7242
DUAL FORMATTER/SENSE AMPLIFIER FOR BUBBLE MEMORIES

7242	0 to 70°C
7242-5	−20 to +85°C

- **Error Detection/Correction Done Automatically**
- **Dual Channel**
- **On-Chip Sense Amplifiers**
- **Automatically Handles Redundant Loops**
- **FIFO Data Block Buffer**
- **Daisy-Chained Selects for Multiple Bubble Memory Systems**
- **MOS N-Channel Technology**
- **Standard 20-Pin Dual-In-Line Package**

The Intel 7242 is a Dual Formatter/Sense Amplifier (FSA) designed to interface directly with Intel Magnetics Bubble Memories. The 7242 features on-chip sense amplifier for system ease of use and minimization of system part count. The 7242 also provides for automatically handling the bubble memories' redundant loops so they are transparent to the user. In addition, complete burst error detection and correction can be done automatically by this device.

The 7242 has a full FIFO data block buffer. This device can be daisy-chained for multiple bubble memory systems. Up to eight FSAs can be controlled by one 7220-1 Bubble Memory Controller (BMC).

The 7242 utilizes an advanced NMOS technology to incorporate the on-chip sense amplifiers and other unique features. The device is mounted in a standard high-density 20-pin dual-in-line package.

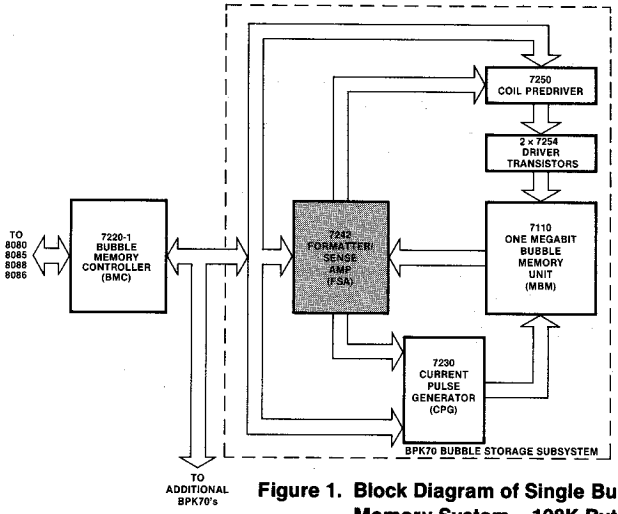


Figure 1. Block Diagram of Single Bubble Memory System—128K Bytes

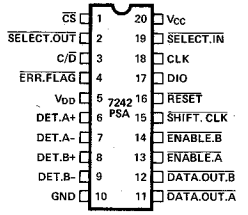


Figure 2. Pin Configuration

NOVEMBER 1982
ORDER NUMBER: 210799-001

Table 1. Pin Description

Symbol	Pin No.	Description
C/$\overline{D}$	3	Command/$\overline{Data}$ signal. This signal shall cause the FSA to enter a receive command mode when high and to interpret the serial data line as data when low. Any previously active command will be immediately terminated by C/$\overline{D}$.
CLK	18	Same TTL-level clock used to generate internal timing as used for 7220-1.
$\overline{CS}$	1	An active low signal used for multiplexing of FSAs. The FSA is disabled whenever $\overline{CS}$ is high (i.e., it presents a high impedance to the bus and ignores all bus activity).
$\overline{DATA.OUT.A}$, $\overline{DATA.OUT.B}$	11, 12	Output data from the FIFO to the MBM generate circuitry. Used to write data into the bubble device (active low).
DET.A+, DET.A−, DET.B+, DET.B−	6, 7, 8, 9	Differential signal lines from the MBM detector.
DIO	17	The Serial Bus data line (a bidirectional active high signal).
$\overline{ENABLE.A}$, $\overline{ENABLE.B}$	13, 14	TTL-level outputs utilized as chip selects for other interface circuits. They shall be set and reset by the Command Decoder under instruction of the Controller (active low).
$\overline{ERR.FLG}$	4	An error flag used to interrupt the Controller to indicate that an error condition exists. It shall be an open drain, active low signal.
$\overline{RESET}$	16	An active low signal that shall reset all flags and pointers in the FSA as well as disabling the chip as the CS signal does. The $\overline{RESET}$ pulse width must be 5 clock periods to assure the FSA is properly reset.
$\overline{SELECT.IN}$	19	An input utilized for time-division multiplexing. An active low signal whose presence indicates that the FSA is to send or receive data from the Serial Bus during the next two clock periods.
$\overline{SELECT.OUT}$	2	The $\overline{SELECT.IN}$ pulse delayed by two clocks. It shall be connected to the $\overline{SELECT.IN}$ pin of the next FSA. It is delayed by two clocks because the FSA is a dual-channel device. Channel A shall internally pass $\overline{SELECT.IN}$ to Channel B (delayed by one clock).
$\overline{SHIFT.CLK}$	15	A Controller-generated clock signal that shall be used to clock data out of the bubble I/O Output Latch to the bubble module during a write operation and to cause bubble signals to be converted by the Sense Amp and clocked into the Bubble I/O Input Latch on a read.

