October 1983

8085 To BPK 72 Interface

Ulmont Smith Jr.
Applications Engineer

OCTOBER 1983
ORDER NUMBER: 210849-002

8085 TO BPK 72 INTERFACE

INTRODUCTION

Bubble Memory is quickly emerging as the preferred high density storage medium for a variety of microprocessor applications. Considering their size and reliability, Bubble Memory allows the designer to utilize the advantages of microprocessors in environments that were not possible using other high density peripheral storage technologies. Aside from portable or rugged environmental applications, bubbles also open up new design possibilities for desk-top terminal applications. Some of the benefits that can be realized from the implementation of Bubble Storage are increased flexibility, reduced maintenance, and non-volatility.

In addition to a one megabit Bubble Memory, Intel magnetics also manufactures a complete family of integrated-support circuits that simplify the task of designing with Bubble Memory. The family of support circuits provides an easy-to-use microprocessor interface via a single VLSI component, the Bubble Memory Controller (BMC). The remaining support circuits are controlled by the Bubble Memory Controller allowing the designer total freedom from the control signals associated with Bubble Memory technology.

At the component level, the BPK 72 (Bubble Memory Prototype Kit) provides the best opportunity to discover the potential of bubble storage. The BPK 72 comes complete with all the hardware and documentation necessary to prototype a one megabit (128K-bytes) Bubble Memory System. The BPK 72 is completely assembled and tested leaving the designer with the simple task of interfacing to a host processor.

This application note demonstrates how little effort is required to interface a BPK 72 with an 8085 microprocessor. The first four sections, "Introduction, BPK 72 Overview, Constructing the Hardware Interface, Implementing the 8085/BPK 72 Software Driver," and Appendix A (software listing) provide all the information necessary to interface a BPK 72 with an 8085 microprocessor based system. The remaining chapters describe in detail the hardware and software considerations involved with designing and implementing a Bubble Memory Interface.

A set of generalized flowcharts describing the software driver may also be found in Appendix A to facilitate the task of interfacing with other microprocessors.

BPK 72 OVERVIEW

The BPK 72 consists of a completely assembled and tested 10cm x 10cm printed circuit board containing a one megabit Bubble Memory and the complete family of integrated support circuits.

A block diagram of the BPK 72 is presented in Figure 1. It illustrates the key components in a one megabit, 128K-byte Bubble Memory System.

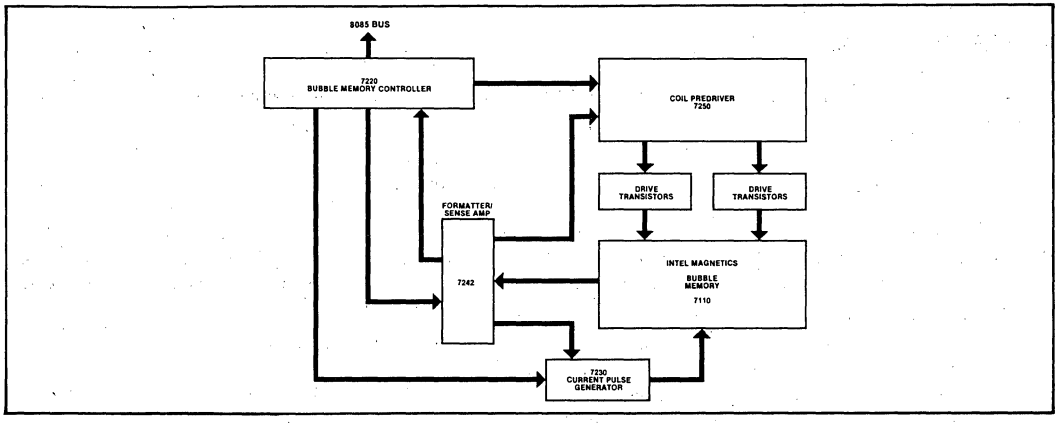


Figure 1. Block Diagram of the BPK 72

The 7110 Bubble Memory Module is supported by the following integrated circuits:

7220-1 Bubble Memory Controller (BMC)
The 7220-1 provides a convenient microprocessor interface and generates the timing signals necessary for the proper operation of the remaining support circuitry.

7242 Formatter Sense Amplifier (FSA)
The 7242 is responsible for detecting and enabling the generation of magnetic bubbles within the 7110. The 7242 also performs data formatting tasks and the option of automatic error detection and correction.

7250 Coil Predriver and 7254 Drive Transistors
The 7250 and two 7254s supply the drive currents for the rotating magnetic field that move the magnetic bubbles within the 7110 Bubble Memory Module.

7230 Current Pulse Generator (CPG)
The 7230 generates a set of waveforms necessary to input and output data from the 7110.

CONSTRUCTING THE HARDWARE INTERFACE

The hardware necessary to interface a BPK 72 with an 8085 microprocessor consists of a few simple connections to the system bus and the addition of only three integrated circuits; 7406—hex inverter (open collector), 7430—eight input nand gate, and an 8284A—Intel clock generator.

A schematic is presented in Figure 2 of the interface logic between a BPK 72 and the demultiplexed bus from an 8085 microprocessor.

The interface uses the eight input nand gate to enable chip-select on the BPK 72 when an I/O instruction is executed at ports 0FEH ("H" designates hexadecimal notation) or 0FFH. The address line A8 from the microprocessor bus is connected to A0 on the BPK 72 to select one of two internal ports. If the ports 0FEH and 0FFH are not available, simply connect A8 to the input of the nand gate and move a higher order address line (A9–A15) to A0 on the BPK 72. In the event that the I/O addresses are changed, the user must enter the new port locations into the software driver (see Appendix A). The I/O port locations are initialized as equates at the beginning of the program. All system dependent variables have been parameterized whenever possible.

The designer has the option of memory mapping the BPK 72 or utilizing 2 of the 256 I/O ports available on the 8085. The I/O ports were chosen for this interface to simplify the address decoding and to provide easy access to existing systems.

POWER SUPPLY REQUIREMENTS

The BPK 72 operates on standard +5V and +12V DC power within a 5% tolerance. The worst case power consumption is a follows:

+5VDC = 2 watts maximum
+12VDC = 5 watts maximum

When power is applied or removed from a Bubble Memory System, the rotating magnetic field within the 7110 Bubble Memory is held in the proper phase to insure non-volatility. This is accomplished through the use of a power fail reset circuit. The following power supply specifications must be observed to effectively support the power fail circuitry:

A. VDD = +12V, ±5% tolerance
   Power off/power fail voltage decay rate—less than 1.1 volts/millisecond
B. VCC = +5V, ±5% tolerance
   Power off/power fail voltage decay rate—less than 0.45 volts/millisecond
C. Voltage sequencing—no restrictions
D. Power on voltage rate of rise—no restrictions

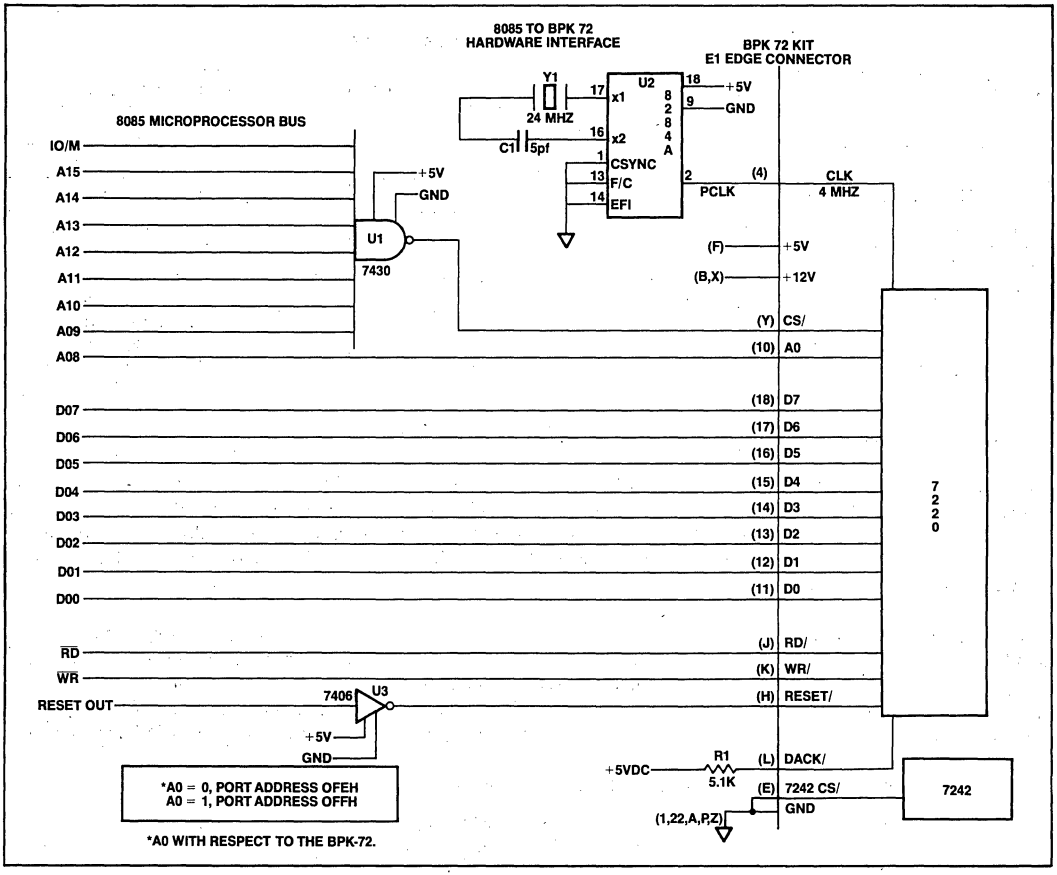


Figure 2. Hardware Interface

The interface designer should verify that the system power supply decay rates meets the specifications previously listed. To simulate worst case conditions, connect a 2 watt load on the +5 volt supply and a 5 watt load on the +12 volt supply. The power supply decay rates can be easily measured during the removal of power with a standard oscilloscope. No attempt should be made to use the BPK 72 until the power supply decay rates have been verified.

Table 1. 8085/BPK 72 Interface Parts List

Item	Description	Quantity	Reference	Manufacturer
1	IC-7430—8 input nand gate	1	U1	any
2	IC-8284A—clock generator	1	U2	Intel
3	IC-7406—hex inverter open collector	1	U3	any
4	Crystal—24.0000MHz fundamental mode, series resonant	1	Y1	any
5	Resistor—5.1Kohm, 1/4W, 5%	1	R1	any
6	Mica Capacitor—5pf, 100VDC, 5%	1	C1	any
7	Edge connector, 44 pin	1	E1	TRW, CINCH #50-44B-10

IMPLEMENTING THE 8085/BPK 72 SOFTWARE DRIVER

An 8085 to BPK 72 software driver program listing is presented in Appendix A. The driver consists of a set of subroutines that can be called to perform commonly used Bubble Memory commands. A detailed description and flowchart of each subroutine is provided with the program listing. The software driver is relocatable and may be linked with other programs. The name of the program is "BPK72." It begins at 0800H and requires less than 1K bytes of memory allocation.

The software driver is written in 8085 assembly language. It can be easily incorporated into existing systems as part of a utility program to transfer data between the BPK 72 and the 8085's addressable memory. The subroutines have been designed to eliminate the need for any further software development concerning the operation of the BPK 72. Assembly was chosen over higher level languages to provide the most efficient and portable code. With only minor modifications to the parameterized variables, the program, "BPK72," will run on almost any 8085 based system.

The following subroutines in the program "BPK72" will now be discussed:

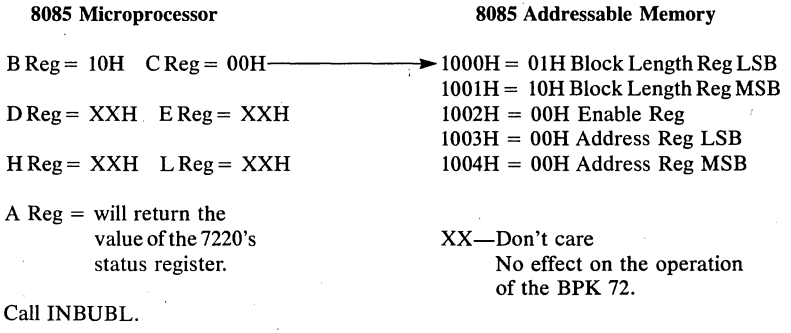
INBUBL—Initialize Bubble Memory
WRBUBL—Write Bubble Memory data
RDBUBL—Read Bubble Memory data
ABORT—Abort present command, reset BPK 72

INITIALIZING THE BUBBLE

After powering up, the BPK 72 must be initialized before any data transfers can begin. Initialization is needed to synchronize the 7220 Bubble Memory Controller with the data in the 7110 Bubble Memory storage loops and also because the 7110 employs redundancy. The 7110 Bubble Memory contains 320 storage loops. However, only 272 of the 320 loops are necessary for a 100% functional one megabit part. The additional 48 loops provide a 15% redundancy. Redundancy is used to significantly increase the yield of Bubble Memory modules during manufacture.

A map of the active and inactive loops is placed on a label attached to the case of the 7110. The same map is also placed in the 7110 during final test. When the system is initialized, the 7220 reads the map (boot loop) from the 7110 and decodes it. The boot loop is transferred from the 7220 into a pair of boot loop registers in the 7242 formatter sense amplifier. The boot loop registers are used to format data to insure that only functional loops are enabled during read or write operations.

Only one call to the initialization subroutine, INBUBL, is necessary to initialize a BPK 72. The following is an example of how to call INBUBL:



The example shown above demonstrates how to set up the B–C registers prior to calling the initialization subroutine, INBUBL. The B–C register pair must contain the address of the first of five consecutive locations within the 8085's addressable memory. In this example, the B–C registers are pointing to the first of five memory locations starting at 1000H. The data contained in 1000H through 1004H is a memory image of the parametric registers within the Bubble Memory Controller. The parametric registers contain a set of flags and parameters that determine exactly how the 7220 will respond to a software command.

Note the values used for the block length and address registers. These values must always be used during the initialization process with a one megabit Bubble Memory System. The enable register is shown with a 00H indicating the absence of error detection and correction. The 7220 and 7242 provide an optional error detection and correction feature to enhance data integrity. It is recommended that first time users begin without the use of error correction. Later on if error correction is desired, a 20H should be placed in the memory location designated as the enable register. A discussion concerning the use of error correction may be found in the section titled, "Communicating with the 7220."

Figure 3 illustrates the sequence of program flow necessary to initialize a Bubble Memory System using the subroutine INBUBL. Note that Figure 3 includes a test of the Bubble Memory Controller's status register. The status register is separate from the parametric registers and contains information about error conditions, completion or termination of commands, and the 7220's readiness to transfer data. To simplify the task of verifying a successful initialization, INBUBL returns the value of the 7220's status register to the calling routine through the 8085's "A" register. A successful initialization will return a 40H status. All other values indicate a BPK 72 system failure. Consult Appendix B in the unlikely event that the subroutine INBUBL fails to return a successful status.

READING AND WRITING

Only one call to the subroutine RDBUBL or WRBUBL is necessary to transfer data between the BPK 72 and the 8085's addressable memory.

Like many high density peripheral storage devices, Bubble Memory data is organized into pages rather than bytes. The 7220 Bubble Memory Controller partitions the one megabit Bubble Memory into 2048 pages of either 64 or 68 bytes in length. The page length is dependent upon the use of automatic error detection and correction—64 bytes with error correction and 68 bytes without. Data transfers are specified in terms of whole pages. Therefore the minimum amount of data that can be transferred from one read or write command is 64 or 68 bytes.

The parametric registers are used to communicate to the controller which page or pages will be transferred during a read or write command. The address register LSB and the first three bits of the address register MSB define the starting page address for read or write commands. The block length register determines how many pages will be transferred starting at the location defined by the address register. Theoretically, data transfers can range from 1 to 2048 pages in length. However, this application limits the maximum data transfer between the BPK 72 and the 8085's

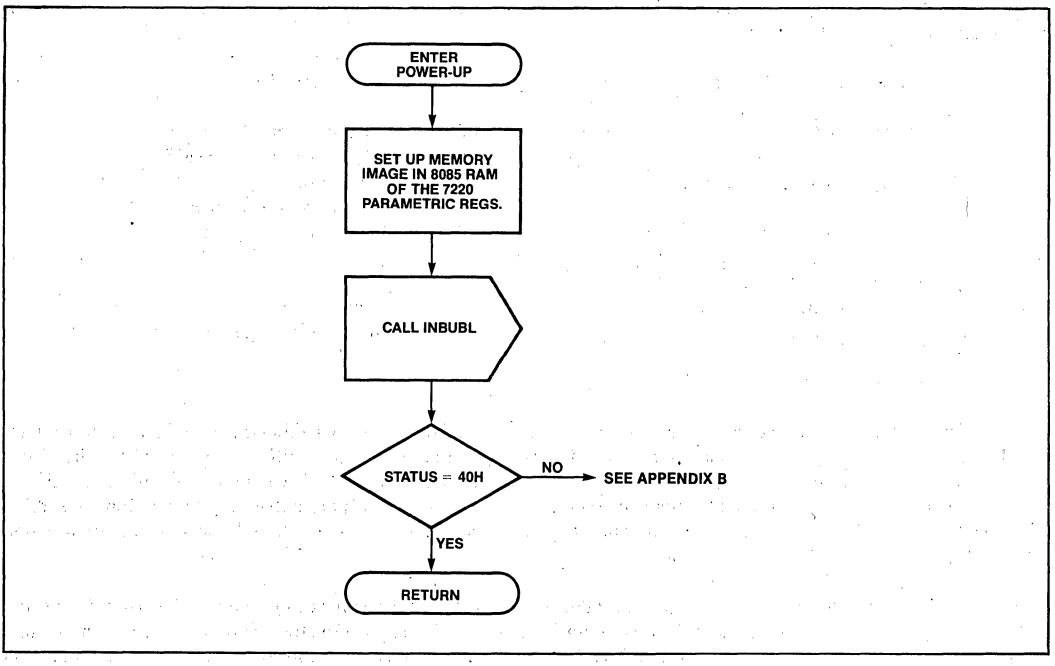


Figure 3. Initializing the BPK 72

memory to no more than 255 contiguous pages. This limitation results from the need to prevent data transfers that could exceed the addressable memory space of the 8085. The block length register LSB may be assigned any value between 1 and 255 depending on the size of the transfer. A detailed description of the parametric registers may be found in the section titled, "Communicating with the 7220."

