbol	Parameter	Min	Max	Units	Test Condition
t_{AC}	Select Setup to $\overline{WR}\downarrow$	0		ns	
t_{CA}	Select Hold from $\overline{WR}\uparrow$	0		ns	
t_{WW}	$\overline{WR}$ Pulse Width	200		ns	
t_{DW}	Data Setup to $\overline{WR}\uparrow$	200		ns	
t_{WD}	Data Hold from $\overline{WR}\uparrow$	0		ns	
t_{DC}	$\overline{DACK}$ Setup to $\overline{WR}\downarrow$	0		ns	
t_{CD}	$\overline{DACK}$ Hold from $\overline{WR}\uparrow$	0		ns	
t_{CYCW}	"Write" Cycle Time	$4\,t_P + t_{WW}$			Not Tested
t_{DEADW}	Inactive Time between $\overline{WR}\uparrow$ and $\overline{WR}\downarrow$	$4\,t_P$		ns	Not Tested
t_{CQ}	Request Hold from $\overline{RD}$ or $\overline{WR}$ (Non-Burst Mode)		150	ns	$\overline{WR}$ Hold Time Not Tested

FSA INTERFACE TIMINGS (under pin loading)

Symbol	Parameter	Min	Max	Units	Test Condition
t_{CDV}	CLK to DIO Valid Delay		150	ns	Under Pin Loads*
t_{CDF}	CLK to DIO Entering Float	10	250	ns	Under Pin Loads*
t_{CDE}	CLK to DIO Enabled from Float		150	ns	Under Pin Loads*
t_{CDH}	CLK to DIO Hold Time	0	140	ns	Under Pin Loads*
t_{CSOL}	CLK to $\overline{SYNC}$ Leading Edge Delay		100	ns	Under Pin Loads*
t_{CSOT}	CLK to $\overline{SYNC}$ Trailing Edge Delay	10	100	ns	Under Pin Loads*
t_{DC}	DIO Setup Time to Clock	80		ns	Under Pin Loads*
t_{DHC}	DIO Hold Time from Clock	0		ns	Under Pin Loads*
t_{COL}	CLK to Output Leading Edge		150	ns	Under Pin Loads*
t_{COT}	CLK to Output Trailing Edge	0	190	ns	Under Pin Loads*
t_{EW}	$\overline{ERR.FLG}$ Pulse Width	200		ns	Under Pin Loads*
t_{SCFT}	$\overline{SHIFTCLK}$ to CLK		125	ns	Under Pin Loads*

7250–7234 Interface Timings

Symbol	Parameter	Min	Max	Units	Test Condition
t_{CBL}	CLK to Bubble Signal Leading Edge		250	ns	Under Pin Loads*
t_{CBT}	CLK to Bubble Signal Trailing Edge	0	250	ns	Under Pin Loads*

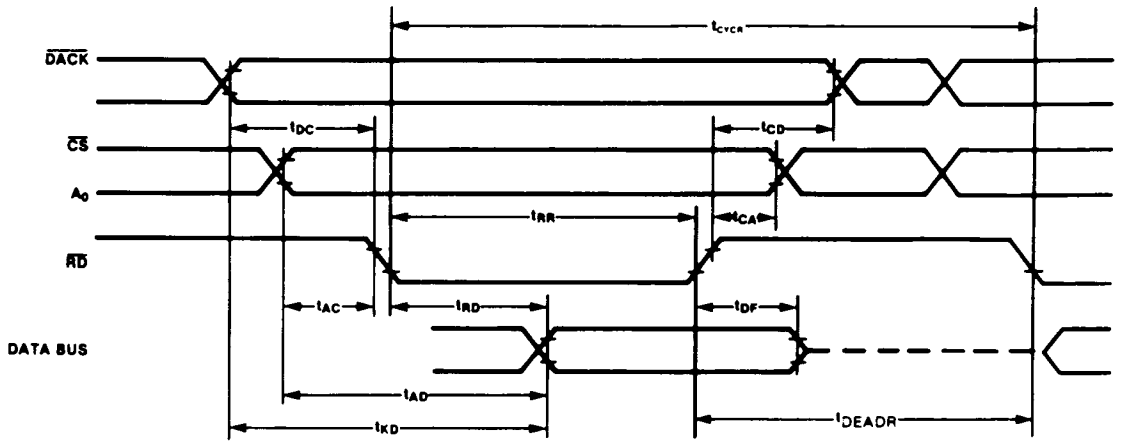
*Bubble Pin Loads Shown Below

Pin Loadings

Pin Names	Value	Units
$\overline{X+}$, $\overline{X-}$, $\overline{Y+}$, $\overline{Y-}$	200	pF
$\overline{TM.A}$, $\overline{TM.B}$, $\overline{REP.EN}$, $\overline{BOOT.EN}$, $\overline{SWAP.EN}$, $\overline{BOOT.SW.EN}$, $\overline{C/D}$, $\overline{ERR.FLG}$, $\overline{WAIT}$, $\overline{SYNC}$, DIO	150	pF
$\overline{DET.ON}$ & $\overline{SHIFT.CLK}$	100	pF
$\overline{BUS.READ}$	60	pF