FUNCTIONAL DESCRIPTION

The following is a brief description of each block of the 7242 FSA,

Serial Communications—The Serial Communications block handles all transfers on the Serial Bus and is shared by both channels of FSA.

Command Decoder—The Command Decoder interprets commands by the Serial Communication logic and sets the appropriate command and enable lines. It also maintains FSA status, and generates various reset lines.

Internal Data Bus—The Internal Data Bus is the main data link between the Serial Communications block and all other data sources in each half of the FSA.

I/O Latches, Flags, and Bus Control—Each channel of the FSA has its own internal Data Bus, on which all data transfers are made. There is a Flag and a bidirectional Latch in each "I/O Latches–Flag" block. Only one Latch is used in a given operation and the Flag tells the Bus Controller whether or not the Latch is full. The Bus Controller monitors these flags, and other control signals, to determine when each device should have access to the internal Data Bus. When a transfer is to be made, the appropriate devices are enabled, the Bus is enabled, and the transfer takes place synchronously by virtue of a transparent State Machine Sequencer.

FIFO—The FIFO is a variable-length First-In-First-Out buffer utilized to store data passing to and from the MBM module. The FIFO is logically 272 bits in length in the "no error correction" mode. It is 270 bits in the "error correction" mode, since 256 bits of the

data and a 14-bit error correction code must be used in this mode of operation.

The FIFO pointers are reset by hardware or software resets or each time a command to read or write is received by the Command Decoder.

If a block length other than 272 bits is used in the no error correction mode, the FIFO pointers will not return to word zero at the end of each block transfer. This is of no consequence if one is not concerned about the absolute location of data in the FIFO. Keeping in mind that the FIFO is only 272 bits physically, any block length may be used up to and including 320.

Bootstrap Loop Register—The Bootstrap Loop Register is a 160-bit register that contains information detailing the location of bad loops in the MBM module. This data will enable bubble I/O to ensure that bits are not loaded in the FIFO from bad loops, or written from the FIFO into bad loops. A logic zero (absence of a bubble) is written into bad loops.

Error Correction Logic—The Error Correction Logic contains the circuitry to implement a burst error correcting code capable of correcting any single burst error of length equal to or less than 5, anywhere in the 270-bit data stream, including the error correction code which is 14 bits in length. A Correction Enable bit may be set or reset via a special command. When reset, the entire error correction network is disabled and block length may vary from 270 bits. Error detection shall be accomplished on all data transfers (when enabled); however, correction cannot take place unless the FSA is operated in a buffered mode (i.e., an entire block is read prior to passing any data to the Controller).

Bubble I/O—The Bubble I/O consists of an integrated Sense Amplifier and an output driver. The Sense Amplifier consists of a sample-and-hold circuit and a differential, chopper-stabilized comparator.

Enables—The $\overline{\text{ENABLE.A}}$ and $\overline{\text{ENABLE.B}}$ outputs are utilized as chip selects for external circuitry. To set an ENABLE line, the desired channel of the FSA must be selected and Read or Write MBM, Set Enable Bit, Initialize, Read Corrected Data, or Internally Correct Data command is sent. Any other command sequence will reset the ENABLE lines.

COMMANDS

FSA Commands

The FSA shall receive a four-bit command word via the Serial Bus. In addition, some of the commands require additional data bits, e.g., status to be passed serially. The four bits shall be interpreted as shown in Table 2. The effects on the Status bits, Correction Enable bit, and Enable pins are summarized in Table 3.

The following is a brief description of each command available in the 7242 FSA.

No Operation—Deselects the chip and prevents further internal activity (default state for reset, unselected or unaddressed channels). Resets the FIFO and Bootloop pointers. The Enable pins ($\overline{\text{ENABLE.A}}$ and $\overline{\text{ENABLE.B}}$) become inactive.

Software Reset—Resets all FIFO and Bootloop pointers and flags. Status flags, Error Correction Enable bit, error correction shift register, and the Enable pins become inactive.

Initialize—The chip is set to read data from the MBM Bootloop and pass it to the Controller. Resets the FIFO and Bootloop pointers and the Error Correction Logic, and disables the Bootloop register (so that it does not interfere with the data flow). The Enable pins become active in addressed channels.