The following is an example of how to use the Read Bubble Memory subroutine, RDBUBL, to transfer the first 16 pages (00H–0FH) of data from the BPK 72 to the 8085's addressable memory, starting at location 2000H:

8085 Microprocessor	8085 Addressable Memory
B Reg = 10H C Reg = 00H →	1000H = 10H Block Length Reg LSB
	1001H = 10H Block Length Reg MSB
D Reg = 20H E Reg = 00H →	1002H = 00H Enable Reg*
	1003H = 00H Address Reg LSB
H Reg = XXH L Reg = XXH	1004H = 00H Address Reg MSB
A Reg = will return the value of the 7220's status register.	2000H = start data transfer 243FH = last data transfer (1088 byte transfer)
	XX—Don't care. No effect on the operation of the BPK 72.
Call RDBUBL.	*—Assumes that the BPK 72 was initialized without error correction.

The Write Bubble Memory subroutine, WRBUBL, can be substituted for the call to RDBUBL to transfer data from the 8085's addressable memory to the first 16 pages in the BPK 72.

The example shown above demonstrates how to set up the B–C and D–E registers prior to calling a read or write subroutine. Just as in the case of initialization, the B–C registers contain the address of the first of five consecutive memory locations within the 8085's addressable memory. The data contained in the memory addressed by the B–C registers is used to load the 7220's parametric registers. The D–E register pair contains the address of the first byte of data to be transferred to or from the 8085's addressable memory.

Figure 4 illustrates how the read and write subroutines, RDBUBL and WRBUBL, should be called from another routine. The flowchart includes a program path to handle errors in the unlikely event that the read or write subroutines fail to return a successful status. First time users can omit the additional program flow for preliminary evaluation. The next section, "Checking the Status," describes the appropriate status values necessary to verify a successful data transfer.

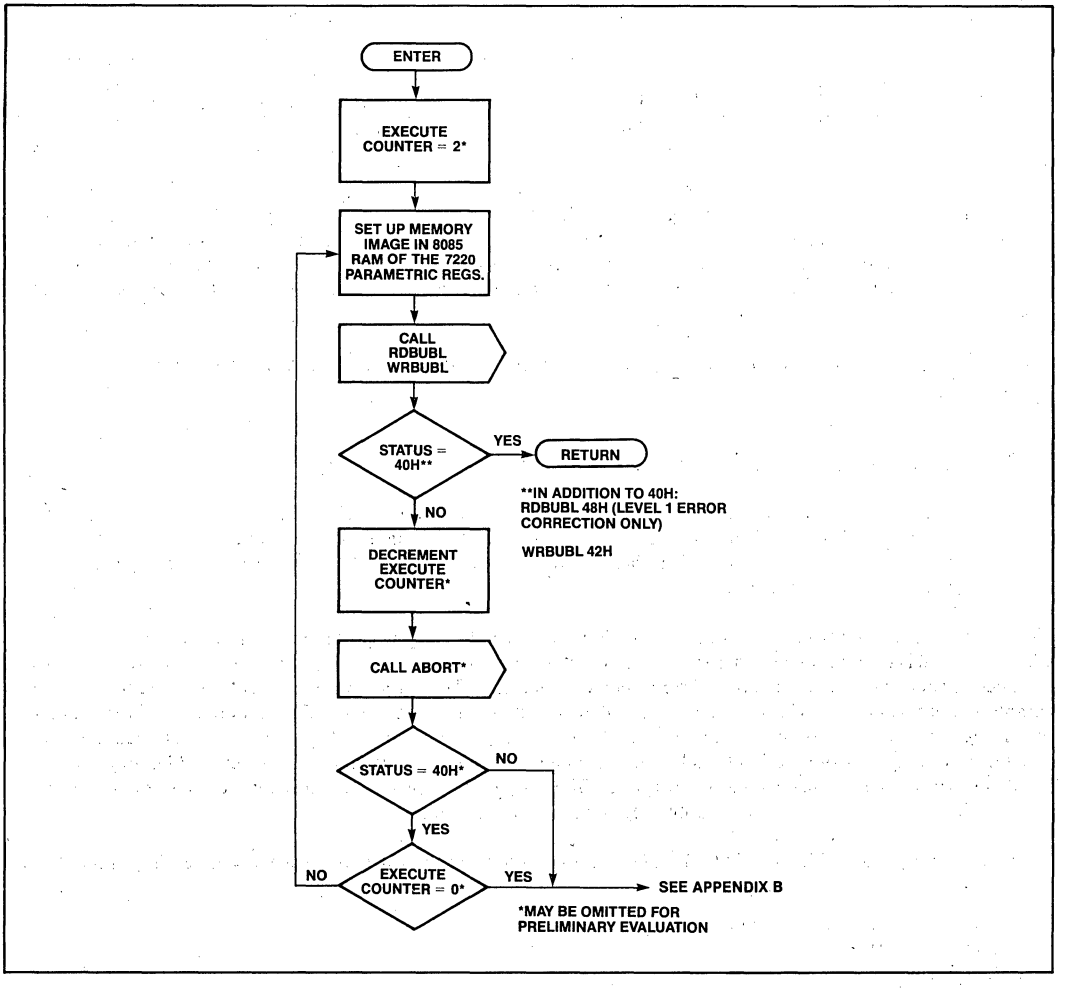


Figure 4. Reading and Writing to the BPK 72

CHECKING THE STATUS

After calling a subroutine to initialize, read, or write Bubble Memory data, the 7220's status register should be read to verify that the command was successfully executed. Note that flowcharts 1 and 2 include a test of the status register to detect for any errors. In order to facilitate the task of verification, each of the commonly used subroutines in the program "BPK72" return the contents of the 7220's status register to the calling routine through the 8085's "A" register. It is the responsibility of the calling routine to verify the success of each subroutine. A list of acceptable status register values for each of the subroutines in the program "BPK72" is presented in Table 2.

Table 2. Acceptable Status Register Values

Subroutine	Acceptable Status Register Value(s)	Comments
INBUBL	40H	OP-complete
WRBUBL	40H	OP-complete
	42H	OP-complete, parity error
RDBUBL	40H	OP-complete
	48H	OP-complete, correctable error*
ABORT	40H	OP-complete

*Level 1 error correction only

If any read errors are encountered during the transfer of data, they will almost always result from external noise interfering with the signal path between the 7110 Bubble Memory and the 7242 formatter sense amplifier. Since the data within the Bubble Memory is usually correct, a second attempt to transfer data should be successful. Figure 4 illustrates the use of the ABORT command to reset the Bubble Memory Controller before making another attempt to read or write Bubble Memory data.

Service information is presented in Appendix B in the unlikely event that any of the subroutines in Table 2 do not function properly.

7220 MICROPROCESSOR INTERFACE OVERVIEW

The key to any interface incorporating a BPK 72 is the Bubble Memory Controller. The controller provides a complete interface to a TTL level microprocessor bus that allows the designer total freedom from the intricate timing and waveforms necessary to support a Bubble Memory System. A block diagram of the 7220 Bubble Memory Controller is presented in Figure 5.

The 7220 interface circuitry consists of one 8-bit bidirectional port. The port provides access to internal registers. The address line A0 is used to select either the command/status or parametric/data registers. A command register is used to issue instructions such as read or write Bubble Memory data. The status register provides information about the completion or termination of commands and the 7220's readiness to transfer data. The parametric registers contain a set of flags and parameters that determine exactly how the 7220 will respond to a software command. The data register is actually a forty byte FIFO to buffer the timing differences between the 7110 Bubble Memory and a host processor. In order to transfer data to (from) the BPK 72, the host processor must load the parametric registers followed by issuing a read or write Bubble Memory data command.

To maintain design flexibility, the 7220 Bubble Memory Controller provides the user with three different modes of data transfer:

1. DMA, direct memory access
2. Interrupt-driven
3. Polled I/O

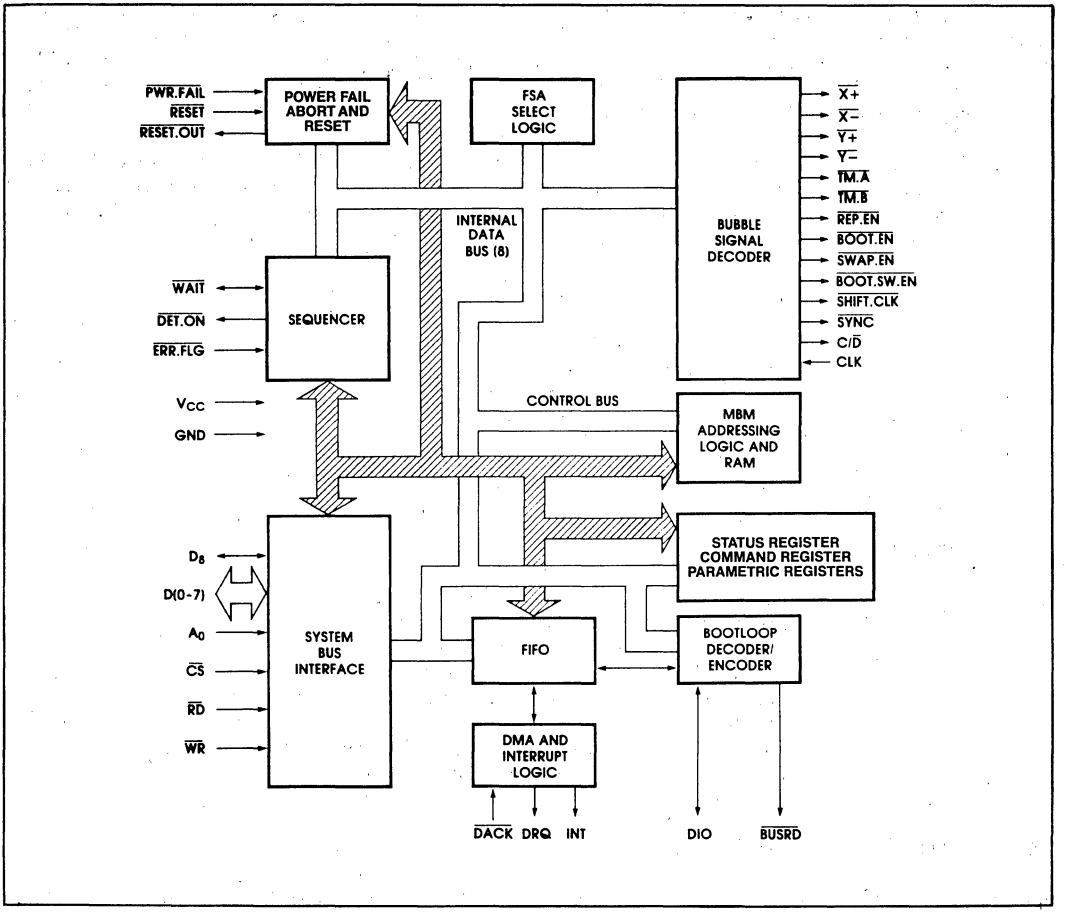


Figure 5. Block Diagram of the 7220 Bubble Memory Controller

In the DMA data transfer mode, the 7220 operates in conjunction with a DMA controller (such as Intel's 8257) using the DRQ (data request) and DACK (data acknowledge) lines for handshaking. With the help of a DMA controller, the 7220 transfers the data to (from) the host processor's memory. Once the data transfer begins, program intervention is not required until the entire data transfer has been completed.

In the interrupt mode, the 7220 along with an interrupt controller (such as Intel's 8259) uses the DRQ (data request) line to initiate a data transfer. The DRQ line becomes active when the 7220 is ready to send or receive a burst of data. A typical data burst is 22 contiguous bytes for an interrupt-driven interface. A set of software drivers are also necessary to service the interrupts to coordinate the transfer of data between the 7220 and the memory associated with a host processor. One advantage to the interrupt mode is multitasking. Since the host processor is only servicing the 7220 during data transfers, dead time between data transfers can be utilized for other processor tasks.

A polled mode interface reads the 7220 status register to determine when to transfer one byte of data. Of all the interface modes, polled I/O is the simplest configuration to implement. No special hardware or external controllers are necessary to interface the 7220 with a microprocessor. The major portion of a polled mode design is the software. Just as in the interrupt mode, a set of software drivers are required to read and write data to the 7220.

This application uses a polled mode configuration. The polled I/O data transfer mode was selected over DMA and interrupt-driven to simplify the interface design. A polled mode interface does not require the use of a DMA or interrupt controller. Furthermore, the polled mode interface provides the most flexibility for incorporating a BPK 72 into existing 8085 systems. Since the majority of a polled mode design consists of software, simple program modifications to accommodate existing systems can be easily entered into the software driver provided in Appendix A.

In terms of performance, the polled I/O transfer mode is the lowest compared to DMA or interrupt-driven. The DMA and interrupt modes offer the advantage of multitasking. However, the average access time and data transfer rate remain the same for each data transfer mode. The following formulas and examples demonstrate how to calculate the transfer time for a one megabit Bubble Memory System:

READ N-page transfer:
Transfer time = seek time + 8.7 ms + 7.5 ms (N−1)

WRITE N-page transfer:
Transfer time = seek time +7.5 ms (N)

Average seek time = 41 ms
Worst case seek time = 82 ms
Average data rate = 8.5 K-bytes/sec

For Example:

A. Time to read 1 page (assuming avg seek time):
Transfer time = 41 ms + 8.7 ms = 49.7 ms
B. Time to write 1 page (assuming avg seek time):
Transfer time = 41 ms + 7.5 ms = 48.5 ms
C. Time to read 10 contiguous pages (assuming avg seek time):
Transfer time = 41 ms + 8.7 ms + 7.5 ms (10–1) = 117.2 ms
D. Time to write 10 contiguous pages (assuming avg seek time):
Transfer time = 41 ms + 7.5 ms (10) = 116.0 ms

HARDWARE INTERFACE DESCRIPTION

To simplify the task of interfacing a BPK 72 with a microprocessor, the 7220 Bubble Memory Controller provides a convenient set of TTL signals that may be directly connected to a system bus. The interface signals on the BPK 72 necessary to implement a polled mode configuration are presented in Table 3.

PARITY BETWEEN THE 8085 AND BPK 72

The 7220 has the capability of generating and detecting odd parity using the bidirectional data line D8. The parity bit may be used to increase the reliability of the data path between the 7220 and a host processor. During data transfers, odd parity is generated for read operations and tested for write operations. The host processor may read the 7220 status register to determine if a parity error occurred during a write operation. Parity is typically implemented when a long transmission path exists between the host processor and the 7220. Since most systems utilize a simple edge connector backplane and a short transmission path (less than 18 inches), parity is not necessary. Parity is not implemented in this application to minimize the hardware complexity.

The parity bit, D8, is not stored within the 7110 Bubble Memory module. A separate and more effective error detection and correction feature is available as an option to increase the data integrity within the 7110. See the section titled, "Communicating with the 7220" for further details about the option of automatic error detection and correction.

Table 3. BPK 72 Polled Mode Interface Signals

Signal	Function
A0	Address line A0 = 0 Selects the FIFO data buffer or the parametric registers. A0 = 1 Selects command/status registers.
D0–D7	8 bit bidirectional data bus.
D8	Optional odd parity bit, not used in this application.
CS/	Chip select input. A logic high will tri-state the 7220 interface signals. (Slash, "/" designates a low active signal, system ground)
RD/	Read 7220 registers or data FIFO.
WR/	Write 7220 registers or data FIFO.
DACK/	DMA acknowledge. If DMA is not used, DACK/ requires an external pullup resistor to VCC (5.1 Kohm).
CLK	4 MHz TTL level clock. Clock period = 250 ns, 0.25 ns tolerance. Duty cycle = 50%, 5% tolerance.
RESET/	A low on this pin forces the interruption of any 7220 activity, performs a controlled shut-down, and initiates a reset sequence. The next instruction following RESET/ must be an abort command.
7242 CS/	7242 chip select signal is used to select banks of 7242s. 7242 CS/ must be tied low (system ground) for a single bank configuration.

4 MHZ CLOCK

The BPK 72 requires an external 4 MHz (may be asynchronous with respect to a host processor) TTL level clock. The specifications for the period and duty cycle are presented in Table 3. The 7220 uses the external clock to generate the timing signals that control the rotating magnetic field within the 7110 Bubble Memory. For reliable operation, the clock tolerances must be observed to assure that the rotating field is stable and accurate.

An Intel integrated circuit, 8284A clock driver, is used to generate the 4 MHz external clock. The 8284A along with a 24MHz series resonant crystal (fundamental mode) will provide a precise and accurate clock for any interface incorporating a BPK 72. The circuit configuration for the 8284A is illustrated in Figure 2. Other techniques of clock generation are acceptable as long as the duty cycle and period are within the specifications listed in Table 3.

SOFTWARE INTERFACE DESCRIPTION

The software driver presented in Appendix A contains the following subroutines that may be called from another routine:

* INBUBL —Initialize the BPK 72.
* RDBUBL —Read Bubble Memory data.
* WRBUBL —Write Bubble Memory data.
ABORT —Abort present command, reset BPK 72.
FIFORS —Reset 7220 FIFO data buffer.
WRFIFO —Write 7220 FIFO data buffer.
RDFIFO —Read 7220 FIFO data buffer.
WRBLRS —Write 7242 boot loop registers.
RDBLRS —Read 7242 boot loop registers.
MBMPRG —Bubble Memory purge command.
** RDBOOT —Read Bubble Memory boot loop.
** BOOTUP —Write Bubble Memory boot loop.

* Most commonly used commands.
** Diagnostic routines (see Appendix B).