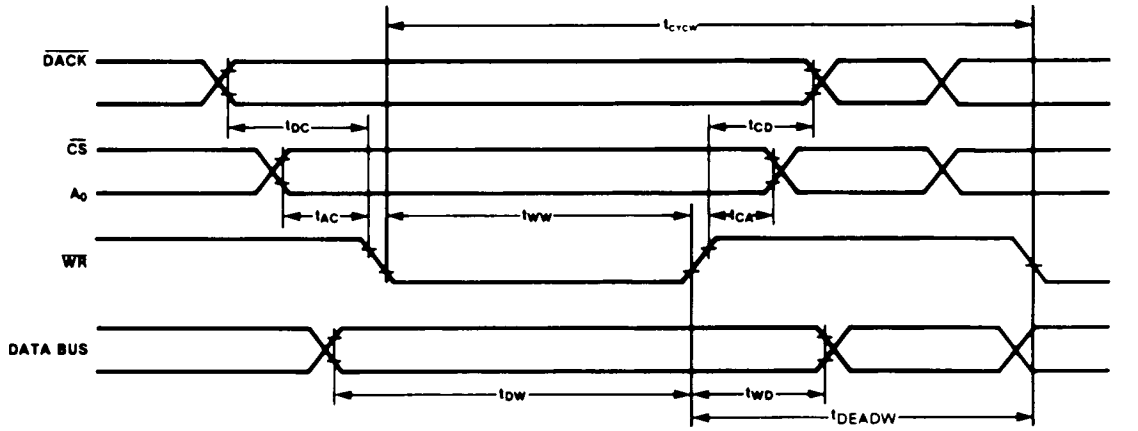
WAVEFORMS

Read (Host Interface)



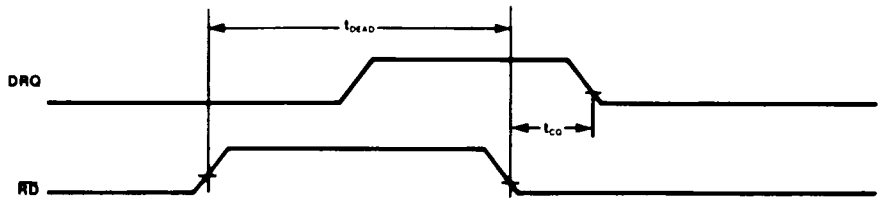
0004-8

Write (Host Interface)



0004-9

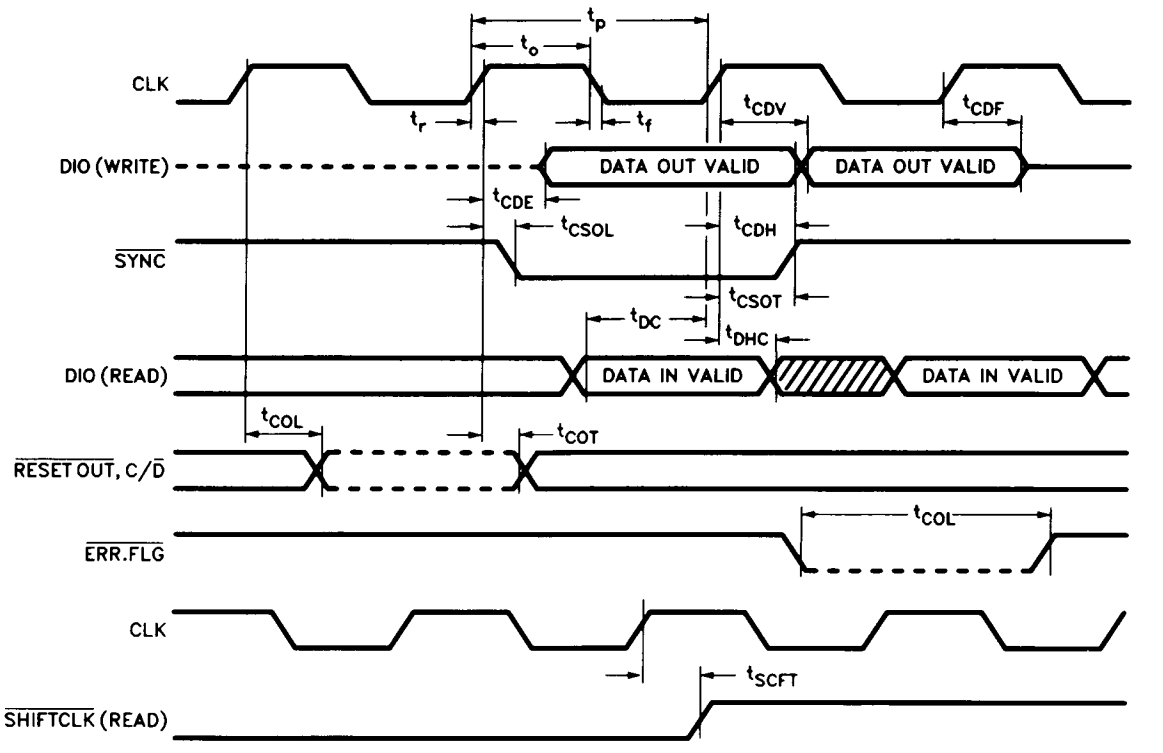
DMA (Host Interface)