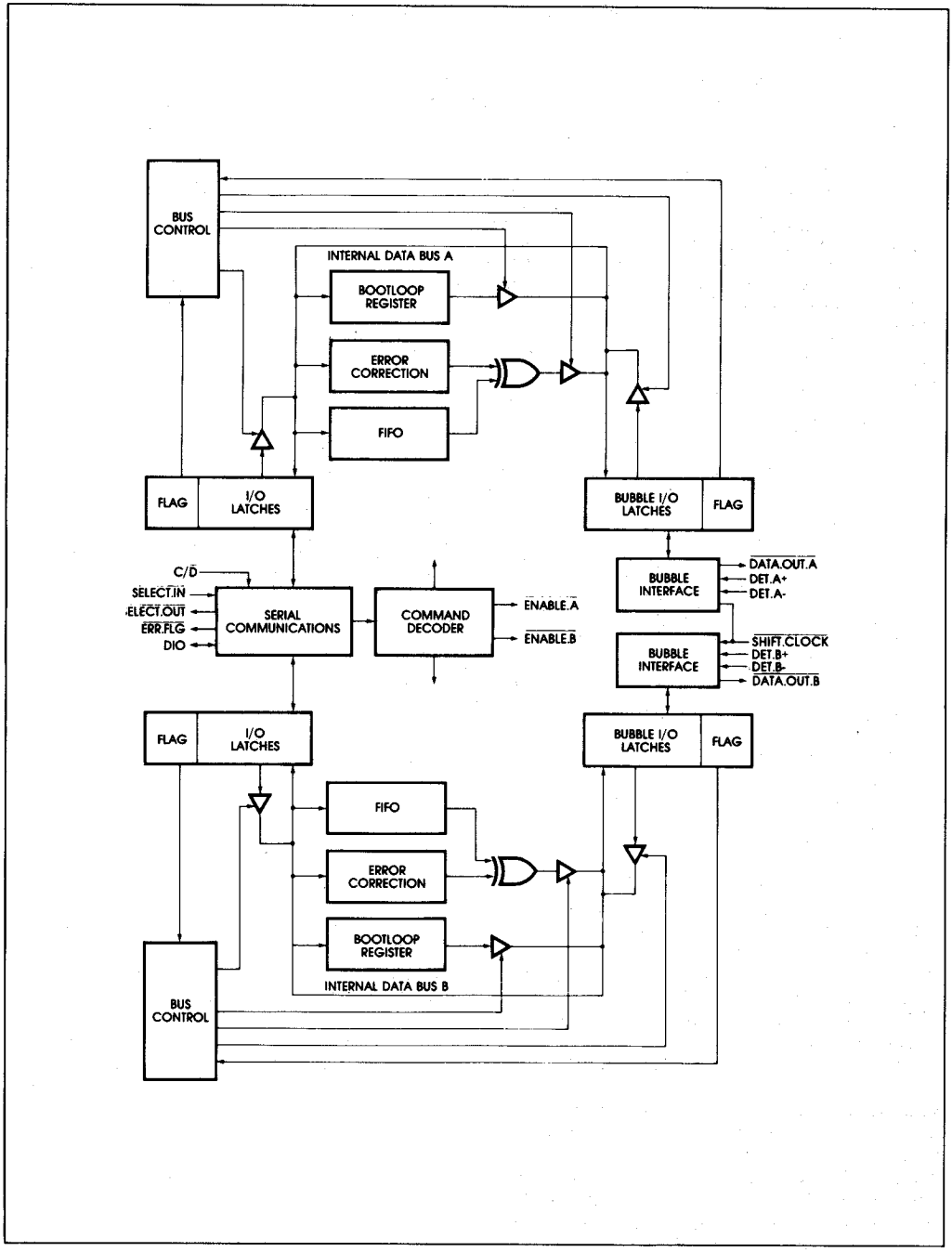


Figure 3. Logic Diagram

AFN-01358B

Table 2. Command Code Description

Code	Description	Data	
		Correction Enabled	Not Enabled
0000	No Operation	None	None
0001	(Reserved)	—	—
0010	Software Reset	None	None
0011	Initialize	MBM Bootloop	MBM Bootloop
0100	Write MBM Data	270 Bits In	Variable
0101	Read MBM Data	270 Bits Out	Variable
0110	Internally Correct Data	None	—
0111	Read Corrected Data	270 Bits Out	—
1000	Write Bootloop Register	160 Bits In	160 Bits In
1001	Read Bootloop Register	160 Bits Out	160 Bits Out
1010	(Reserved)	—	—
1011	(Reserved)	—	—
1100	Set Enable Bit	None	None
1101	Read ERR.FLG Status	1 Bit Out	1 Bit Out
1110	Set Correction Enable Bit	None	None
1111	Read Status Register	8 Bits Out	8 Bits Out

Table 3. Command Function Summary

Command Description	Command Code	Data Flow (R/W)	Reset FIFO & Bootloop Pointers	Reset Status (Errors)	Reset Error Correction Logic	Enable
No Operation	0000	—	X			H
Software Reset	0010	—	X	X	X	H
Initialize	0011	R	X	X	X	L
Write MBM Data	0100	W	X		X	L
Read MBM Data	0101	R	X		X	L
Internally Correct Data	0110	—	X		—	L
Read Corrected Data	0111	R	X		—	L
Write Bootloop Register	1000	W	X		—	H
Read Bootloop Register	1001	R	X			H
Set Enable Bit	1100	—	X			L
Read ERR.FLG Status	1101	R				H
Set Error Correction Enable Bit	1110	—	X			H
Read Status Register	1111	R		X		H

Write MBM Data—Data input by the Controller is written into the good loops in use in the MBM (under control of the Bootloop register) each time a SHIFT.CLK is received. It also activates the Enable pins and resets the FIFO and Bootloop pointers. If the Correction Enable bit is set, the FSA computes the correction code and appends it to the data stream to be stored in the MBM (last 14 of 270 bits).