Each of the subroutines listed above is described in further detail in Appendix A. Along with each subroutine is a generalized flowchart displaying the program flow. The user is encouraged to read the software driver to better understand the software interaction necessary to interface a BPK 72 with an 8085 microprocessor.

COMMUNICATING WITH THE 7220

Some additional background is necessary to understand the operation of the 7220 Bubble Memory Controller. Figure 6 illustrates the user-accessible registers that control and format the flow of data between the 7110 Bubble Memory and a host processor.

The address assignments for the user-accessible registers within the 7220 are presented in Table 4. The registers are listed in two groups. The first group (status, command, register address counter) consists of those registers that are selected and accessed in one operation. The second group contains the FIFO data buffer and the parametric registers (utility, block length, enable, address), they are selected according to the contents of the register address counter (RAC).

Table 4. 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	0	1	C	C	C	C	CMDR	Command Register	Write Only
1	0	0	0	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

NOTES:
SSSSSSSS = 8-bit status information returned to the user from the STR
CCCC = 4-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

Table 5. Parametric Registers and FIFO Data Buffer

	RAC						
A0	B3	B2	B1	B0	Symbol	Name of Register	Read/Write
0	1	0	1	0	UR	Utility Register	Read or 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	Write Only
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

To successfully implement the hardware and software presented in this application, certain restrictions are placed on the contents of the user-accessible registers. Each of the user-accessible registers and any necessary restrictions will now be discussed in further detail.

COMMAND REGISTER

The 7220 command set consists of 16 commands identified by a 4 bit command code. A list of the commands is presented in Table 6.

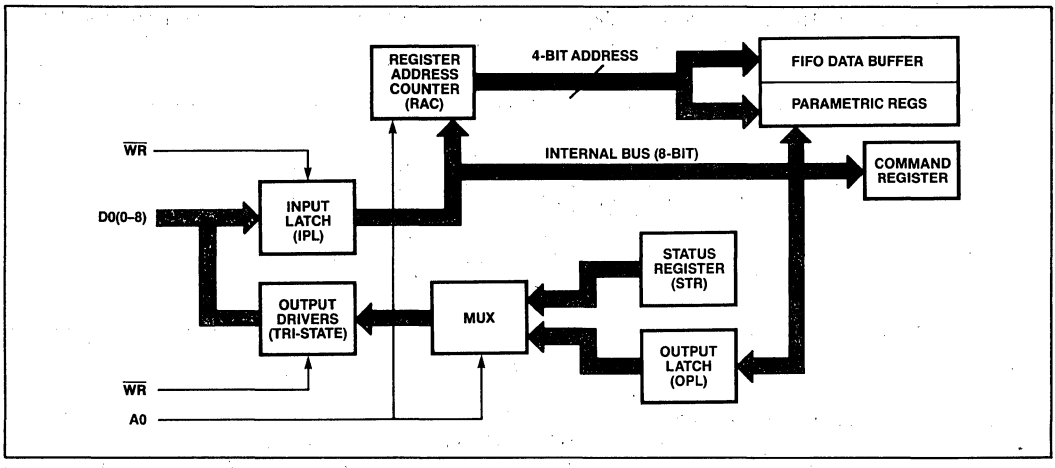


Figure 6. 7220 User Accessible Registers

Table 6. 7220 Commands

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

The commands listed in Table 6 are provided for reference purposes only. The software driver in Appendix A consists of a series of subroutines that automatically issue the appropriate commands to perform a data transfer.

The function of each command is usually apparent from the command name (e.g., initialize, read bubble data, write bubble data). Additional detail concerning the function of each command may be found in the BPK 72 user's manual.

REGISTER ADDRESS COUNTER

The register address counter consists of a 4 bit address that points to one of the six parametric registers:

Utility register (UT)—The utility register is a general purpose register available to the user in connection with Bubble Memory System operations. It has no direct effect on the operation of the 7220. It is provided as a convenience to the user.

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

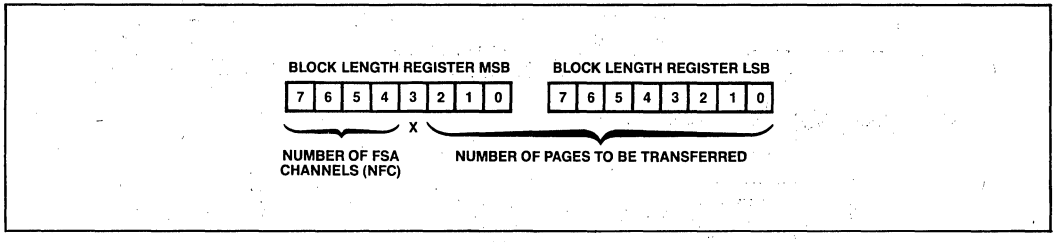


Figure 7. Block Length Registers

The 7220 has the capability of supporting up to eight 7110 Bubble Memory modules. Each 7110 contains two channels that are sensed by a 7242 formatter sense amplifier (FSA). In multiple Bubble Memory configurations, the BLR allows the user to select the page size. Since the BPK 72 consists of only one Bubble Memory module, the field specifying the number of FSA channels in the BLR MSB must contain −0001B ("B" designates a binary notation). After the FSA field is set, the page size is dependent upon the use of error detection and correction. Error correction will be discussed in the next section describing the function of the enable register.

The BLR LSB and the first 3 bits of the BLR MSB determine the number of pages to be transferred during a single read or write command. This application restricts the user to no more than 255 contiguous pages to prevent data transfers that could exceed the addressable memory space of the 8085.

For This Application

BLR MSB—10H at all times.
("H" designates a hexadecimal notation)

BLR LSB—Selectable from 01H to FFH (1 to 255 pages).

CAUTION: 00H in the BLR LSB will enable a 2048 page transfer resulting in a timing error.

Enable Register (ER)—The user sets the bits in the enable register to enable or disable various functions within the 7220. The individual bit descriptions are as follows:

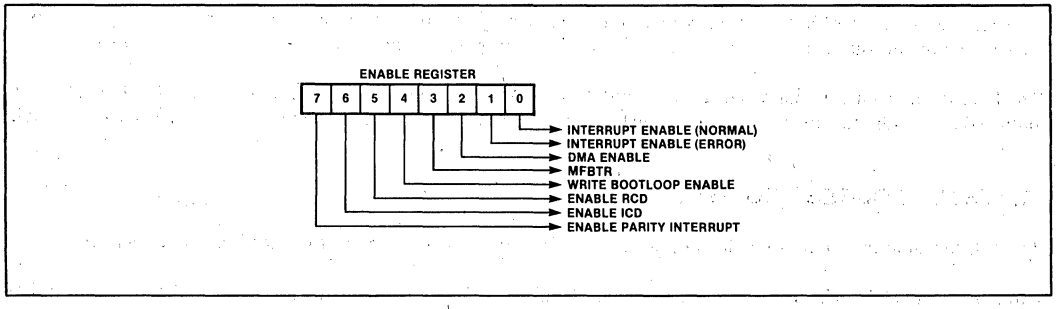


Figure 8. Enable Register

One of the most important functions concerning the enable register is the option of automatic error detection and correction. If error correction is enabled during a write operation, the 7242 formatter sense amplifier appends each 256 bit block of data with a 14 bit fire code. Both the data and the fire code are stored within the 7110 Bubble Memory module. During a read operation, the 7242 compares the data with the fire code to check for any errors. With respect to the FSA, errors are either correctable (the FSA is able to reconstruct the data using an error correction algorithm before transferring the data to the 7220) or uncorrectable. Additional information about the fire code is available in the BPK 72 user's manual.

The enable register offers three levels of error correction. All three levels utilize the same error correction algorithm but differ in their interaction with a host processor. Table 6 defines the relevant register bits for the various levels of error correction.

Table 6. Error Correction Levels

Error Correction Level	Bit 6 (ICD)	Enable Register Bit 5 (RCD)	Bit 1 (Int Enable)
Level 0	0	0	0
Level 1	0	1	0
Level 2	1	0	0
Level 3	1	0	1

Level 0 does not enable the error detection and correction algorithm. In this mode, the 7220 partitions one megabit systems into 2048 pages consisting of 68 bytes per page.

Level 1 is the most popular level of error correction. If an error is detected during a read operation, the 7242 automatically cycles the data through its error correction algorithm and transfers the data to the 7220. If the error was correctable, the 7220 will continue to function normally i.e., correctable errors in Level 1 are transparent to the host processor. If the error was uncorrectable, the 7220 will stop reading at the end of the page wherein the error was encountered. In the unlikely event that the 7220 stops because of an uncorrectable error, the host processor should try at least one more attempt to read the data. In most cases, errors result from random noise that can interfere with the signal path between the 7110 and 7242. Since the data is usually correct within the 7110, another attempt to read the data should yield a successful status.

Level 2 and Level 3 differ from Level 1 in that page-specific logging of uncorrectable errors is possible and the transfer of erroneous data can be prevented. Level 3 differs from Level 2 in that Level 3 also allows the logging of correctable errors.

Neither Level 2 nor Level 3 is supported by this application because the probability of an uncorrectable error is typically one in 10^{16} bits read. An error rate of this magnitude will produce few if any uncorrectable errors throughout the useful life of a Bubble Memory System.

It is recommended that Level 1 error correction be utilized to improve the integrity of the data within the 7110. In Level 1, the 7220 assigns 64 bytes to a page in one megabit Bubble Memory Systems.

Aside from error correction, the enable register performs many other functions.

Enable Parity Interrupt—If this bit is set, any parity errors between the host and the 7220 during write operations will generate an interrupt. Since parity and the interrupt mode are not used in this application, the enable parity interrupt bit should be reset to a logical zero.

Write Bootloop Enable—This bit must be reset to prevent accidental erasure of the boot loop within the 7110.

MFBTR—The MFBTR bit should always be reset to maximize the data transfer rate between the 7220 and 7242 during read operations.

DMA Enable—If this bit is set, the 7220 will attempt to transfer data in the DMA mode. Since this application utilizes a polled mode interface, this bit must be reset to a logical zero.

Interrupt Enable (Normal)—If this bit is set, an interrupt is sent to the host processor after the successful completion of a Bubble Memory command. Since this application uses a polled mode interface, this bit should be reset to a logical zero.

For This Application

Enable Reg—00H. No error correction.
—20H. Level 1 error correction.

Address Register (AR)—The contents of the address register determine which starting address locations will be used during a read or write command. For systems with a multiple Bubble Memory configuration, an additional magnetic Bubble Memory (MBM) select field is used to specify which Bubble Memory(s) will be selected. The bit configuration is as follows:

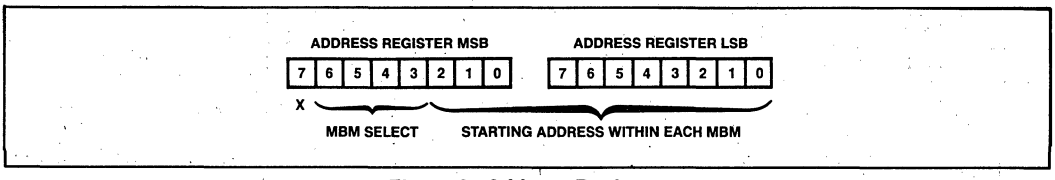


Figure 9. Address Registers

Since the BPK 72 consists of only one 7110 Bubble Memory module, the MBM select field must contain −0000B ("B" designates a binary notation).

For This Application

AR MSB—00000XXX

AR LSB—XXXXXXXX, X = user selectable page address
from 0 to 2047.

STATUS REGISTER

In a polled data transfer mode, the status register provides information about error conditions, completion or termination of commands, and the 7220's readiness to transfer data or accept new commands. The bit configuration for the status register is as follows:

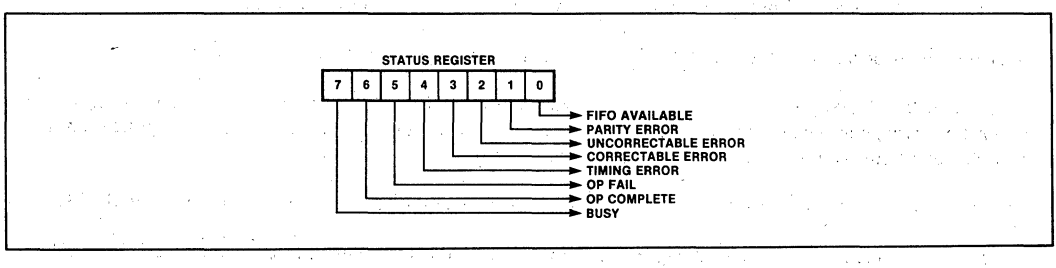


Figure 10. Status Register

Busy—When active (Logic 1), the Busy bit indicates that the 7220 is in the process of executing a command. Bits 1 through 6 of the status register are valid only when the busy bit is not active (Logic 0).

OP Complete—When active (Logic 1), the OP Complete bit indicates the successful completion of a command.

OP Fail—When active (Logic 1), the OP Fail bit indicates that the 7220 was unable to successfully complete the current command.

Timing Error—When active (Logic 1), the Timing Error bit indicates that an FSA has reported a timing error to the 7220, or that the host system has failed to keep up with the required data rate during a read or write operation.

Correctable Error—When active (Logic 1), the Correctable Error bit indicates that an FSA has detected a correctable error in the last block of data read from the 7110.

Uncorrectable Error—When active (Logic 1), the Uncorrectable Error bit indicates that an FSA has detected an uncorrectable error in the last block of data read from the 7110.

Parity Error—When active (Logic 1), the Parity Error bit indicates that a parity error was detected between the 7220 and the host processor. Parity errors are only detected by the 7220 during write operations. Since parity is not used in this application, ignore all parity errors.

FIFO Ready—When the 7220 is busy, an active FIFO Ready bit (Logic 1) indicates that the FIFO has data for reading or space for writing. When the 7220 is not busy, the FIFO Ready bit (Logic 0) indicates that the 40 byte FIFO and the input and output latches are completely empty.

SUMMARY

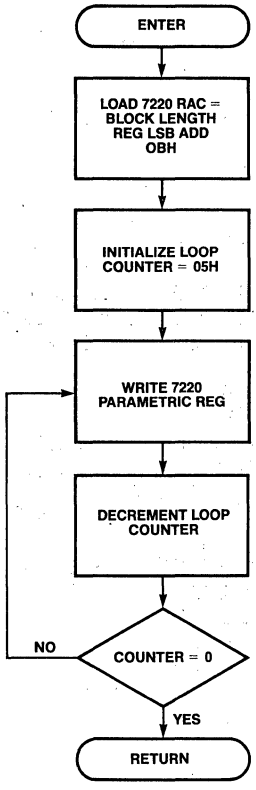
This application note is intended to eliminate almost all of the development effort necessary to interface an 8085 microprocessor with a BPK 72. With the addition of only a few IC's and the software driver presented in Appendix A, the designer is well on the way to incorporating the benefits of improved reliability, reduced maintenance, and non-volatility into any 8085 microprocessor based system.

APPENDIX A
8085 TO BPK-72 INTERFACE SOFTWARE DRIVER LISTING AND FLOWCHARTS

```
ASM80 :F1:BPKHDR


ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0        BPK72    PAGE    1


 LOC  OBJ        LINE       SOURCE STATEMENT

                   1 ;*******************************************************************************
                   2 ;
                   3 ;
                   4 ;                        PROGRAM: 8085 TO BPK72 SOFTWARE DRIVER V1.0
                   5 ;                        ULMONT S. SMITH JR.
                   6 ;                        INTEL CORPORATION
                   7 ;                        3065 BOWERS AVENUE
                   8 ;                        SANTA CLARA, CALIFORNIA 95051
                   9 ;
                  10 ;
                  11 ;*******************************************************************************
                  12 ;
                  13 ;
                  14 ;
                  15 ; ABSTRACT:
                  16 ;
                  17 ;       THIS PROGRAM  CONSISTS OF A SET OF  BUBBLE MEMORY SOFTWARE DRIVERS
                  18 ;       THAT SUPPORT A POLLED MODE INTERFACE BETWEEN A BPK72, 1MBIT BUBBLE
                  19 ;       MEMORY PROTOTYPE KIT,  AND  A STANDARD  8085  MICROPROCESSOR.  THE
                  20 ;       PROGRAM  UTILIZES A SET  OF PUBLIC  DIRECTIVES THAT  CAN BE CALLED
                  21 ;       TO PERFORM A BUBBLE MEMORY INITIALIZATION, READ, WRITE,  AND OTHER
                  22 ;       COMMONLY USED COMMANDS.  IN THE UNLIKELY EVENT THAT THE 7110 BUBBLE
                  23 ;       MEMORY BOOT LOOP IS LOST, TWO ROUTINES ARE PROVIDED TO EXAMINE AND
                  24 ;       REWRITE THE BOOT LOOP CODE.
                  25 ;
                  26 ;
                  27 ;
                  28 ; PROGRAM ORGANIZATION:
                  29 ;
                  30 ;       FUNCTIONS:
                  31 ;                        INTPAR
                  32 ;                        FIFORS
                  33 ;                        BYTCNT
                  34 ;                        WRITE
                  35 ;                        READ
                  36 ;                        ABORT
                  37 ;                        WRBUBL
                  38 ;                        RDBUBL
                  39 ;                        INBUBL
                  40 ;                        BOOTUP
                  41 ;                        RDBOOT
                  42 ;                        WRFIFO
                  43 ;                        RDFIFO
                  44 ;                        WRBLRS
                  45 ;                        RDBLRS
                  46 ;                        MBMPRG
                  47 ;
                  48 ;
                  49 ; EXTERNAL DECLARATIONS: NONE
                  50 ;
                  51 ;
                  52 $EJECT
```

```
ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0        BPK72    PAGE    2


 LOC  OBJ        LINE       SOURCE STATEMENT

                   53 ;
                   54 ; PUBLIC SYMBOLS:
                   55 ;
                   56 ;              FIFORS - RESET 7220 FIFO DATA BUFFER
                   57 ;              ABORT  - ABORT PRESENT COMMAND, RESET BPK72
                   58 ;              WRBUBL - WRITE BUBBLE MEMORY DATA
                   59 ;              RDBUBL - READ BUBBLE MEMORY DATA
                   60 ;              INBUBL - INITIALIZE THE BPK72
                   61 ;              BOOTUP - WRITE BUBBLE MEMORY BOOT LOOP
                   62 ;              RDBOOT - READ BUBBLE MEMORY BOOT LOOP
                   63 ;              WRFIFO - WRITE 7220 FIFO DATA BUFFER
                   64 ;              RDFIFO - READ 7220 FIFO DATA BUFFER
                   65 ;              WRBLRS - WRITE 7242 BOOT LOOP REGISTERS
                   66 ;              RDBLRS - READ 7242 BOOT LOOP REGISTERS
                   67 ;              MBMPRG - BUBBLE MEMORY PURGE COMMAND
                   68 ;
                   69 ;
                   70 ;*******************************************************************************
                   71 ;
                   72                          NAME BPK72
                   73 ;
                   74 ;*******************************************************************************
                   75 ;
                   76 ;
                   77 ;*******************************************************************************
                   78 ;
 0800              79                          ORG     0800H
                   80 ;
                   81 ;*******************************************************************************
                   82 ;
                   83 ;
                   84 ;*******************************************************************************
                   85 ;
                   86 ;                        PROGRAM EQUATES
                   87 ;
                   88 ;*******************************************************************************
                   89 ;
                   90 ;
 00FE              91 PRTA00  EQU     0FEH    ; A POLLED  MODE  INTERFACE  REQUIRES  ONLY  TWO  I/O
 00FF              92 PRTA01  EQU     0FFH    ; PORTS DESIGNATED BY THE A0 LINE ON THE BPK72 BOARD.
                   93                         ; THIS APPLICATION USES:
                   94                         ;
                   95                         ;       0FEH - A0=0 FOR PRTA00 (PORT A0= 0)
                   96                         ;              RD/WR BUBBLE MEMORY DATA AND REGS
                   97                         ;
                   98                         ;       0FFH - A0=1 FOR PRTA01 (PORT A0= 1)
                   99                         ;              RD STATUS REG
                  100                         ;              WR BUBBLE MEMORY COMMANDS
                  101 ;
                  102 $EJECT
```