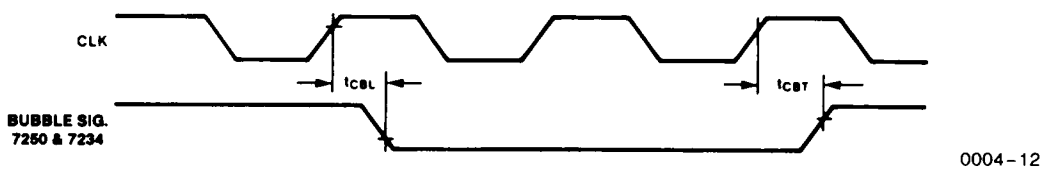
0004-10

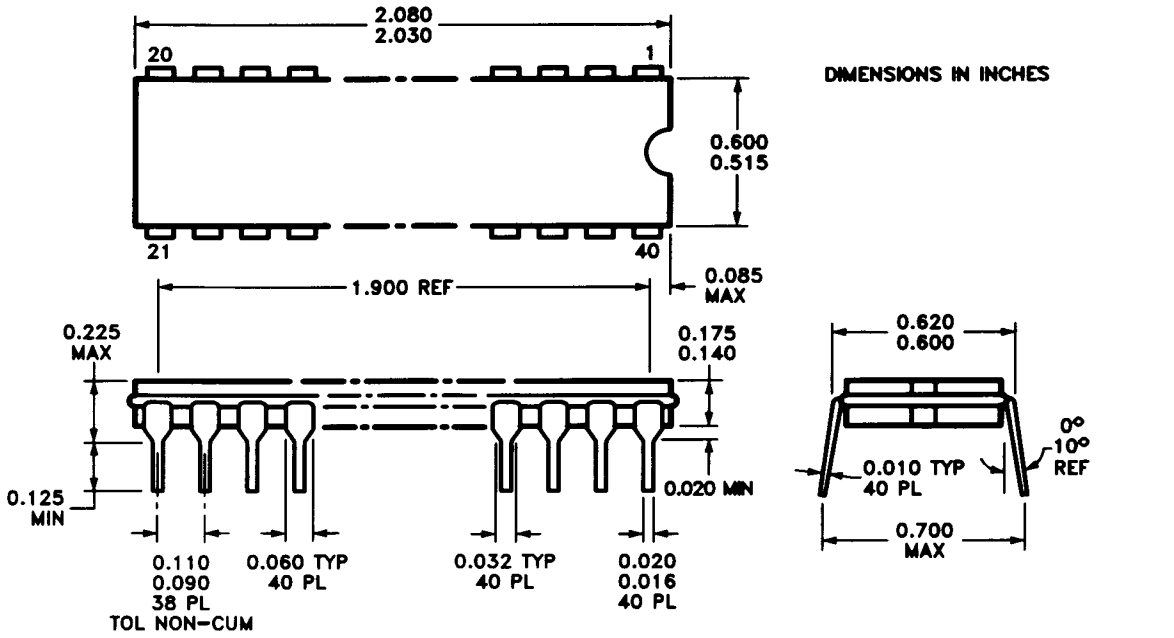
WAVEFORMS (Continued)

7245 Interface Timings



7250 & 7234 Interface Timings





40-Lead Dual-In-Line Package

MEMTECH TECHNOLOGY CORPORATION
101 BLUE RAVINE ROAD
FOLSOM, CA 95630

MARKETING, CUSTOMER SERVICE: (408) 980-5300
TECHNICAL HOTLINE: (916) 351-3132
FAX: (408) 980-5383
TELEX: 910 250 1368 MEMTECH UQ
EASYLINK: 62020974

Printed in U.S.A/1187/5K/RJ/DT
20