Read MBM Data—This command activates the ENABLE pins and resets the FIFO and Bootloop pointers independent of the state of the Correction Enable bit. If the Correction Enable bit is reset, data from the MBM, of block length dictated by 2 times the number of logic "1s" in the Bootloop register, is sensed and screened by the FSA Sense Amp and Bootloop register, and stored in the FIFO. As soon as one bit is guaranteed in the FIFO, simultaneous reading from the FIFO may be done by the Controller. The FIFO need not be emptied after each page is read, but one must insure that more than 272 bits of FIFO are not needed at any time during the transfer.

If the Correction Enable is set, data must be read in a buffered mode. First, a full block of data is read from the MBM . At that point the FIFO contains 270 bits of data. If an error is detected by the Error Correction network, the FSA raises the UNCORR.ERR and CORR.ERR flags which generate an interrupt to the Controller. If no error is detected, the 270 bits of data may be read from the FIFO while simultaneously reading and checking the next block of data from the MBM. When an error is detected the Controller may respond to the interrupt in one of three ways.

1. Ignore it and try again (must make sure to reset the Error Correction shift register before a retry).

2. Send a Read Corrected Data command to the FSA. This command will correct the data stream (if possible) and interrupt the Controller when the block has been read. At this time the Controller can send a Read Status command to see if the error was correctable (CORR.ERR) or uncorrectable (UNCORR.ERR).

3. Send an internally Correct Data command to the FSA. The FSA corrects the data without transferring it to the Controller. When finished, the FSA interrupts the Controller. At this point it can be determined whether or not the error is correctable. If so, a Read Corrected Data command may be sent to read the good data.

Internally Correct Data—Internally cycles the data through the error correction network and returns status as to whether or not the data is correctable.

Requires approximately 1400 clock cycles to complete. $\overline{\text{ERR.FLG}}$ will be inactive during internal cycling, but will return active at its completion. Also activates the ENABLE pins and resets the FIFO and Bootloop pointers.

Read Corrected Data—Cycles data through the error correction network with each Controller read (SELECT.IN at the FSA). At the end of 270 reads, status is available to indicate whether or not the data was successfully corrected. $\overline{\text{ERR.FLG}}$ acts as in Internally Correct Data. This command is required to read data corrected internally as well, but has no effect on the data read if it was successfully corrected. Activates the ENABLE pins and resets the FIFO and Bootloop pointers.

Write Bootloop Register—Contents of the FSA's Bootloop register are written with 160 bits from the Controller. The Controller must read the MBM Bootloop first, to determine which loops are good. The number of good bits in the 160-bit register is 135 if correction is used, and variable up to 160 if operating in the no correction mode. ENABLE pins become inactive and the FIFO and Bootloop pointers are reset.

Read Bootloop Register—As above except that data is read from the FSA Bootloop to the Controller.

Set Enable Bit—ENABLE pins become active for addressed channels, inactive for unaddressed channels. Also resets the FIFO and Bootloop pointers.

Read $\overline{\text{ERR.FLG}}$ Status—Reads the composite error status for addressed channels of the FSA. (The composite status is the logic OR of CORR.ERR, UNCORR.ERR and TIMER.R. The $\overline{\text{ERR.FLG}}$ pin is the logic NOR of both channels' composite error status: ERR.FLG.A and ERR.FLG.B.) ENABLE pins become inactive.

Set Error Correction Enable Bit—Enables the Error Correction Logic in addressed FSAs and disables it in unaddressed FSAs. ENABLE pins become inactive and FIFO and Bootloop pointers are reset. Furthermore, when this enable is set, the corresponding FIFO becomes a 270-bit FIFO (logically) instead of a 272-bit FIFO as in the no correction mode.

Read Status Register—The 8-bit Status Word for the addressed FSA is output to the Controller. Only one FSA channel can be addressed at a time, or bus contention may result. ENABLE pins become inactive and error flags in the addressed FSA channel are reset.

SERIAL INTERFACE

Command Sequence—The FSA communicates with the Controller via a Serial Interface. The Controller/FSA Interface contains the following signals:

1. CLK
2. $\overline{\text{SELECT.IN}}$ (Formatter)
3. $\overline{\text{SELECT.OUT}}$ (Formatter)
4. $\overline{\text{SYNC}}$ (Controller)
5. DIO
6. C/$\overline{\text{D}}$
7. $\overline{\text{SHIFT.CLK}}$
8. $\overline{\text{ERR.FLG}}$

Commands from the Controller to the FSA shall take place in the following format (see Figure 4).

1. Controller raises C/$\overline{\text{D}}$ flag indicating that a command is coming, and simultaneously outputs a $\overline{\text{SYNC}}$ pulse. This $\overline{\text{SYNC}}$ pulse is shifted down the FSA chain in shift register fashion via the FSA $\overline{\text{SELECT.IN}}$/$\overline{\text{SELECT.OUT}}$ lines.