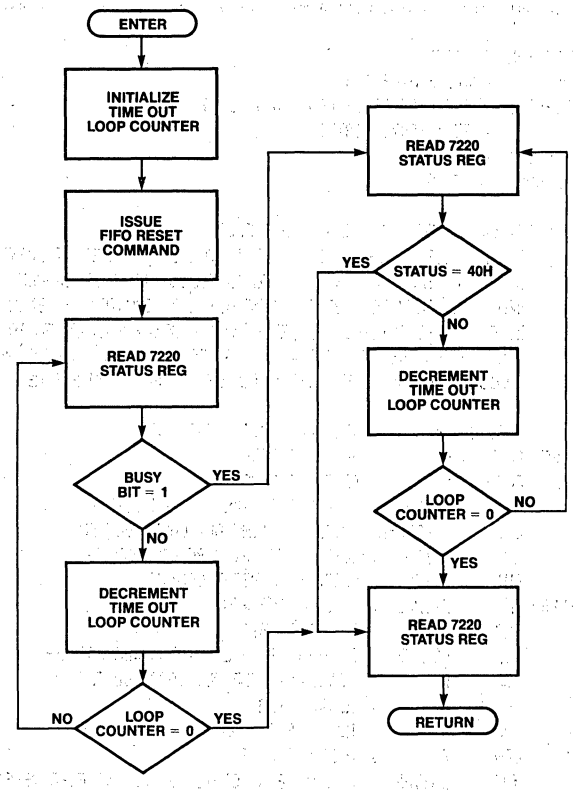
```
ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0        BPK72    PAGE    3


 LOC  OBJ         LINE        SOURCE STATEMENT

                   103 ;*******************************************************************************
                   104 ;
                   105 ; FUNCTION: INTPAR
                   106 ; INPUTS:   B-C REGS, STARTING ADDRESS OF PARAMETRIC REGS IN RAM
                   107 ; OUTPUTS:  7220 PARAMETRIC REGS
                   108 ; CALLS:    NONE
                   109 ; DESTROYS: A, F/FS
                   110 ;
                   111 ; DESCRIPTION: LOAD THE 7220 PARAMETRIC REGS
                   112 ;       THE B-C REGS  CONTAIN THE ADDRESS TO THE FIRST OF FIVE CONTIGUOUS
                   113 ;       MEMORY LOCATIONS IN RAM.  THE DATA  ADDRESSED BY THE  B-C REGS IS
                   114 ;       USED TO LOAD  THE PARAMETRIC  REGISTERS IN THE 7220 BUBBLE MEMORY
                   115 ;       CONTROLLER.  INTPAR COPIES THE DATA IN RAM TO THE PARAMETRIC REGS.
                   116 ;
 0800 C5           117 INTPAR: PUSH    B       ; SAVE B-C REGS
 0801 D5           118         PUSH    D       ; SAVE D-E REGS
 0802 3E0B         119         MVI     A,0BH   ; LOAD A REG WITH BLR LSB ADDRESS
 0804 D3FF         120         OUT     PRTA01  ; LOAD 7220 RAC WITH BLR LSB ADDRESS
 0806 1E05         121         MVI     E,05H   ; INITIALIZE LOOP COUNTER
 0808 0A           122 LOAD:   LDAX    B       ; LOAD A REG FROM B-C REG ADDRESS
 0809 D3FE         123         OUT     PRTA00  ; WRITE PARAMETRIC REG
 080B 03           124         INX     B       ; INCREMENT B-C REGS TO THE NEXT ADDRESS IN RAM
 080C 1D           125         DCR     E       ; DECREMENT LOOP COUNTER
 080D C20808       126         JNZ     LOAD    ; IF NOT ZERO, JMP LOAD
 0810 D1           127         POP     D       ; RESTORE D-E REGS
 0811 C1           128         POP     B       ; RESTORE B-C REGS
 0812 C9           129         RET             ; RETURN TO CALL
                   130 ;
                   131 ;
                   132 ;
                   133 $EJECT
```



COMMENTS: THE UTILITY REGISTER IS NOT USED. THE RAC IS AUTOMATICALLY INCREMENTED AFTER EACH WRITE (WR/) IS EXECUTED. THE RAC WILL NOT INCREMENT BEYOND 00H.

Figure 11. INTPAR

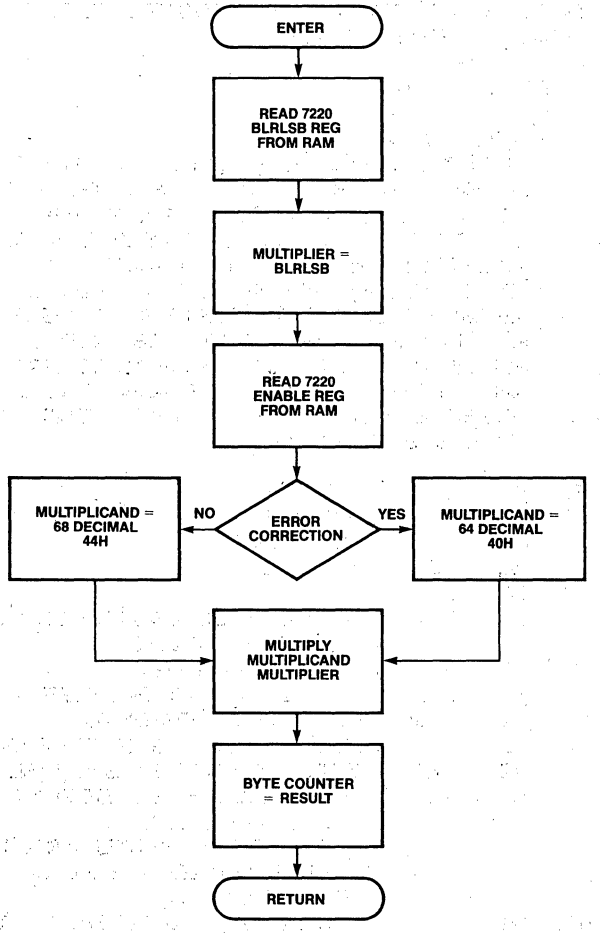
```
ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0        BPK72    PAGE    4


 LOC  OBJ        LINE       SOURCE STATEMENT

                  134 ;*******************************************************************************
                  135 ;
                  136 ; FUNCTION: FIFORS
                  137 ; INPUTS:   BPK72 STATUS REG
                  138 ; OUTPUTS:  ISSUE FIFO RESET COMMAND TO BPK72
                  139 ;           A REG= BPK72 STATUS REG
                  140 ; CALLS:    NONE
                  141 ; DESTROYS: A, F/FS
                  142 ;
                  143 ; DESCRIPTION: RESET 7220 FIFO DATA BUFFER
                  144 ;       A FIFO RESET COMMAND IS ISSUED TO THE BPK72.  AFTER ISSUING THE
                  145 ;       COMMAND,  THE BPK72 STATUS REG IS POLLED UNTIL AN OP-COMPLETE,
                  146 ;       40H,  HAS BEEN READ OR THE TIME OUT LOOP COUNTER DECREMENTS TO
                  147 ;       ZERO.  FIFORS RETURNS THE VALUE OF THE BPK72 STATUS REG TO  THE
                  148 ;       CALLING ROUTINE  VIA THE 8085'S A REG.  ONLY  A  STATUS OF 40H
                  149 ;       INDICATES A SUCCESSFUL EXECUTION OF THE FUNCTION FIFORS.
                  150 ;
                  151         PUBLIC  FIFORS  ; DECLARE PUBLIC FUNCTION
 0813 D5          152 FIFORS: PUSH    D       ; SAVE D-E REGS
 0814 C5          153         PUSH    B       ; SAVE B-C REGS
 0815 0640        154         MVI     B,40H   ; LOAD B REG= 40H, OP-COMPLETE
 0817 11FFFF      155         LXI     D,0FFFFH; INTIALIZE TIME OUT LOOP COUNTER
 081A 3E1D        156         MVI     A,1DH   ; LOAD A REG= FIFO RESET COMMAND
 081C D3FF        157         OUT     PRTA01  ; WRITE FIFO RESET COMMAND
 081E DBFF        158 BUSYFR: IN      PRTA01  ; READ STATUS REG
 0820 07          159         RLC             ; TEST BUSY BIT= 1
 0821 DA2E08      160         JC      POLLFR  ; IF BUSY= 1, POLL STATUS REG FOR 40H
 0824 1B          161         DCX     D       ; DECREMENT TIME OUT LOOP COUNTER
 0825 AF          162         XRA     A       ; CLEAR A REG
 0826 B2          163         ORA     D       ; TEST D REG= 00H
 0827 B3          164         ORA     E       ; TEST E REG= 00H
 0828 C21E08      165         JNZ     BUSYFR  ; IF NOT ZERO, CONTINUE POLLING FIFO RESET COMMAND
 082B C33B08      166         JMP     RETFR   ; TIME OUT ERROR, RETURN
 082E DBFF        167 POLLFR: IN      PRTA01  ; READ STATUS REG
 0830 A8          168         XRA     B       ; TEST STATUS= 40H, OP-COMPLETE
 0831 CA3B08      169         JZ      RETFR   ; IF OP-COMPLETE, JMP RETFR
 0834 1B          170         DCX     D       ; DECREMENT TIME OUT LOOP COUNTER
 0835 AF          171         XRA     A       ; CLEAR A REG
 0836 B2          172         ORA     D       ; TEST D REG= 00H
 0837 B3          173         ORA     E       ; TEST E REG= 00H
 0838 C22E08      174         JNZ     POLLFR  ; IF NOT ZERO, CONTINUE POLLING FIFO RESET COMMAND
 083B C1          175 RETFR:  POP     B       ; RESTORE B-C REGS
 083C D1          176         POP     D       ; RESTORE D-E REGS
 083D DBFF        177         IN      PRTA01  ; READ STATUS REG
 083F C9          178         RET             ; RETURN TO CALL
                  179 ;
                  180 ;
                  181 ;
                  182 $EJECT
```



COMMENT: MINIMUM TIME OUT LOOP COUNT = 50μS.

Figure 12. FIFORS

```
ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0        BPK72    PAGE    5


 LOC  OBJ        LINE       SOURCE STATEMENT

                  183 ;*********************************************************************************
                  184 ; FUNCTION: BYTCNT
                  185 ; INPUTS:   B-C REGS, STARTING ADDRESS OF PARAMETRIC REGS IN RAM
                  186 ; OUTPUTS:  H-L REGS= BYTE COUNTER
                  187 ; CALLS:    NONE
                  188 ; DESTROYS: A, H, L, F/FS
                  189 ;
                  190 ; DESCRIPTION: BYTE COUNTER
                  191 ;       THE B-C REGS CONTAIN THE ADDRESS TO THE FIRST OF FIVE CONTIGUOUS MEMORY
                  192 ;       LOCATIONS IN RAM.  THE DATA  ADDRESSED BY  THE B-C REGS IS USED TO LOAD
                  193 ;       THE PARAMETRIC REGS IN  THE 7220  BUBBLE MEMORY CONTROLLER.  THE ENABLE
                  194 ;       REG IS READ FROM RAM TO DETERMINE IF ERROR CORRECTION HAS BEEN ENABLED.
                  195 ;       THE USE OF ERROR CORRECTION REQUIRES A 64 BYTE TRANSFER/PAGE  - 68 BYTE
                  196 ;       TRANSFER/PAGE WITHOUT ERROR CORRECTION.  THE  BLOCK  LENGTH  REG LSB IS
                  197 ;       ALSO READ FROM RAM TO  DETERMINE THE NUMBER  OF PAGES TO BE TRANSFERRED
                  198 ;       DURING THE NEXT READ OR WRITE COMMAND.  THE  NUMBER  OF  BYTES PER PAGE
                  199 ;       MULTIPLIED BY THE NUMBER OF PAGES IS COMPUTED AND PASSED TO THE CALLING
                  200 ;       ROUTINE VIA THE 8085'S H-L REGS.  DATA TRANSFERS ARE LIMITED TO  16,320
                  201 ;       BYTES WITH ERROR CORRECTION AND  17,340 BYTES WITHOUT.  ONLY THE BLRLSB
                  202 ;       IS USED TO GENERATE THE BYTE COUNTER.
                  203 ;
  0840 C5         204 BYTCNT: PUSH    B       ; SAVE B-C REGS
  0841 D5         205         PUSH    D       ; SAVE D-E REGS
  0842 0A         206         LDAX    B       ; LOAD A REG WITH BLRLSB
  0843 6F         207         MOV     L,A     ; MOVE BLRLSB TO L REG
  0844 03         208         INX     B       ;
  0845 03         209         INX     B       ; INCREMENT B-C REGS TO ADDRESS THE ENABLE REG IN RAM
  0846 0A         210         LDAX    B       ; LOAD A REG WITH ENABLE REG
  0847 67         211         MOV     H,A     ; MOVE ENABLE REG TO H REG
  0848 1640       212         MVI     D,40H   ; INITIALIZE D REG 64 BYTES/PAGE XFER, 40H
  084A 3E60       213         MVI     A,60H   ; ERROR CORRECTION DETECTION MASK
  084C A4         214         ANA     H       ; LOGICAL AND MASK WITH H REG, TEST FOR ERROR CORRECTION
  084D C25208     215         JNZ     MULT    ; IF ZERO, ERROR CORRECTION IS NOT ENABLED
  0850 1644       216         MVI     D,44H   ; NO ERROR CORRECTION, 68 BYTES/PAGE XFER, 44H
                  217                         ; MULTIPLY (D REG) X (L REG)
                  218                         ; 64 OR 68 BYTES X NO. OF PAGES IN BLRLSB
                  219                         ; RESULT WILL BE PLACED IN THE H-L REGS
                  220                         ; BEGIN MULTIPLY ROUTINE
  0852 2600       221 MULT:   MVI     H,0H    ; INITIALIZE MOST SIGNIFICANT BYTE OF RESULT
  0854 1E09       222         MVI     E,09H   ; INITIALIZE BIT COUNTER
  0856 7D         223 MULT0:  MOV     A,L     ; MOVE LOW ORDER BYTE INTO A REG
  0857 1F         224         RAR             ; ROTATE LEAST SIGNIFICANT BIT OF MULTIPLIER
  0858 6F         225         MOV     L,A     ; MOVE LOW ORDER BYTE OF RESULT INTO L REG
  0859 1D         226         DCR     E       ; DECREMENT BIT COUNTER
  085A CA6708     227         JZ      DONE    ; EXIT IF COMPLETE
  085D 7C         228         MOV     A,H     ; MOVE HIGH ORDER BYTE INTO A REG
  085E D26208     229         JNC     MULT1   ; IF CARRY= 0, JMP MULTI
  0861 82         230         ADD     D       ; ADD D REG TO A REG
  0862 1F         231 MULT1:  RAR             ; CARRY= 0, SHIFT HIGH ORDER BYTE OF RESULT
  0863 67         232         MOV     H,A     ; MOVE HIGH ORDER RESULT INTO H REG
  0864 C35608     233         JMP     MULT0   ; CONTINUE LOOPING
  0867 D1         234 DONE:   POP     D       ; RESTORE D-E REGS
  0868 C1         235         POP     B       ; RESTORE B-C REGS
  0869 C9         236         RET             ; RETURN TO CALL
                  237 ;
```



COMMENTS: THE PARAMETRIC REGS—BLRLSB AND THE ENABLE REG CAN NOT BE READ FROM THE 7220. THEY MUST BE READ FROM A MEMORY IMAGE IN RAM. SINCE ONLY THE BLRLSB IS USED TO COMPUTE THE BYTE COUNTER, DATA TRANSFERS ARE LIMITED TO 255 PAGES.