2. Controller outputs a serial data stream on the DIO line beginning in the clock period following SYNC. Each bit in the stream corresponds to an address bit for a particular FSA (up to 16 channels). Each FSA, upon receiving $\overline{\text{SELECT.IN}}$ will look for the presence or absence of a logic one on

DIO in the clock period following receipt of $\overline{\text{SELECT.IN}}$. (A logic one indicates that the FSA shall accept the command.)

3. Twenty clock periods after the first SYNC the Controller sends C/$\overline{\text{D}}$ low followed by a four-bit command on the DIO line.

4. If the command is a Read Status command (1111), the addressed FSA returns 8 bits of Status starting 4 clock periods after the last command bit is received. Note that the Status is returned during this period for any FSA position. Therefore only one FSA channel should be addressed at a time to avoid contention.

5. If the command requires further data (see section on FSA Commands), more $\overline{\text{SYNC}}$ pulses are sent by the Controller. This will occur at integral multiples of 80 or 20 clock periods starting no sooner than 40 clocks after the first command $\overline{\text{SYNC}}$ pulse. Some number of $\overline{\text{SYNC}}$ periods may pass before the second $\overline{\text{SYNC}}$ to allow the FSA to set itself up and get data ready for the Controller. There are several possibilities:

   a. For the Read $\overline{\text{ERR.FLG}}$ Status command the second $\overline{\text{SYNC}}$ can occur 40 clocks after the first $\overline{\text{SYNC}}$. This $\overline{\text{SYNC}}$ (or $\overline{\text{SELECT.IN}}$) causes each addressed FSA to send the appropriate Status Information. No further $\overline{\text{SYNCs}}$ (without C/$\overline{\text{D}}$ high) should be sent.

   b. For the Read MBM Data (or initialize) command the second $\overline{\text{SYNC}}$ must wait the appropriate number of $\overline{\text{SHIFT.CLOCKs}}$ to assure that valid data is available in the FIFO. After this wait, each addressed FSA channel sends one bit of data on the DIO line for each $\overline{\text{SYNC}}$ (or $\overline{\text{SELECT.IN}}$) pulse.

   c. For the Read Bootloop Register command, the second $\overline{\text{SYNC}}$ can occur 60 clock cycles after the first $\overline{\text{SYNC}}$. The data transfer then proceeds as in b. above.

   d. For the Write MBM Data or Write Bootloop commands, the DIO line is used to transfer data to the FSA on successive SYNC pulses. The first data bit can be transferred by a second $\overline{\text{SYNC}}$ pulse, 40 clock cycles after the first $\overline{\text{SYNC}}$. (However, data to the MBM will not be available at the Dataout pins until 40 clock cycles after the $\overline{\text{SYNC}}$ which transferred it.) Each transfer to the addressed FSA will be initiated by a $\overline{\text{SYNC}}$ (or $\overline{\text{SELECT.IN}}$).

6. $\overline{\text{SYNC}}$ ($\overline{\text{SELECT.IN}}$) precedes the data it transfers by 1 clock cycle. Data Transfers to or from the FSA's FIFO must contain the proper number of $\overline{\text{SYNCs}}$ (externally counted) or a timing error may occur ($\overline{\text{TIMERR}}$ flag will be set, causing an interrupt to the Controller).

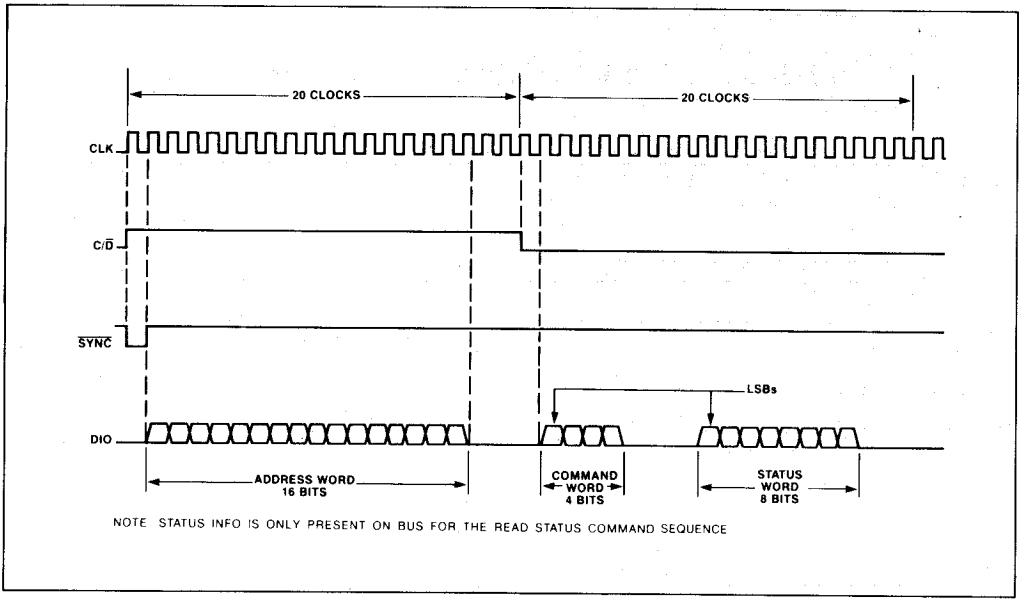


Figure 4. Command Sequences

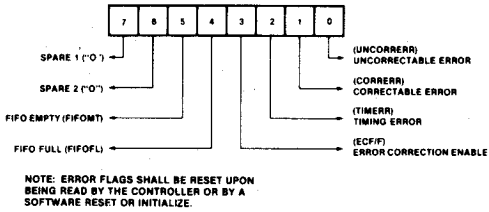
Data Sequences—Bubble data shall be passed between the Controller and FSAs in the following fashion (see Figure 5).