Figure 13. BYTCNT

```
ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0        BPK72    PAGE    6


 LOC  OBJ         LINE       SOURCE STATEMENT

                   238 ;*******************************************************************************
                   239 ;
                   240 ; FUNCTION: WRITE
                   241 ; INPUTS:   D-E REGS, STARTING ADDRESS OF DATA IN RAM
                   242 ;           H-L REGS, BYTE COUNTER
                   243 ;           BPK72 STATUS REG
                   244 ; OUTPUTS:  WRITE DATA TO BUBBLE MEMORY
                   245 ; CALLS:    NONE
                   246 ; DESTROYS: A, H, L, F/FS
                   247 ;
                   248 ; DESCRIPTION: TRANSFER DATA FROM RAM TO BUBBLE MEMORY
                   249 ;       THE  D-E REGS  CONTAIN THE STARTING  ADDRESS  IN  RAM OF  DATA
                   250 ;       TO BE WRITTEN  INTO  THE  BUBBLE  MEMORY.  THE  H-L  REGS MUST
                   251 ;       CONTAIN  A BYTE COUNTER  INDICATING  THE  NUMBER OF DATA BYTES
                   252 ;       TO BE TRANSFERRED.  THIS FUNCTION  BEGINS BY ISSUING THE WRITE
                   253 ;       BUBBLE MEMORY  DATA COMMAND FOLLOWED BY POLLING THE STATUS REG
                   254 ;       TO  DETERMINE IF THE 7220 FIFO DATA BUFFER IS READY TO RECEIVE
                   255 ;       DATA.  DATA IS  TRANSFERRED  UNTIL  THE  BYTE  COUNTER OR TIME
                   256 ;       OUT LOOP COUNTER DECREMENTS TO ZERO.  THE PARAMETRIC REGISTERS
                   257 ;       MUST BE LOADED  WITH THE DESIRED VALUES  PRIOR TO CALLING THIS
                   258 ;       FUNCTION.
                   259 ;
 086A D5           260 WRITE:  PUSH    D       ; SAVE D-E REGS
 086B C5           261         PUSH    B       ; SAVE B-C REGS
 086C 01FFFF       262         LXI     B,0FFFFH; INITIALIZE TIME OUT LOOP COUNTER
 086F 3E13         263         MVI     A,13H   ; LOAD A REG= WRITE BUBBLE MEMORY DATA COMMAND
 0871 D3FF         264         OUT     PRTA01  ; WRITE, WRITE BUBBLE MEMORY DATA COMMAND
 0873 0B           265 BUSYWR: DCX     B       ; DECREMENT TIME OUT LOOP COUNTER
 0874 AF           266         XRA     A       ; CLEAR A REG
 0875 B0           267         ORA     B       ; TEST B REG= 00H
 0876 B1           268         ORA     C       ; TEST C REG= 00H
 0877 CA8108       269         JZ      FINSHW  ; IF ZERO, TIME OUT ERROR, JMP FINSHW
 087A DBFF         270         IN      PRTA01  ; READ STATUS REG
 087C 07           271         RLC             ; TEST BUSY BIT= 1
 087D D27308       272         JNC     BUSYWR  ; IF ZERO, CONTINUE POLLING BUSY BIT
                   273                         ; CONTINUED ON NEXT PAGE
                   274 $EJECT
```

```
ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0        BPK72    PAGE    7


 LOC  OBJ        LINE        SOURCE STATEMENT

 0880 DBFF        275 POLLWR: IN      PRTA01  ; READ STATUS REG
 0882 0F          276         RRC             ; TEST FIFO READY BIT= 1
 0883 DA9608      277         JC      WFIFO   ; IF FIFO READY= 1, JMP WFIFO
 0886 DBFF        278         IN      PRTA01  ; READ STATUS REG
 0888 07          279         RLC             ; TEST BUSY BIT= 1
 0889 D2A108      280         JNC     FINSHW  ; IF ZERO, ERROR, JMP FINSHW
 088C 0B          281         DCX     B       ; DECREMENT TIME OUT LOOP COUNTER
 088D AF          282         XRA     A       ; CLEAR A REG
 088E B0          283         ORA     B       ; TEST B REG= 00H
 088F B1          284         ORA     C       ; TEST C REG= 00H
 0890 CAA108      285         JZ      FINSHW  ; IF ZERO, TIME OUT ERROR, JMP FINSHW
 0893 C38008      286         JMP     POLLWR  ; CONTINUE POLLING FIFO READY BIT
 0896 1A          287 WFIFO:  LDAX    D       ; LOAD A REG FROM D-E REG ADDRESS
 0897 D3FE        288         OUT     PRTA00  ; WRITE A REG TO 7220 FIFO DATA BUFFER
 0899 13          289         INX     D       ; INCREMENT D-E REGS TO NEXT ADDRESS IN RAM
 089A 2B          290         DCX     H       ; DECREMENT BYTE COUNTER
 089B AF          291         XRA     A       ; CLEAR A REG
 089C B4          292         ORA     H       ; TEST H REG= 00H
 089D B5          293         ORA     L       ; TEST L REG= 00H
 089E C28008      294         JNZ     POLLWR  ; IF BYTE COUNTER NOT ZERO, JMP POLLWR
 08A1 C1          295 FINSHW: POP     B       ; RESTORE B-C REGS
 08A2 D1          296         POP     D       ; RESTORE D-E REGS
 08A3 C9          297         RET             ; RETURN TO CALL
                  298 ;
                  299 ;
                  300 ;
                  301 $EJECT
```

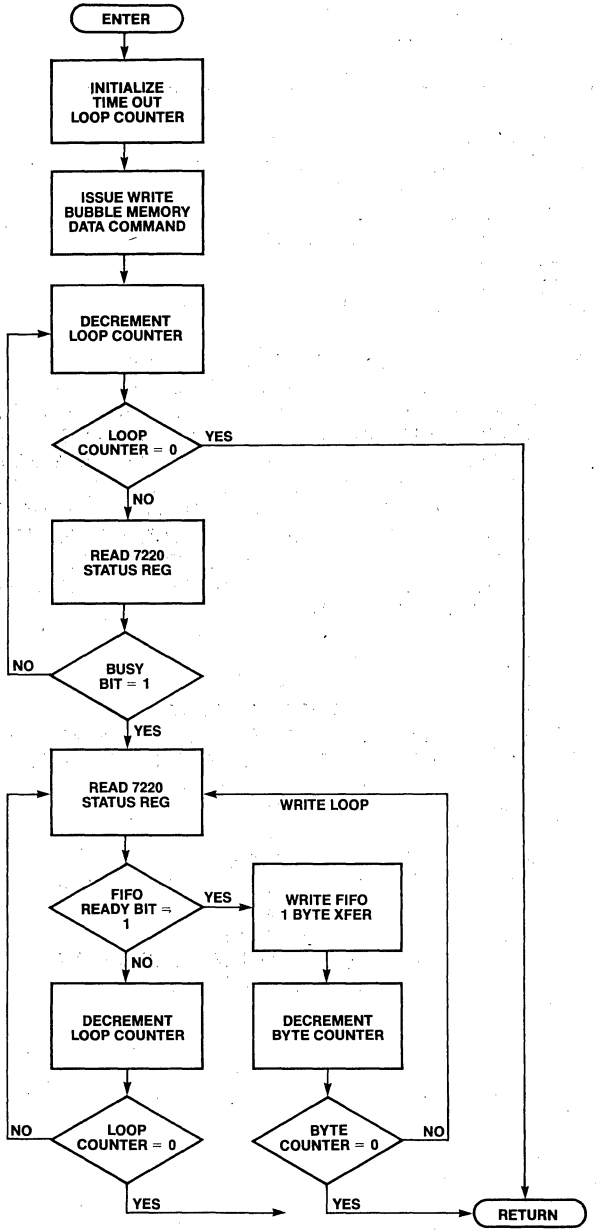


Figure 14. WRITE

```
ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0        BPK72    PAGE    8

 LOC  OBJ        LINE       SOURCE STATEMENT

                  302 ;*******************************************************************************
                  303 ;
                  304 ; FUNCTION:'READ
                  305 ; INPUTS:   D-E REGS, STARTING ADDRESS IN RAM
                  306 ;           H-L REGS, BYTE COUNTER
                  307 ;           BPK72 STATUS REG
                  308 ;           READ DATA FROM BUBBLE MEMORY
                  309 ; OUTPUTS:  WRITE DATA TO RAM
                  310 ; CALLS:    NONE
                  311 ; DESTROYS: A, H, L, F/FS
                  312 ;
                  313 ; DESCRIPTION: TRANSFER DATA FROM BUBBLE MEMORY TO RAM
                  314 ;       THE D-E REGS CONTAIN THE STARTING ADDRESS IN RAM USED TO STORE
                  315 ;       DATA READ FROM THE  BUBBLE MEMORY.  THE H-L  REGS MUST CONTAIN
                  316 ;       A BYTE  COUNTER  INDICATING  THE  NUMBER  OF DATA  BYTES TO BE
                  317 ;       TRANSFERRED.  THIS FUNCTION BEGINS  BY ISSUING THE READ BUBBLE
                  318 ;       MEMORY  DATA  COMMAND  FOLLOWED  BY  POLLING  THE  STATUS  REG
                  319 ;       TO DETERMINE  IF  THE  7220 FIFO  DATA  BUFFER  CONTAINS  DATA
                  320 ;       AVAILABLE FOR READING.  DATA  IS  TRANSFERRED  UNTIL  THE BYTE
                  321 ;       COUNTER  OR  TIME  OUT  LOOP COUNTER  DECREMENTS TO ZERO.  THE
                  322 ;       PARAMETRIC REGS MUST BE LOADED  WITH THE  DESIRED VALUES PRIOR
                  323 ;       TO CALLING THIS FUNCTION.
                  324 ;
 08A4 D5          325 READ:   PUSH    D       ; SAVE D-E REGS
 08A5 C5          326         PUSH    B       ; SAVE B-C REGS
 08A6 01FFFF      327         LXI     B,0FFFFH; INITIALIZE TIME OUT LOOP COUNTER
 08A9 3E12        328         MVI     A,12H   ; LOAD A REG= READ BUBBLE MEMORY DATA COMMAND
 08AB D3FF        329         OUT     PRTA01  ; WRITE, READ BUBBLE MEMORY DATA COMMAND
 08AD 0B          330 BUSYRD: DCX     B       ; DECREMENT TIME OUT LOOP COUNTER
 08AE AF          331         XRA     A       ; CLEAR A REG
 08AF B0          332         ORA     B       ; TEST B REG= 00H
 08B0 B1          333         ORA     C       ; TEST C REG= 00H
 08B1 CADB08      334         JZ      FINSHR  ; IF ZERO, TIME OUT ERROR, JMP FINSHR
 08B4 DBFF        335         IN      PRTA01  ; READ STATUS REG
 08B6 07          336         RLC             ; TEST BUSY BIT= 1
 08B7 D2AD08      337         JNC     BUSYRD  ; IF ZERO, CONTINUE POLLING BUSY BIT
                  338                         ; CONTINUED ON NEXT PAGE
                  339 $EJECT
```

```
ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0        BPK72    PAGE    9


 LOC  OBJ        LINE        SOURCE STATEMENT

 08BA DBFF        340 POLLRD: IN      PRTA01  ; READ STATUS REG
 08BC 0F          341         RRC             ; TEST FIFO READY BIT= 1
 08BD DAD008      342         JC      RFIFO   ; IF FIFO READY= 1, JMP RFIFO
 08C0 DBFF        343         IN      PRTA01  ; READ STATUS REG
 08C2 07          344         RLC             ; TEST BUSY BIT= 1
 08C3 D2DB08      345         JNC     FINSHR  ; IF ZERO, ERROR, JMP FINSHR
 08C6 0B          346         DCX     B       ; DECREMENT TIME OUT LOOP COUNTER
 08C7 AF          347         XRA     A       ; CLEAR A REG
 08C8 B0          348         ORA     B       ; TEST B REG= 00H
 08C9 B1          349         ORA     C       ; TEST C REG= 00H
 08CA CADB08      350         JZ      FINSHR  ; IF ZERO, TIME OUT ERROR, JMP FINSHR
 08CD C3BA08      351         JMP     POLLRD  ; CONTINUE POLLING FIFO READY BIT
 08D0 DBFE        352 RFIFO:  IN      PRTA00  ; LOAD A REG WITH ONE BYTE FROM FIFO DATA BUFFER
 08D2 12          353         STAX    D       ; STORE A REG IN REG D-E ADDRESS
 08D3 13          354         INX     D       ; INCREMENT D-E REGS TO NEXT ADDRESS IN RAM
 08D4 2B          355         DCX     H       ; DECREMENT BYTE COUNTER
 08D5 AF          356         XRA     A       ; CLEAR A REG
 08D6 B4          357         ORA     H       ; TEST H REG= 00H
 08D7 B5          358         ORA     L       ; TEST L REG= 00H
 08D8 C2BA08      359         JNZ     POLLRD  ; IF BYTE COUNTER NOT ZERO, JMP POLLRD
 08DB C1          360 FINSHR: POP     B       ; RESTORE B-C REGS
 08DC D1          361         POP     D       ; RESTORE D-E REGS
 08DD C9          362         RET             ; RETURN TO CALL
                  363 ;
                  364 ;
                  365 ;
                  366 $EJECT
```

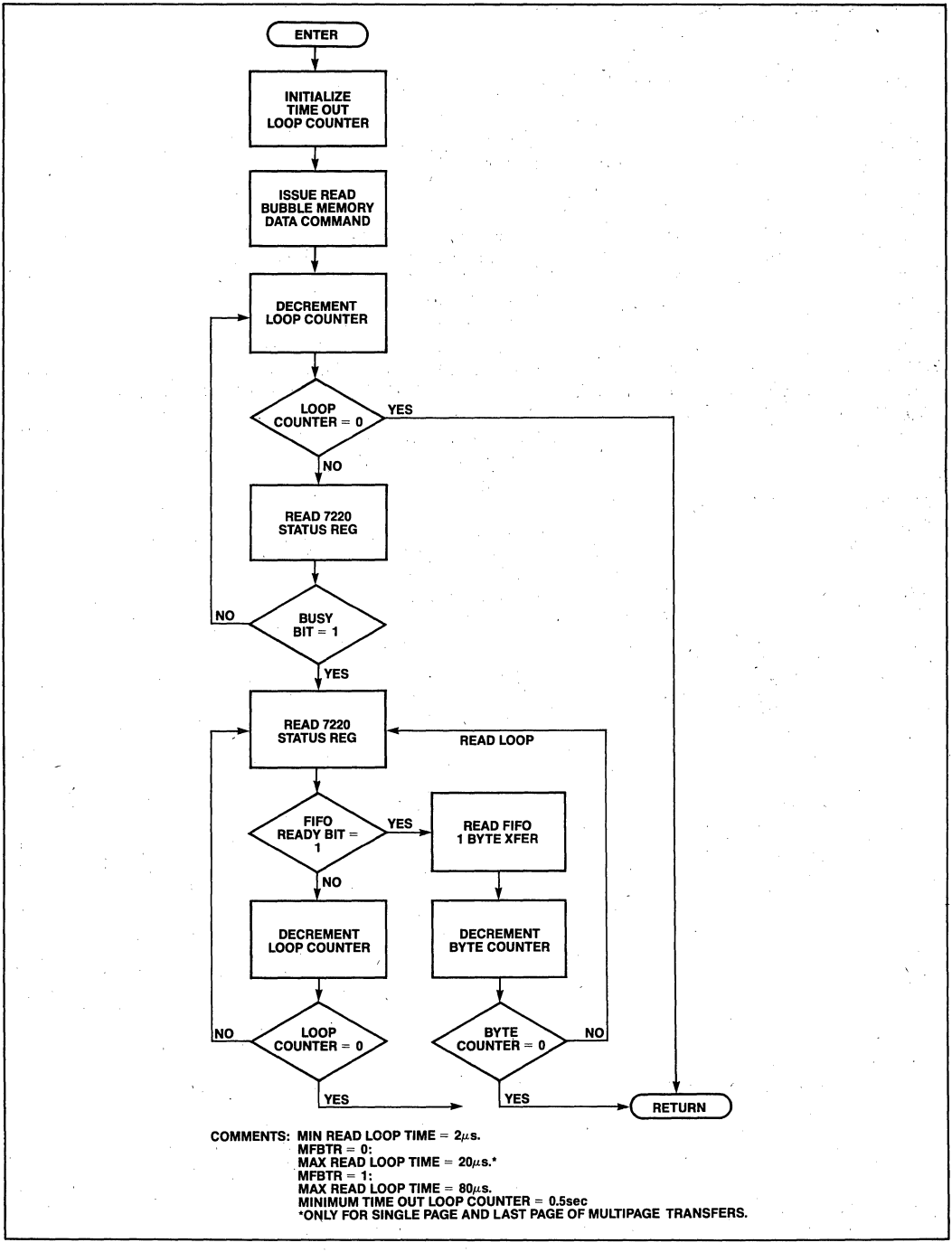


Figure 15. READ

```
ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0        BPK72    PAGE   10


 LOC  OBJ        LINE       SOURCE STATEMENT

                  367 ;*******************************************************************************
                  368 ;
                  369 ; FUNCTION: ABORT
                  370 ; INPUTS:   BPK72 STATUS REG
                  371 ; OUTPUTS:  ISSUE ABORT COMMAND TO BPK72
                  372 ;           A REG= BPK72 STATUS REG
                  373 ; CALLS:    NONE
                  374 ; DESTROYS: A, F/FS
                  375 ;
                  376 ; DESCRIPTION: ABORT PRESENT COMMAND, RESET BPK72
                  377 ;       AN ABORT  COMMAND IS  ISSUED  TO THE BPK72.  AFTER  ISSUING THE
                  378 ;       COMMAND,  THE  BPK72 STATUS REG IS POLLED UNTIL AN OP-COMPLETE,
                  379 ;       40H,  HAS BEEN  READ  OR THE  TIME OUT  LOOP COUNTER DECREMENTS
                  380 ;       TO ZERO.  THE ABORT FUNCTION  RETURNS  THE  VALUE  OF THE BPK72
                  381 ;       STATUS REG TO THE CALLING ROUTINE VIA THE 8085'S A REG.  ONLY A
                  382 ;       STATUS  OF 40H INDICATES A  SUCCESSFUL  EXECUTION  OF THE ABORT
                  383 ;       FUNCTION.
                  384 ;
                  385         PUBLIC  ABORT   ; DECLARE PUBLIC FUNCTION
  08DE D5         386 ABORT:  PUSH    D       ; SAVE D-E REGS
  08DF C5         387         PUSH    B       ; SAVE B-C REGS
  08E0 11FFFF     388         LXI     D,0FFFFH; INITIALIZE TIME OUT LOOP COUNTER
  08E3 0640       389         MVI     B,40H   ; LOAD B REG= 40H, OP-COMPLETE
  08E5 3E19       390         MVI     A,19H   ; LOAD A REG= ABORT COMMAND
  08E7 D3FF       391         OUT     PRTA01  ; WRITE ABORT COMMAND
  08E9 DBFF       392 BUSYA:  IN      PRTA01  ; READ STATUS REG
  08EB 07         393         RLC             ; TEST BUSY BIT= 1
  08EC DAF908     394         JC      POLLA   ; IF BUSY= 1, POLL STATUS REG FOR 40H
  08EF 1B         395         DCX     D       ; DECREMENT TIME OUT LOOP COUNTER
  08F0 AF         396         XRA     A       ; CLEAR A REG
  08F1 B2         397         ORA     D       ; TEST D REG= 00H
  08F2 B3         398         ORA     E       ; TEST E REG= 00H
  08F3 C2E908     399         JNZ     BUSYA   ; IF NOT ZERO, CONTINUE POLLING ABORT COMMAND
  08F6 C30609     400         JMP     RETA    ; TIME OUT ERROR, RETURN
  08F9 DBFF       401 POLLA:  IN      PRTA01  ; READ STATUS REG
  08FB A8         402         XRA     B       ; TEST STATUS= 40H, OP-COMPLETE
  08FC CA0609     403         JZ      RETA    ; IF OP-COMPLETE, JMP RETA
  08FF 1B         404         DCX     D       ; DECREMENT TIME OUT LOOP COUNTER
  0900 AF         405         XRA     A       ; CLEAR A REG
  0901 B2         406         ORA     D       ; TEST D REG= 00H
  0902 B3         407         ORA     E       ; TEST E REG= 00H
  0903 C2F908     408         JNZ     POLLA   ; IF NOT ZERO, CONTINUE POLLING ABORT COMMAND
  0906 C1         409 RETA:   POP     B       ; RESTORE B-C REGS
  0907 D1         410         POP     D       ; RESTORE D-E REGS
  0908 DBFF       411         IN      PRTA01  ; READ STATUS REG
  090A C9         412         RET             ; RETURN TO CALL
                  413 ;
                  414 ;
                  415 ;
                  416 $EJECT
```

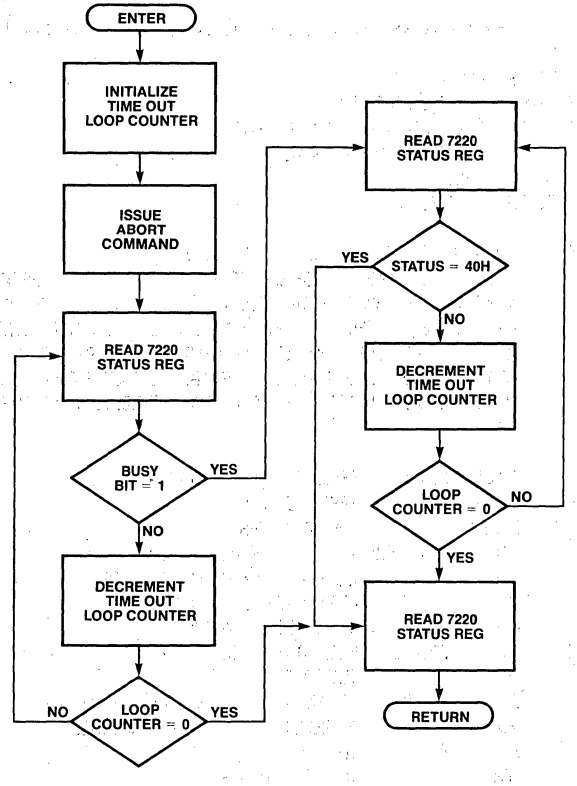


Figure 16. ABORT

```
ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0        BPK72    PAGE   11

 LOC  OBJ        LINE       SOURCE STATEMENT

                  417 ;*******************************************************************************
                  418 ;
                  419 ; FUNCTION: WRBUBL
                  420 ; INPUTS:   B-C REGS, STARTING ADDRESS OF PARAMETRIC REGS IN RAM
                  421 ;           D-E REGS, STARTING ADDRESS OF DATA IN RAM
                  422 ;           BPK72 STATUS REG
                  423 ; OUTPUTS:  WRITE DATA TO BUBBLE MEMORY
                  424 ;           A REG= BPK72 STATUS REG
                  425 ; CALLS:    FIFORS
                  426 ;           INTPAR
                  427 ;           BYTCNT
                  428 ;           WRITE
                  429 ; DESTROYS: A, F/FS
                  430 ;
                  431 ; DESCRIPTION: WRITE BUBBLE MEMORY DATA
                  432 ;       THE B-C  REGS CONTAIN THE ADDRESS  TO THE  FIRST  OF FIVE
                  433 ;       CONTIGUOUS  MEMORY  LOCATIONS IN RAM.  THE DATA ADDRESSED
                  434 ;       BY THE  B-C  REGS IS USED  TO LOAD THE  PARAMETRIC  REGS.
                  435 ;       THE D-E  REGS  CONTAIN  THE STARTING ADDRESS  IN  RAM  OF
                  436 ;       DATA TO BE WRITTEN INTO THE BUBBLE MEMORY. GIVEN THE DATA
                  437 ;       IN RAM  USED TO  LOAD THE PARAMETRIC REGS,  THIS FUNCTION
                  438 ;       WILL  RESET  THE 7220 FIFO,  LOAD  THE  PARAMETRIC  REGS,
                  439 ;       COMPUTE THE BYTE COUNTER, AND COPY THE DATA FROM RAM INTO
                  440 ;       THE BUBBLE MEMORY.  WRBUBL RETURNS THE VALUE OF THE BPK72
                  441 ;       STATUS  REG TO THE CALLING ROUTINE VIA  THE 8085'S A REG.
                  442 ;       ONLY  A  STATUS  OF  40H OR 42H  INDICATES  A  SUCCESSFUL
                  443 ;       EXECUTION OF WRBUBL.
                  444 ;
                  445         PUBLIC  WRBUBL  ; DECLARE PUBLIC FUNCTION
090B E5           446 WRBUBL: PUSH    H       ; SAVE H-L REGS
090C C5           447         PUSH    B       ; SAVE B-C REGS
090D 0640         448         MVI     B,40H   ; LOAD B REG= 40H, OP-COMPLETE
090F CD1308       449         CALL    FIFORS  ; CALL FIFORS, WRITE FIFO RESET COMMAND
0912 A8           450         XRA     B       ; TEST FOR STATUS= 40H, OP-COMPLETE
0913 C23109       451         JNZ     RETWR   ; IF NOT ZERO, FIFO ERROR, JMP RETWR
0916 C1           452         POP     B       ; RESTORE B-C REGS
0917 CD0008       453         CALL    INTPAR  ; CALL INTPAR, LOAD PARAMETRIC REGS
091A CD4008       454         CALL    BYTCNT  ; CALL BYTCNT, COMPUTE BYTE COUNTER
091D CD6A08       455         CALL    WRITE   ; CALL WRITE,  WRITE BUBBLE DATA
0920 C5           456         PUSH    B       ; SAVE B-C REGS
                  457                         ; CONTINUED ON NEXT PAGE
                  458 $EJECT
```

```
ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0        BPK72    PAGE   12


 LOC  OBJ        LINE        SOURCE STATEMENT

 0921 21FFFF      459         LXI     H,0FFFFH; INITIALIZE TIME OUT LOOP COUNTER
 0924 DBFF        460 LOOPWR: IN      PRTA01  ; READ STATUS REG
 0926 07          461         RLC             ; TEST FOR BUSY BIT= 1
 0927 D23109      462         JNC     RETWR   ; IF ZERO, NOT BUSY, JMP RETWR
 092A 2B          463         DCX     H       ; DECREMENT TIME OUT LOOP COUNTER
 092B AF          464         XRA     A       ; CLEAR A REG
 092C B4          465         ORA     H       ; TEST H REG= 00H
 092D B5          466         ORA     L       ; TEST L REG= 00H
 092E C22409      467         JNZ     LOOPWR  ; CONTINUE POLLING STATUS REG
 0931 C1          468 RETWR:  POP     B       ; RESTORE B-C REGS
 0932 E1          469         POP     H       ; RESTORE H-L REGS
 0933 DBFF        470         IN      PRTA01  ; READ STATUS REG
 0935 C9          471         RET             ; RETURN TO CALL
                  472 ;
                  473 ;
                  474 ;
                  475 $EJECT
```

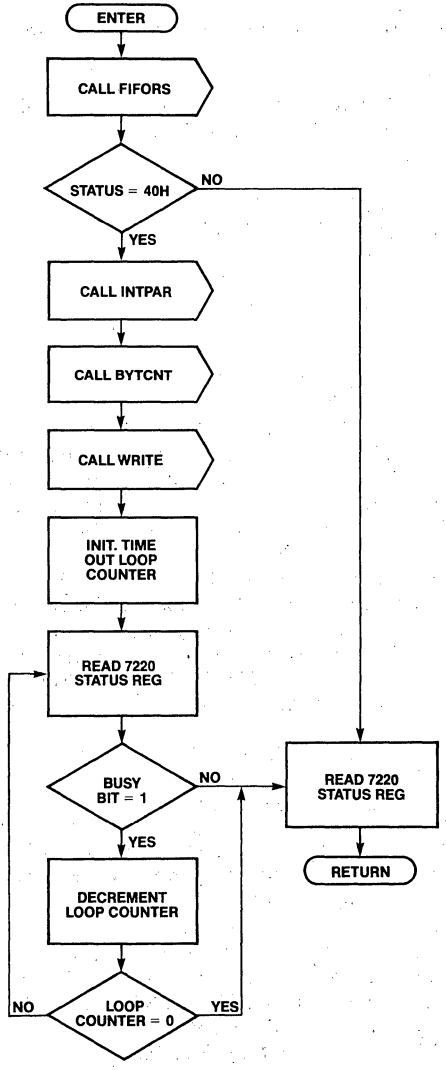


Figure 17. WRBUBL

```
ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0        BPK72    PAGE   13


 LOC  OBJ         LINE       SOURCE STATEMENT

                   476 ;*******************************************************************************
                   477 ;
                   478 ; FUNCTION: RDBUBL
                   479 ; INPUTS:   B-C REGS, STARTING ADDRESS OF PARAMETRIC REGS IN RAM
                   480 ;           D-E REGS, STARTING ADDRESS IN RAM
                   481 ;           BPK72 STATUS REG
                   482 ;           READ DATA FROM BUBBLE MEMORY
                   483 ; OUTPUTS:  WRITE DATA TO RAM
                   484 ;           A REG= BPK72 STATUS REG
                   485 ; CALLS:    FIFORS
                   486 ;           INTPAR
                   487 ;           BYTCNT
                   488 ;           READ
                   489 ; DESTROYS: A, F/FS
                   490 ;
                   491 ; DESCRIPTION: READ BUBBLE MEMORY DATA
                   492 ;       THE B-C REGS CONTAIN THE ADDRESS TO THE FIRST OF FIVE
                   493 ;       CONTIGUOUS MEMORY LOCATIONS IN RAM. THE DATA ADDRESSED
                   494 ;       BY THE B-C REGS IS USED TO LOAD THE PARAMETRIC REGS. THE D-E
                   495 ;       REGS CONTAIN THE STARTING ADDRESS IN RAM USED TO STORE
                   496 ;       DATA READ FROM THE BUBBLE MEMORY. GIVEN THE DATA IN RAM
                   497 ;       USED TO LOAD THE PARAMETRIC REGS, THIS FUNCTION WILL RESET
                   498 ;       THE 7220 FIFO, LOAD THE PARAMETRIC REGS, COMPUTE THE
                   499 ;       BYTE COUNTER, AND COPY THE DATA FROM THE BUBBLE MEMORY INTO
                   500 ;       RAM. RDBUBL RETURNS THE VALUE OF THE BPK72 STATUS REGISTER
                   501 ;       TO THE CALLING ROUTINE VIA THE 8085'S A REG. ONLY A STATUS
                   502 ;       OF 40H OR 48H WITH ERROR CORRECTION INDICATES A SUCCESSFUL
                   503 ;       EXECUTION OF RDBUBL.
                   504 ;
                   505         PUBLIC  RDBUBL  ; DECLARE PUBLIC FUNCTION
 0936 E5           506 RDBUBL: PUSH    H       ; SAVE H-L REGS
 0937 C5           507         PUSH    B       ; SAVE B-C REGS
 0938 0640         508         MVI     B,40H   ; LOAD B REG= 40H, OP-COMPLETE
 093A CD1308       509         CALL    FIFORS  ; CALL FIFORS, WRITE FIFO RESET COMMAND
 093D A8           510         XRA     B       ; TEST FOR STATUS= 40H, OP-COMPLETE
 093E C25C09       511         JNZ     RETRD   ; IF NOT ZERO, FIFO ERROR, JMP RETRD
 0941 C1           512         POP     B       ; RESTORE B-C REGS
 0942 CD0008       513         CALL    INTPAR  ; CALL INTPAR, LOAD PARAMETRIC REGS
 0945 CD4008       514         CALL    BYTCNT  ; CALL BYTCNT, COMPUTE BYTE COUNTER
 0948 CDA408       515         CALL    READ    ; CALL READ, READ BUBBLE DATA
 094B C5           516         PUSH    B       ; SAVE B-C REGS
                   517                         ; CONTINUED ON NEXT PAGE
                   518 $EJECT
```

```
ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0        BPK72    PAGE   14


 LOC  OBJ        LINE        SOURCE STATEMENT

 094C 21FFFF      519         LXI     H,0FFFFH; INITIALIZE TIME OUT LOOP COUNTER
 094F DBFF        520 LOOPRD: IN      PRTA01  ; READ STATUS REG
 0951 07          521         RLC             ; TEST FOR BUSY BIT= 1
 0952 D25C09      522         JNC     RETRD   ; IF ZERO, NOT BUSY, JMP RETRD
 0955 2B          523         DCX     H       ; DECREMENT TIME OUT LOOP COUNTER
 0956 AF          524         XRA     A       ; CLEAR A REG
 0957 B4          525         ORA     H       ; TEST H REG= 00H
 0958 B5          526         ORA     L       ; TEST L REG= 00H
 0959 C24F09      527         JNZ     LOOPRD  ; CONTINUE POLLING STATUS REG
 095C C1          528 RETRD:  POP     B       ; RESTORE B-C REGS
 095D E1          529         POP     H       ; RESTORE H-L REGS
 095E DBFF        530         IN      PRTA01  ; READ STATUS REG
 0960 C9          531         RET             ; RETURN TO CALL
                  532 ;
                  533 ;
                  534 ;
                  535 $EJECT
```