1. Controller outputs a $\overline{\text{SYNC}}$ pulse.
2. Each FSA then outputs (inputs) a single bit on DIO after $\overline{\text{SYNC}}$ ($\overline{\text{SELECT.IN}}$) has been clocked into its control section. Only previously enabled FSAs output (input) data and the Controller must know when to input (output) data bits.
3. After 80 or 20 clocks, another $\overline{\text{SYNC}}$ pulse is output and the sequence repeats until all data has been transferred.

Error Conditions—Each FSA shall upon detection of an error set a Status bit and pull down $\overline{\text{ERR.FLG}}$. This signal can be asynchronous to $\overline{\text{SYNC}}$. Error Status bits shall be:

1. Correctable Error
2. Uncorrectable Error
3. Timing Error

The Status Word that shall be passed to the Controller after receipt of a Read Status command shall be in the following format:



NOTE: ERROR FLAGS SHALL BE RESET UPON BEING READ BY THE CONTROLLER OR BY A SOFTWARE RESET OR INITIALIZE.

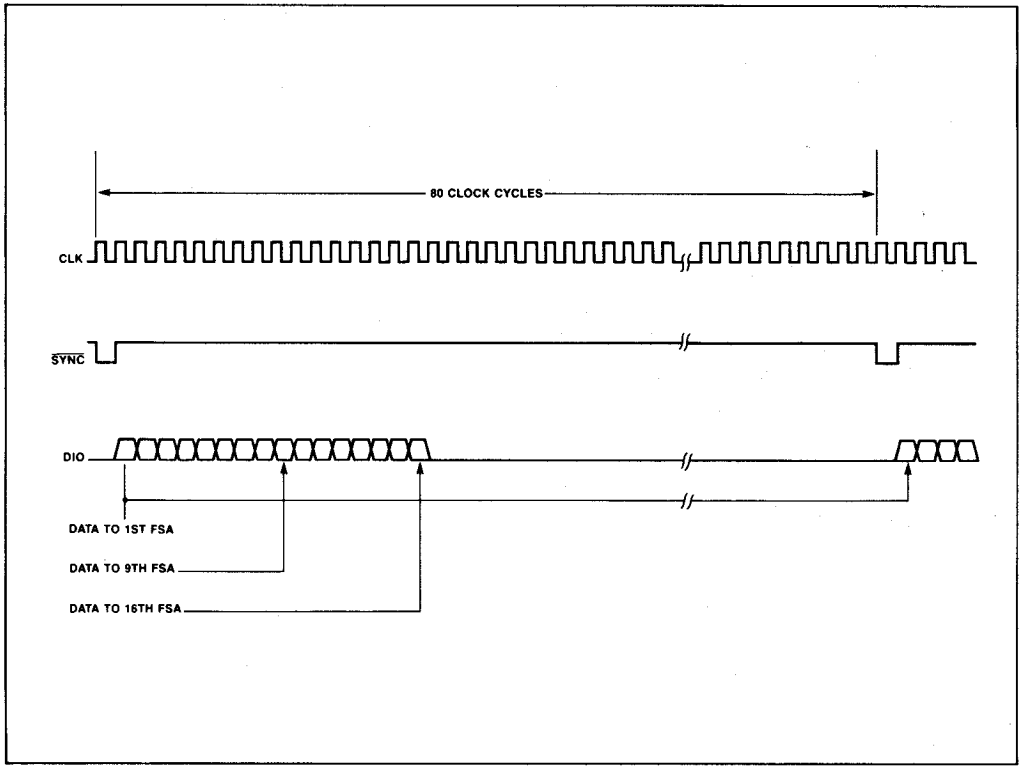


Figure 5. Data Sequences

BUBBLE INTERFACE

Bubble Interface—Each Bubble Interface shall consist of a DATAOUT signal and a pair of differential inputs from the MBM detector bridge.

Read Timing—The timing for reading a bit from the memory shall be as follows:

1. Controller outputs a $\overline{\text{SHIFT.CLK}}$. FSA samples bubble signal during $\overline{\text{SHIFT.CLK}}$ and holds signal after trailing edge.
2. Trailing edge of $\overline{\text{SHIFT.CLK}}$ initiates signal conversion timing.
3. Data is latched at end of conversion period in the Bubble input latch, and will subsequently be loaded into the FIFO.