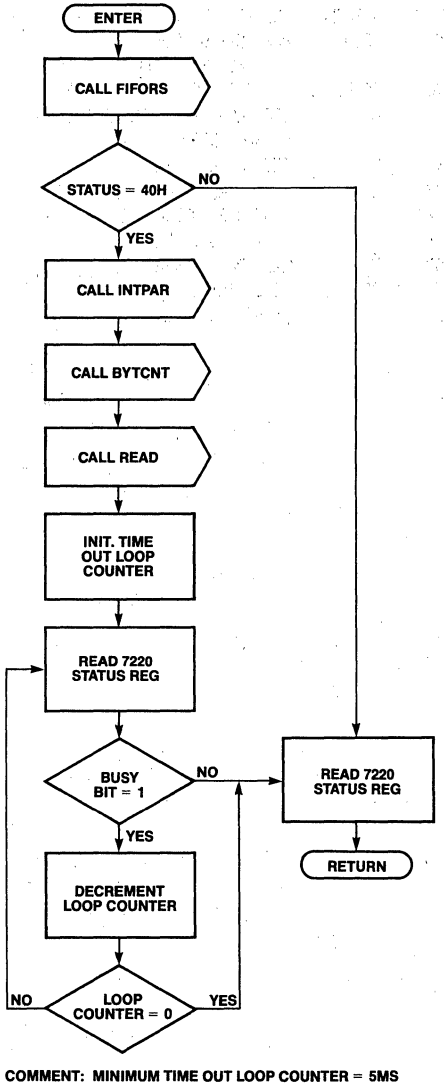


Figure 18. RDBUBL

```
ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0        BPK72    PAGE   15


 LOC  OBJ        LINE       SOURCE STATEMENT

                  536 ;*******************************************************************************
                  537 ;
                  538 ; FUNCTION: INBUBL
                  539 ; INPUTS:   B-C REGS, STARTING ADDRESS OF PARAMETRIC REGS IN RAM
                  540 ;           BPK72 STATUS REG
                  541 ; OUTPUTS:  A REG= BPK72 STATUS REG
                  542 ; CALLS:    ABORT
                  543 ;           INTPAR
                  544 ; DESTROYS: A, F/FS
                  545 ;
                  546 ; DESCRIPTION: INITIALIZE THE BPK72
                  547 ;       THE B-C REGS CONTAIN THE ADDRESS TO THE FIRST OF  FIVE CONTIGUOUS
                  548 ;       MEMORY LOCATIONS IN RAM.  THE  DATA ADDRESSED  BY  THE  B-C  REGS
                  549 ;       IS  USED TO  LOAD THE  PARAMETRIC REGS.  THIS FUNCTION WILL WRITE
                  550 ;       THE  PARAMETRIC  REGS FOLLOWED  BY  ISSUING  A  BUBBLE  MEMORY
                  551 ;       INITIALIZATION  COMMAND.  AFTER  ISSUING  THE COMMAND,  THE BPK72
                  552 ;       STATUS REG IS POLLED UNTIL AN OP-COMPLETE,  40H,  IS  READ OR THE
                  553 ;       TIME  OUT  LOOP  COUNTER  DECREMENTS TO ZERO.  THIS  COMMAND MUST
                  554 ;       PRECEED ALL  OTHER  COMMANDS AFTER POWERING UP THE BPK72.  INBUBL
                  555 ;       RETURNS THE VALUE OF THE BPK72 STATUS  REG TO THE CALLING ROUTINE
                  556 ;       VIA THE 8085'S A REG.  ONLY A STATUS OF 40H INDICATES A SUCCESSFUL
                  557 ;       EXECUTION OF INBUBL.
                  558 ;
                  559         PUBLIC  INBUBL  ; DECLARE PUBLIC FUNCTION
0961 D5           560 INBUBL: PUSH    D       ; SAVE D-E REGS
0962 C5           561         PUSH    B       ; SAVE B-C REGS
0963 0640         562         MVI     B,40H   ; LOAD B REG= 40H, OP-COMPLETE
0965 CDDE08       563         CALL    ABORT   ; CALL ABORT COMMAND
0968 A8           564         XRA     B       ; TEST STATUS= 40H, OP-COMPLETE
0969 C29709       565         JNZ     RETIN   ; IF ZERO, OP-COMPLETE, CONTINUE
096C C1           566         POP     B       ; PARAMETRIC REGS STARTING ADDRESS IN REG B
096D CD0008       567         CALL    INTPAR  ; CALL INTPAR, LOAD PARAMETRIC REGS
0970 C5           568         PUSH    B       ; SAVE B-C REGS
0971 0640         569         MVI     B,40H   ; LOAD B REG= 40H, OP-COMPLETE
0973 11FFFF       570         LXI     D,0FFFFH; INITIALIZE TIME OUT LOOP COUNTER
0976 3E11         571         MVI     A,11H   ; LOAD A REG= INITIALIZE COMMAND
0978 D3FF         572         OUT     PRTA01  ; WRITE INITIALIZE COMMAND
                  573                         ; CONTINUED ON NEXT PAGE
                  574 $EJECT
```

```
ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0        BPK72    PAGE   16

 LOC  OBJ        LINE        SOURCE STATEMENT

 097A DBFF        575 BUSYIN: IN      PRTA01  ; READ STATUS REG
 097C 07          576         RLC             ; TEST BUSY BIT= 1
 097D DA8A09      577         JC      POLLIN  ; IF BUSY= 1, POLL STATUS REG FOR 40H
 0980 1B          578         DCX     D       ; DECREMENT TIME OUT LOOP COUNTER
 0981 AF          579         XRA     A       ; CLEAR A REG
 0982 B2          580         ORA     D       ; TEST D REG= 00H
 0983 B3          581         ORA     E       ; TEST E REG= 00H
 0984 C27A09      582         JNZ     BUSYIN  ; IF NOT ZERO, CONTINUE POLLING THE INITIALIZE COMMAND
 0987 C39709      583         JMP     RETIN   ; TIME OUT ERROR, RETURN
 098A DBFF        584 POLLIN: IN      PRTA01  ; READ STATUS REG
 098C A8          585         XRA     B       ; TEST STATUS= 40H, OP-COMPLETE
 098D CA9709      586         JZ      RETIN   ; IF OP-COMPLETE, JMP RETIN
 0990 1B          587         DCX     D       ; DECREMENT TIME OUT LOOP COUNTER
 0991 AF          588         XRA     A       ; CLEAR A REG
 0992 B2          589         ORA     D       ; TEST D REG= 00H
 0993 B3          590         ORA     E       ; TEST E REG= 00H
 0994 C28A09      591         JNZ     POLLIN  ; IF NOT ZERO, CONTINUE POLLING INITIALIZE COMMAND
 0997 C1          592 RETIN:  POP     B       ; RESTORE B-C REGS
 0998 D1          593         POP     D       ; RESTORE D-E REGS
 0999 DBFF        594         IN      PRTA01  ; READ STATUS REG
 099B C9          595         RET             ; RETURN TO CALL
                  596 ;
                  597 ;
                  598 $EJECT
```

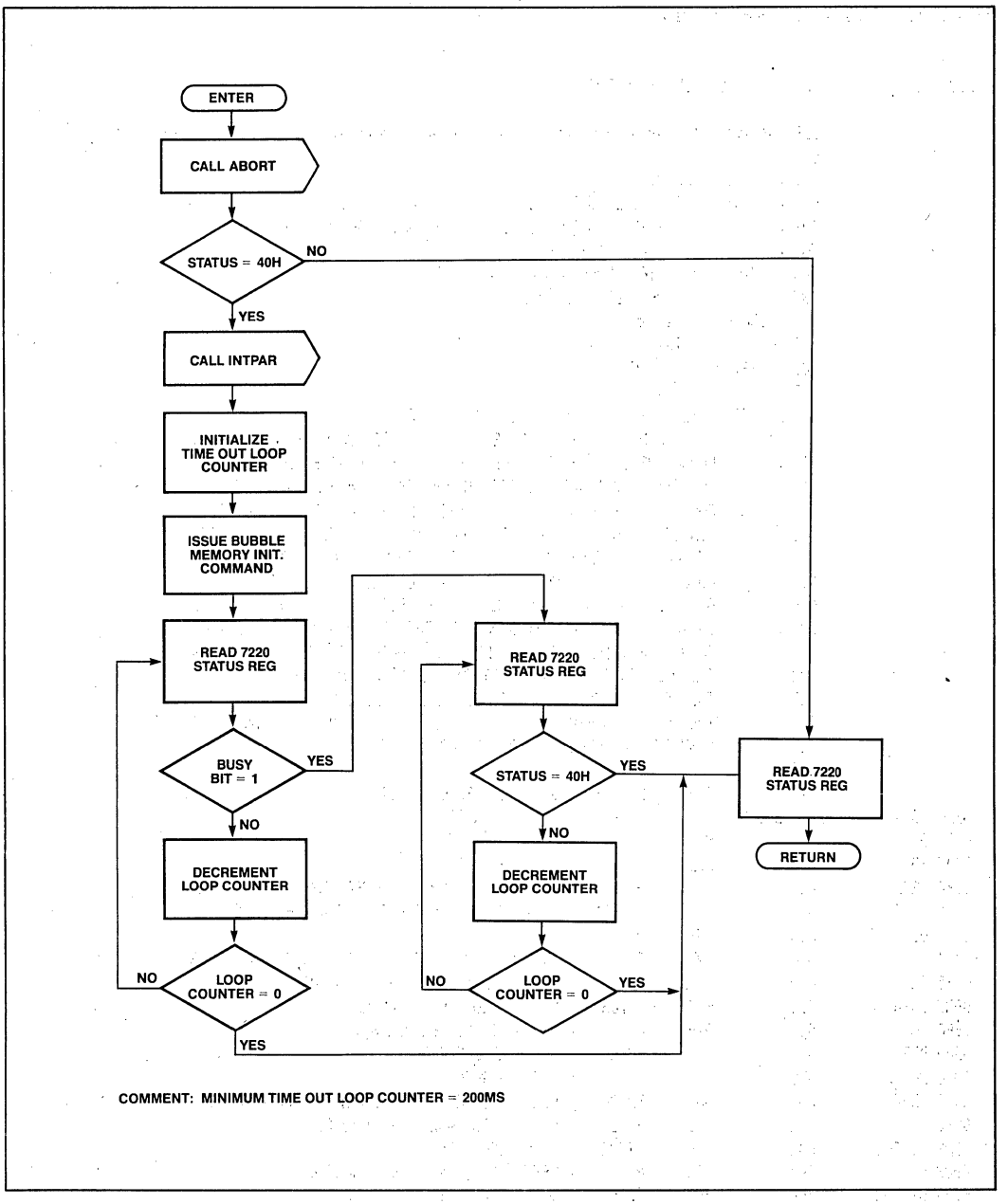


Figure 19. INBUBL

```
ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0        BPK72    PAGE   17


 LOC  OBJ         LINE       SOURCE STATEMENT

                   599 ;*******************************************************************************
                   600 ;
                   601 ; FUNCTION: BOOTUP
                   602 ; INPUTS:   B-C REGS, STARTING ADDRESS OF PARAMETRIC REGS IN RAM
                   603 ;           D-E REGS, STARTING ADDRESS OF BOOT LOOP CODE IN RAM
                   604 ;           BPK72 STATUS REG
                   605 ; OUTPUTS:  WRITE BUBBLE MEMORY BOOT LOOP
                   606 ;           A REG= BPK72 STATUS REG
                   607 ; CALLS:    FIFORS
                   608 ;           INTPAR
                   609 ; DESTROYS: A, F/FS
                   610 ;
                   611 ; DESCRIPTION: WRITE BUBBLE MEMORY BOOT LOOP
                   612 ;       THIS  FUNCTION WILL  WRITE THE  BOOT LOOP CODE  INTO THE 7110
                   613 ;       BUBBLE MEMORY.  THE D-E  REGS PROVIDE THE ADDRESS TO THE FIRST
                   614 ;       OF FORTY  CONTIGUOUS BYTES  IN RAM THAT CONTAIN THE BOOT LOOP
                   615 ;       CODE.  THE B-C REGS  CONTAIN  THE ADDRESS TO THE FIRST OF FIVE
                   616 ;       CONTIGUOUS MEMORY  LOCATIONS ALSO IN RAM.  THE DATA ADDRESSED
                   617 ;       BY THE  B-C  REGS  IS  USED  TO  LOAD  THE  PARAMETRIC  REGS.
                   618 ;       NOTE   THAT  THE  PARAMETRIC  ENABLE  REG  WRITE   BOOT  LOOP
                   619 ;       BIT  IS  AUTOMATICALLY  SET  AND A  FORTY-FIRST BYTE  OF ZERO
                   620 ;       IS WRITTEN  TO THE  FIFO DATA BUFFER TO AVOID A TIMING ERROR.
                   621 ;       BEFORE  A RETURN IS EXECUTED,  THE PARAMETRIC  ENABLE  REG IS
                   622 ;       RESTORED TO ITS VALUE PRIOR TO CALLING BOOTUP.  BOOTUP RETURNS
                   623 ;       THE VALUE OF THE BPK72  STATUS REG TO THE CALLING ROUTINE VIA
                   624 ;       THE 8085'S  A REG.  ONLY  A STATUS  OF  40H OR 42H  INDICATES
                   625 ;       A SUCCESSFUL EXECUTION OF BOOTUP.
                   626 ;
                   627         PUBLIC  BOOTUP  ; DECLARE PUBLIC FUNCTION
 099C C5           628 BOOTUP: PUSH    B       ; SAVE B-C REGS
 099D D5           629         PUSH    D       ; SAVE D-E REGS
 099E E5           630         PUSH    H       ; SAVE H-L REGS
 099F 3E0D         631         MVI     A,0DH   ; LOAD A REG= 0DH, 7220 RAC ENABLE REG ADDRESS
 09A1 D3FF         632         OUT     PRTA01  ; WRITE 7220 RAC WITH ENABLE REG ADDRESS
 09A3 03           633         INX     B       ;
 09A4 03           634         INX     B       ; INCREMENT B-C REGS TO ENABLE REG RAM ADDRESS
 09A5 0A           635         LDAX    B       ; LOAD A REG= ENABLE REG FROM RAM
 09A6 0610         636         MVI     B,10H   ; LOAD B REG= BOOT LOOP ENABLE MASK
 09A8 B0           637         ORA     B       ; SET BOOT LOOP ENABLE BIT
 09A9 D3FE         638         OUT     PRTA00  ; WRITE ENABLE REG
 09AB AF           639         XRA     A       ; CLEAR A REG
 09AC D3FF         640         OUT     PRTA01  ; LOAD 7220 RAC WITH FIFO DATA BUFFER ADDRESS
 09AE 0640         641         MVI     B,40H   ; LOAD B REG= 40H, OP-COMPLETE
 09B0 CD1308       642         CALL    FIFORS  ; CALL FIFO RESET
 09B3 A8           643         XRA     B       ; TEST STATUS= 40H, OP-COMPLETE
 09B4 C2230A       644         JNZ     RETBT   ; IF NOT ZERO, ERROR, JMP RETBT
                   645                         ; CONTINUED ON NEXT PAGE
                   646 $EJECT
```

```
ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0        BPK72    PAGE   18

 LOC  OBJ        LINE        SOURCE STATEMENT

 09B7 0E28        647         MVI     C,28H   ; LOAD C REG= 28H, BYTE COUNTER= 40 DECIMAL
 09B9 3EFF        648         MVI     A,0FFH  ; LOAD A REG= FFH
 09BB D3FE        649 ALLFFS: OUT     PRTA00  ; WRITE A REG INTO FIFO DATA BUFFER
 09BD 0D          650         DCR     C       ; DECREMENT BYTE COUNTER
 09BE C2BB09      651         JNZ     ALLFFS  ; IF BYTE COUNTER= ZERO, CONTINUE
 09C1 21FFFF      652         LXI     H,0FFFFH; INITIALIZE TIME OUT LOOP COUNTER
 09C4 3E16        653         MVI     A,16H   ; LOAD A REG= WRITE BOOT LOOP REG COMMAND
 09C6 D3FF        654         OUT     PRTA01  ; WRITE, WRITE BOOT LOOP REG COMMAND
 09C8 DBFF        655 BUSYB:  IN      PRTA01  ; READ STATUS REG
 09CA 07          656         RLC             ; TEST BUSY BIT= 1
 09CB DAD809      657         JC      POLLBR  ; IF BUSY= 1, POLL STATUS REG FOR 40H
 09CE 2B          658         DCX     H       ; DECREMENT TIME OUT LOOP COUNTER
 09CF AF          659         XRA     A       ; CLEAR A REG
 09D0 B4          660         ORA     H       ; TEST H REG= 00H
 09D1 B5          661         ORA     L       ; TEST L REG= 00H
 09D2 C2C809      662         JNZ     BUSYB   ; IF NOT ZERO, CONTINUE POLLING WRBLRS COMMAND
 09D5 C3230A      663         JMP     RETBT   ; TIME OUT ERROR, RETURN
 09D8 DBFF        664 POLLBR: IN      PRTA01  ; READ STATUS REG
 09DA A8          665         XRA     B       ; TEST STATUS= 40H
 09DB CAE809      666         JZ      CONT    ; IF ZERO, CONTINUE, OP-COMPLETE
 09DE 2B          667         DCX     H       ; DECREMENT TIME OUT LOOP COUNTER
 09DF AF          668         XRA     A       ; CLEAR A REG
 09E0 B4          669         ORA     H       ; TEST H REG= 00H
 09E1 B5          670         ORA     L       ; TEST L REG= 00H
 09E2 CA230A      671         JZ      RETBT   ; IF ZERO, TIME OUT, ERROR
 09E5 C3D809      672         JMP     POLLBR  ; CONTINUE POLLING WRBLRS COMMAND
                  673                         ; CONTINUED ON NEXT PAGE
                  674 $EJECT
```

```
ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0        BPK72    PAGE   19


 LOC  OBJ        LINE        SOURCE STATEMENT

 09E8 CD1308      675 CONT:   CALL    FIFORS  ; CALL FIFO RESET
 09EB A8          676         XRA     B       ; TEST STATUS= 40H
 09EC C2230A      677         JNZ     RETBT   ; IF NOT ZERO, ERROR, JMP RETBT
 09EF 0E28        678         MVI     C,28H   ; LOAD C REG= 28H, BYTE COUNTER= 40 DECIMAL
 09F1 1A          679 BLCODE: LDAX    D       ; LOAD A REG FROM D REG ADDRESS
 09F2 13          680         INX     D       ; INCREMENT D REG TO THE NEXT ADDRESS
 09F3 D3FE        681         OUT     PRTA00  ; WRITE BOOT LOOP CODE INTO FIFO DATA BUFFER
 09F5 0D          682         DCR     C       ; DECREMENT BYTE COUNTER
 09F6 C2F109      683         JNZ     BLCODE  ; IF NOT ZERO, JMP BLCODE
 09F9 AF          684         XRA     A       ; CLEAR A REG
 09FA D3FE        685         OUT     PRTA00  ; WRITE 41ST BYTE OF ZERO INTO FIFO DATA BUFFER
 09FC 21FFFF      686         LXI     H,0FFFFH; LOAD TIME OUT LOOP COUNTER
 09FF 0EFD        687         MVI     C,0FDH  ; MASK, MASK OUT PARITY BIT
 0A01 3E17        688         MVI     A,17H   ; LOAD A REG= WRITE BOOT LOOP COMMAND
 0A03 D3FF        689         OUT     PRTA01  ; WRITE, WRITE BOOT LOOP COMMAND
 0A05 DBFF        690 BUSYBL: IN      PRTA01  ; READ STATUS REG
 0A07 07          691         RLC             ; TEST BUSY BIT= 1
 0A08 DA150A      692         JC      POLLBL  ; IF BUSY=1, POLL STATUS REG FOR OP-COMPLETE
 0A0B 2B          693         DCX     H       ; DECREMENT TIME OUT LOOP COUNTER
 0A0C AF          694         XRA     A       ; CLEAR A REG
 0A0D B4          695         ORA     H       ; TEST H REG= 00H
 0A0E B5          696         ORA     L       ; TEST L REG= 00H
 0A0F C2050A      697         JNZ     BUSYBL  ; IF NOT ZERO, CONTINUE POLLING THE WRBL COMMAND
 0A12 C3230A      698         JMP     RETBT   ; TIME OUT ERROR, RETURN
 0A15 DBFF        699 POLLBL: IN      PRTA01  ; READ STATUS REG
 0A17 A1          700         ANA     C       ; RESET BIT 1, PARITY BIT
 0A18 A8          701         XRA     B       ; TEST STATUS= 40H OR 42H, OP-COMPLETE
 0A19 CA230A      702         JZ      RETBT   ; IF ZERO, CONTINUE, OP-COMPLETE
 0A1C 2B          703         DCX     H       ; DECREMENT TIME OUT LOOP COUNTER
 0A1D AF          704         XRA     A       ; CLEAR A REG
 0A1E B4          705         ORA     H       ; TEST H REG= 00H
 0A1F B5          706         ORA     L       ; TEST L REG= 00H
 0A20 C2150A      707         JNZ     POLLBL  ; CONTINUE POLLING WRITE BOOT LOOP COMMAND
 0A23 E1          708 RETBT:  POP     H       ; RESTORE H-L REGS
 0A24 D1          709         POP     D       ; RESTORE D-E REGS
 0A25 C1          710         POP     B       ; RESTORE B-C REGS
 0A26 CD0008      711         CALL    INTPAR  ; CALL INTPAR, LOAD THE PARAMETRIC REGS
 0A29 DBFF        712         IN      PRTA01  ; READ STATUS REG
 0A2B C9          713         RET             ;
                  714 ;
                  715 ;
                  716 ;
                  717 $EJECT
```