Write Timing—The timing for writing a bit from the FIFO shall be as follows:

1. Controller lowers $\overline{\text{SHIFT.CLK}}$.
2. Data is gated out of FSA by $\overline{\text{SHIFT.CLK}}$.
3. Controller outputs a generate pulse (to external logic, not to FSA).
4. Controller raises $\overline{\text{SHIFT.CLK}}$. The $\overline{\text{DATA.OUT}}$ pin is forced high.
5. FIFO and Bootloop register are incremented after the leading edge of $\overline{\text{SHIFT.CLK}}$.

System Timing—The $\overline{\text{SYNC}}$ pulse (which denotes the beginning of a data transfer from Controller to Formatter or vice-versa) shall be synchronous with the beginning of a bubble memory field rotation. Due to timing constraints in the FSA, the following statements hold:

1. Data read from the bubble memory into the FSA shall not be available to the Controller until 40 clock cycles after $\overline{\text{SHIFT.CLK}}$.
2. Data cannot be written to the bubble memory until 40 clock cycles after $\overline{\text{SYNC}}$.

FSA ERRROR CORRECTION

Error Correction—The error correction logic consists of a burst error correcting File code capable of correcting 5 or fewer bits in a single burst; the number of check bits is 14.* Error correction/detection shall take place on each 256-bit data block. The FSA shall set low $\overline{\text{ERR.FLG}}$ each time a correctable or uncorrectable error is detected. $\overline{\text{ERR.FLG}}$ shall be set high upon being read by the Controller or by a software reset being issued. The polynomial implemented is given below:

$$G(X) = 1 + X^2 + X^5 + X^9 + X^{11} + X^{14}$$

DATA FORMAT

Data Format—Data into a single FSA channel from the bubble memory shall be in the format described below. The two channels of the bubble are represented identically. The following definitions apply:

o_η = data from odd quads of bubble device, loop η
e_η = data from even quads of bubble device, loop η

Data Block Format—

$o_1e_1o_1e_1o_2e_2o_2e_2 \ldots o_{80}e_{80}o_{80}e_{80}$

1st bit 320th bit

When using correction, the first 270 good bits will be used; the last 14 of these are to be used for the error correcting code. The remaining 50 bits must be masked as "bad" bits in the FSA Bootloop register.

When operating without correction, any number of bits may be used by loading the Bootloop register appropriately. The preferred number is 272 bits, however.

*See "Error-Correcting Codes" by W.W. Peterson and E.J. Weldon, Jr., pp. 366-370, M.I.T. Press, 1972.

ABSOLUTE MAXIMUM RATINGS*

Temperature Under Bias ..... −40°C to +100°C
Storage Temperature ..... −65°C to +150°C
All Input or Output Voltages and
V_{CC} Supply Voltage ..... −0.5V to +7V
V_{DD} Supply Voltage ..... −0.5V to +14V

COMMENT: 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 (T_A = 0°C to 70°C; V_{CC} = 5.0V +5%, −10%; V_{DD} = 12V ±5%)

Symbol	Parameter	Limits			Unit	Test Conditions
		Min.	Typ.	Max.		
V_{IL}	Input Low Voltage	−0.5		0.8	V	
V_{IH}	Input High Voltage	2.0*		V_{CC} +0.5	V	
V_{OL}	Output Low Voltage (All Outputs Except $\overline{\text{SELECT.OUT}}$)		.2	0.45	V	I_{OL} = 3.2mA
V_{OLSO}	Output Low Voltage ($\overline{\text{SELECT.OUT}}$)		.2	0.45	V	I_{OL} = 1.6mA
V_{OH}	Output High Voltage (All Outputs Except $\overline{\text{SELECT.OUT}}$)	2.4	3.0		V	I_{OH} = 400 μA
V_{OHSO}	Output High Voltage ($\overline{\text{SELECT.OUT}}$)	2.4			V	I_{OH} = 200 μA
V_{THR}	Detector Threshold	2.3	2.5	2.7	mV	V_{DD} = 12.0V
$\lvert I_{IL} \rvert$	Input Leakage Current		0	5	μA	$0 \leq V_{IN} \leq V_{CC}$
$\lvert I_{OFL} \rvert$	Output Float Leakage		0	10	μA	$0.45 \leq V_{OUT} \leq V_{CC}$
I_{CC}	Power Supply Current from V_{CC}		35	120	mA	
I_{DD}	Power Supply Current from V_{DD}		5	30	mA	

*Minimum V_{IH} is 2.2V for the 7242-5 device.

A.C. CHARACTERISTICS

(T_A = 0°C to +70°C; V_{CC} = 5.0V +5%, −10%; V_{DD} = 12V ±5%; C_L = 120 pF; unless otherwise noted)

Symbol	Parameter	Min.	Max.	Unit	Test Conditions
t_p	Clock Period	240	500	ns	
t_f	Clock Phase Width	.45 t_p	.55 t_p		
t_n t_f	Clock Rise and Fail Time		30	ns	
t_{SIC}	$\overline{\text{SELECT.IN}}$ Setup Time to CLK	50		ns	
t_{CDC}	C/$\overline{\text{D}}$ Setup Time to CLK	50		ns	
t_{CYC}	$\overline{\text{SELECT.IN}}$ or $\overline{\text{SHIFT.CLK}}$ Cycle Time	20 t_p			
t_{DC}	DIO Setup Time to Clock (Read Mode)	50		ns	
t_{CSC}	$\overline{\text{CS}}$ Setup Time to CLK	100		ns	
t_{RIC}	$\overline{\text{RESET.IN}}$ Setup Time to CLK	100		ns	
t_{IH}	Control Input Hold Time for C/D, $\overline{\text{SELECT.IN}}$ and DIO	10		ns	
t_{CSOL}	CLK to $\overline{\text{SELECT.OUT}}$ Leading Edge Delay		100	ns	C_L = 50 pF
t_{CSOT}	CLK to $\overline{\text{SELECT.OUT}}$ Trailing Edge Delay		80	ns	C_L = 50 pF
t_{CDV}	CLK to DIO Valid Delay*		100	ns	
t_{CDH}	CLK to DIO Hold Time*	0		ns	
t_{CDE}	CLK to DIO Enabled from Float*		100	ns	
t_{SIDE}	$\overline{\text{SELECT.IN}}$ Trailing Edge to DIO Enabled from Float*		70	ns	
t_{CDF}	CLK to DIO Entering Float*		100	ns	
t_{SCDO}	$\overline{\text{SHIFT.CLK}}$ to DATAOUT Delay*		200	ns	
t_{SCWR}	$\overline{\text{SHIFT.CLK}}$ Width (Read)	4 t_p	t_{CYC} − 11 t_p		
t_{SCWW}	$\overline{\text{SHIFT.CLK}}$ Width (Write)	t_p	t_{CYC} − 2 t_p		

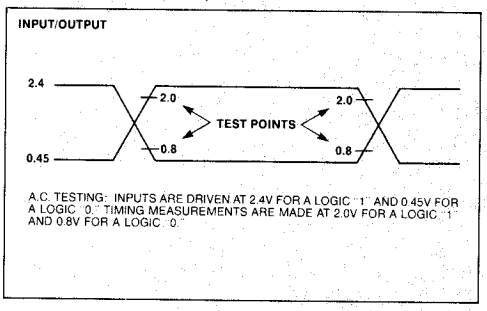
CAPACITANCE

(T_A = 25°C, V_{CC} = 0V, f = 1 MHz)

Symbol	Parameter	Min.	Max.	Unit	Test Conditions
C_{IN}	Input Capacitance		10	pF	
C_{OUT}	Output Capacitance		10	pF	
C_{DIO}	DIO Capacitance		10	pF	

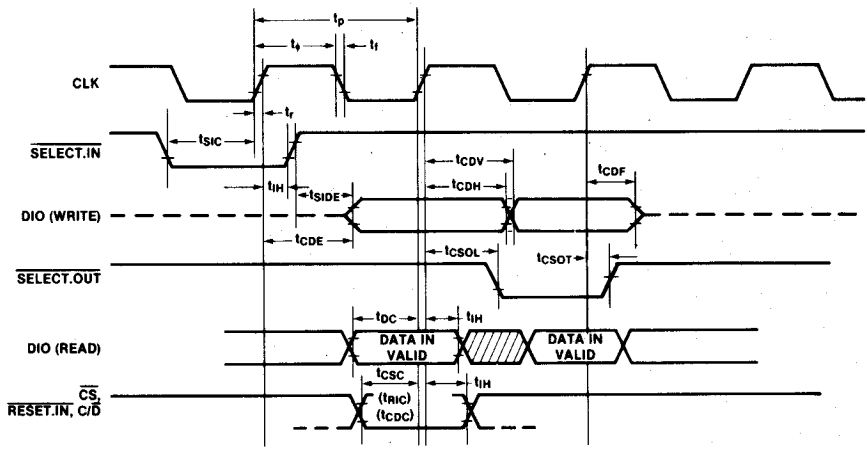
*DIO Write Mode.

A.C. TESTING INPUT, OUTPUT WAVEFORM

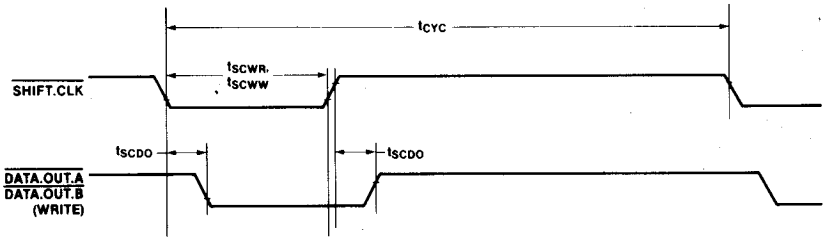


WAVEFORMS

DIO INTERFACE TIMING



BUBBLE DATA INTERFACE TIMING



intel

INTEL CORPORATION, 3065 Bowers Avenue, Santa Clara, CA 95051 (408) 987-8080

Printed in U.S.A./Q-90/11-82/15K/JR/JT

AFN-01358B