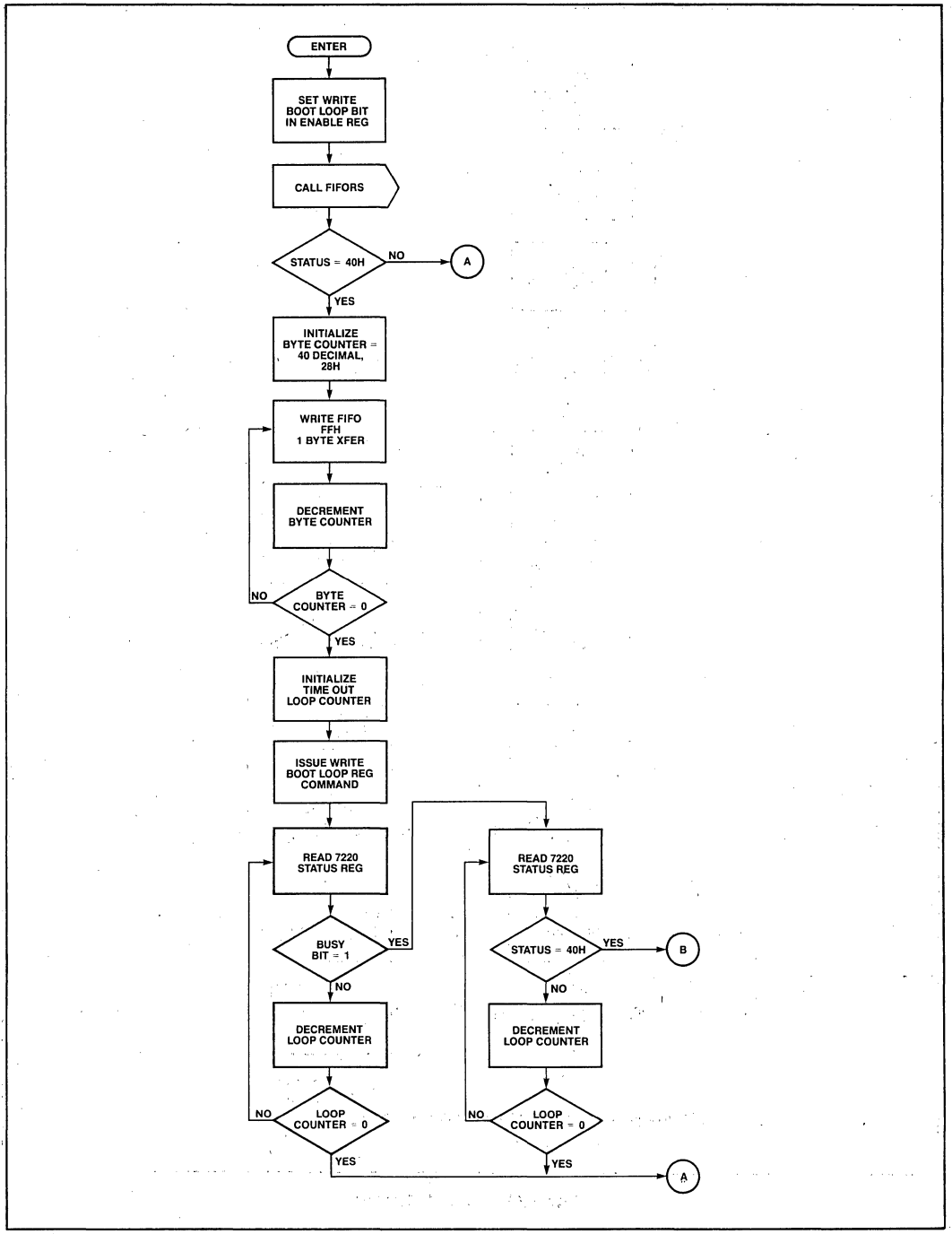


Figure 20. BOOTUP

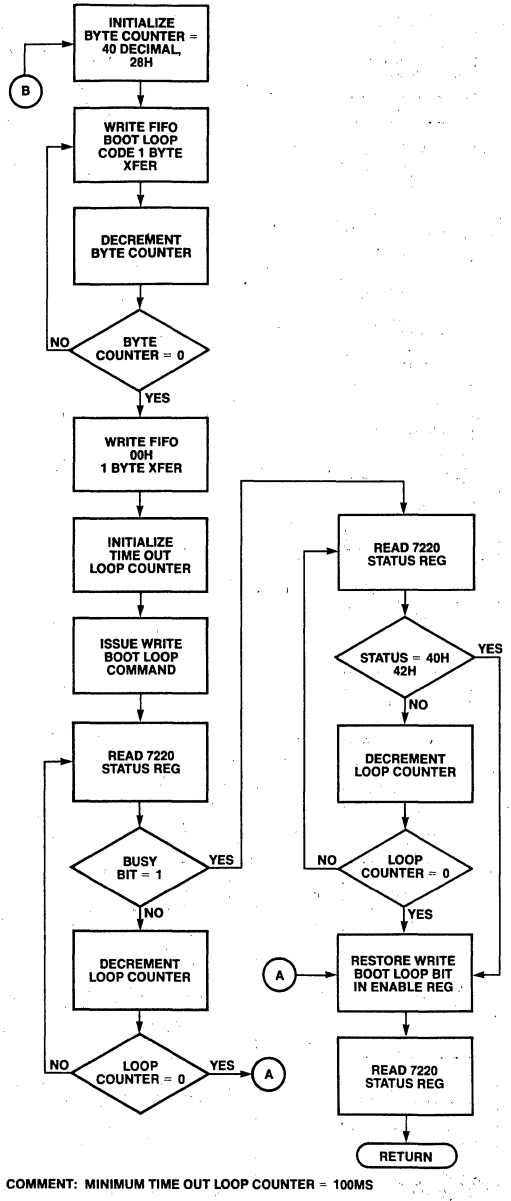


Figure 20 (Con't). BOOTUP

```
ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0        BPK72    PAGE   20


 LOC  OBJ        LINE       SOURCE STATEMENT

                  718 ;*******************************************************************************
                  719 ;
                  720 ; FUNCTION: RDBOOT
                  721 ; INPUTS:   D-E REGS, STARTING ADDRESS IN RAM
                  722 ;           BPK72 STATUS REG
                  723 ;           READ BUBBLE MEMORY BOOT LOOP
                  724 ; OUTPUTS:  COPY BUBBLE MEMORY BOOT LOOP TO RAM
                  725 ;           A REG= BPK72 STATUS REG
                  726 ; CALLS:    FIFORS
                  727 ; DESTROYS: A, F/FS
                  728 ;
                  729 ; DESCRIPTION: READ BUBBLE MEMORY BOOT LOOP
                  730 ;       THE  D-E REGS CONTAIN THE STARTING  ADDRESS  TO THE  FIRST OF 40
                  731 ;       CONTIGUOUS  MEMORY  LOCATIONS  IN  RAM  THAT WILL BE LOADED WITH
                  732 ;       A COPY OF THE BOOT LOOP CODE.  RDBOOT  RETURNS  THE VALUE OF THE
                  733 ;       BPK72  STATUS  REG TO  THE CALLING ROUTINE VIA THE 8085'S A REG.
                  734 ;       ONLY A STATUS OF 40H INDICATES A SUCCESSFUL EXECUTION OF RDBOOT.
                  735 ;
                  736         PUBLIC  RDBOOT  ; DECLARE PUBLIC FUNCTION
 0A2C C5          737 RDBOOT: PUSH    B       ; SAVE B-C REGS
 0A2D D5          738         PUSH    D       ; SAVE D-E REGS
 0A2E E5          739         PUSH    H       ; SAVE H-L REGS
 0A2F 0640        740         MVI     B,40H   ; LOAD B REG= 40H, OP-COMPLETE
 0A31 0E28        741         MVI     C,28H   ; LOAD C REG= 28H, BYTE COUNTER= 40 DECIMAL
 0A33 CD1308      742         CALL    FIFORS  ; CALL FIFO RESET
 0A36 A8          743         XRA     B       ; TEST STATUS= 40H, OP-COMPLETE
 0A37 C26A0A      744         JNZ     RETRDB  ; IF NOT ZERO, ERROR, JMP RETRDB
 0A3A 04          745         INR     B       ; B REG= 41H, OP-COMPLETE, FIFO FULL
 0A3B 21FFFF      746         LXI     H,0FFFFH; INITIALIZE TIME OUT LOOP COUNTER
 0A3E 3E1B        747         MVI     A,1BH   ; LOAD A REG= READ BOOT LOOP COMMAND
 0A40 D3FF        748         OUT     PRTA01  ; WRITE, READ BOOT LOOP COMMAND
 0A42 DBFF        749 BUSYRB: IN      PRTA01  ; READ STATUS REG
 0A44 07          750         RLC             ; TEST BUSY BIT= 1
 0A45 DA520A      751         JC      BTLPRD  ; IF BUSY= 1, POLL STATUS REG FOR 41H
 0A48 2B          752         DCX     H       ; DECREMENT TIME OUT LOOP COUNTER
 0A49 AF          753         XRA     A       ; CLEAR A REG
 0A4A B4          754         ORA     H       ; TEST H REG= 00H
 0A4B B5          755         ORA     L       ; TEST L REG= 00H
 0A4C C2420A      756         JNZ     BUSYRB  ; IF NOT ZERO, CONTINUE POLLING RDBL COMMAND
 0A4F C36A0A      757         JMP     RETRDB  ; TIME OUT ERROR, RETURN
                  758                         ; CONTINUED ON NEXT PAGE
                  759 $EJECT
```

```
ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0        BPK72   PAGE  21


 LOC  OBJ        LINE       SOURCE STATEMENT

 0A52 DBFF        760 BTLPRD: IN      PRTA01  ; READ STATUS REG
 0A54 A8          761         XRA     B       ; TEST STATUS= 41H, OP-COMPLETE, FIFO FULL
 0A55 CA620A      762         JZ      FIFORD  ; IF ZERO, JMP TO FIFO READ
 0A58 2B          763         DCX     H       ; DECREMENT TIME OUT LOOP COUNTER
 0A59 AF          764         XRA     A       ; CLEAR A REG
 0A5A B4          765         ORA     H       ; TEST H REG= 00H
 0A5B B5          766         ORA     L       ; TEST L REG= 00H
 0A5C CA6A0A      767         JZ      RETRDB  ; IF ZERO, TIME OUT, ERROR
 0A5F C3520A      768         JMP     BTLPRD  ; CONTINUE POLLING RDBL COMMAND
 0A62 DBFE        769 FIFORD: IN      PRTA00  ; READ FIFO DATA BUFFER
 0A64 12          770         STAX    D       ; WRITE RAM AT ADDRESS IN D REG
 0A65 13          771         INX     D       ; INCREMENT D REG TO NEXT RAM ADDRESS
 0A66 0D          772         DCR     C       ; DECREMENT BYTE COUNTER
 0A67 C2620A      773         JNZ     FIFORD  ; IF NOT ZERO, JMP FIFO READ
 0A6A DBFF        774 RETRDB: IN      PRTA01  ; READ STATUS REG
 0A6C E1          775         POP     H       ; RESTORE H-L REGS
 0A6D D1          776         POP     D       ; RESTORE D-E REGS
 0A6E C1          777         POP     B       ; RESTORE B-C REGS
 0A6F C9          778         RET             ; RETURN TO CALL
                  779 ;
                  780 $EJECT
```

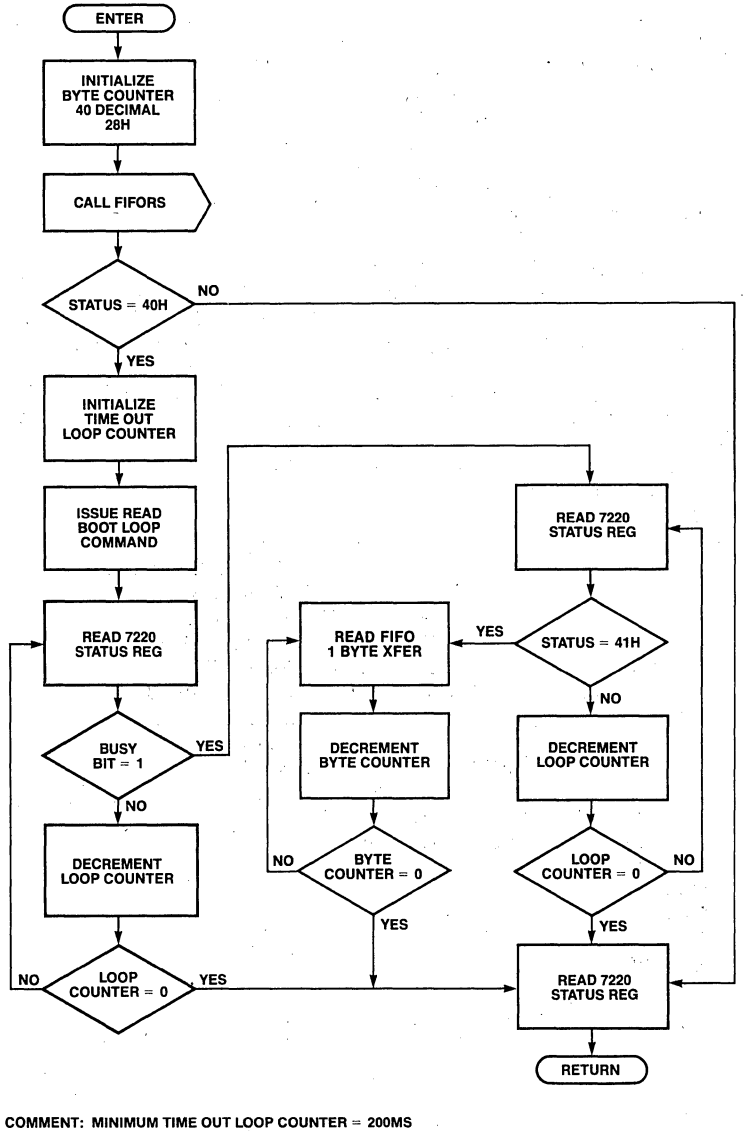


Figure 21. RDBOOT

```
ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0        BPK72    PAGE   22


 LOC  OBJ        LINE       SOURCE STATEMENT

                  781 ;*************************************************************************
                  782 ;
                  783 ; FUNCTION: WRFIFO
                  784 ; INPUTS:   D-E REGS, STARTING ADDRESS OF DATA IN RAM
                  785 ;           BPK72 STATUS REG
                  786 ; OUTPUTS:  WRITE 40 BYTES IN THE BPK72 FIFO DATA BUFFER
                  787 ;           A REG= BPK72 STATUS REG
                  788 ; CALLS:    FIFORS
                  789 ; DESTROYS: A, F/FS
                  790 ;
                  791 ; DESCRIPTION: WRITE 7220 FIFO DATA BUFFER
                  792 ;       THE  D-E REGS PROVIDE THE ADDRESS TO THE FIRST OF 40 CONTIGUOUS
                  793 ;       BYTES IN RAM THAT CONTAIN DATA TO BE LOADED INTO THE BPK72 FIFO
                  794 ;       DATA BUFFER.  WRFIFO WILL TRANSFER THE DATA FROM RAM TO THE FIFO
                  795 ;       DATA BUFFER.  WRFIFO RETURNS  THE VALUE OF THE BPK72 STATUS REG
                  796 ;       TO THE CALLING ROUTINE VIA THE 8085'S A REG.  ONLY  A STATUS OF
                  797 ;       41H OR 43H INDICATES A SUCCESSFUL EXECUTION OF WRFIFO.
                  798 ;
                  799         PUBLIC  WRFIFO  ; DECLARE PUBLIC FUNCTION
 0A70 C5          800 WRFIFO: PUSH    B       ; SAVE B-C REGS
 0A71 D5          801         PUSH    D       ; SAVE D-E REGS
 0A72 0640        802         MVI     B,40H   ; LOAD B REG= 40H, OP-COMPLETE
 0A74 0E28        803         MVI     C,28H   ; LOAD C REG= 28H, INITIALIZE LOOP COUNTER
 0A76 CD1308      804         CALL    FIFORS  ; CALL FIFORS, WRITE FIFO RESET COMMAND
 0A79 A8          805         XRA     B       ; TEST FOR STATUS REG= 40H, OP-COMPLETE
 0A7A C2850A      806         JNZ     RETWF   ; IF NOT ZERO, FIFO ERROR, JMP RETWF
 0A7D 1A          807 INFIFO: LDAX    D       ; LOAD A REG FROM D-E REG ADDRESS
 0A7E D3FE        808         OUT     PRTA00  ; WRITE A REG TO 7220 FIFO DATA BUFFER
 0A80 13          809         INX     D       ; INCREMENT D-E REGS TO NEXT ADDRESS IN RAM
 0A81 0D          810         DCR     C       ; DECREMENT LOOP COUNTER
 0A82 C27D0A      811         JNZ     INFIFO  ; IF LOOP COUNTER NOT ZERO, JMP INFIFO
 0A85 D1          812 RETWF:  POP     D       ; RESTORE D-E REGS
 0A86 C1          813         POP     B       ; RESTORE B-C REGS
 0A87 DBFF        814         IN      PRTA01  ; READ STATUS REG
 0A89 C9          815         RET             ; RETURN TO CALL
                  816 ;
                  817 ;
                  818 ;
                  819 $EJECT
```

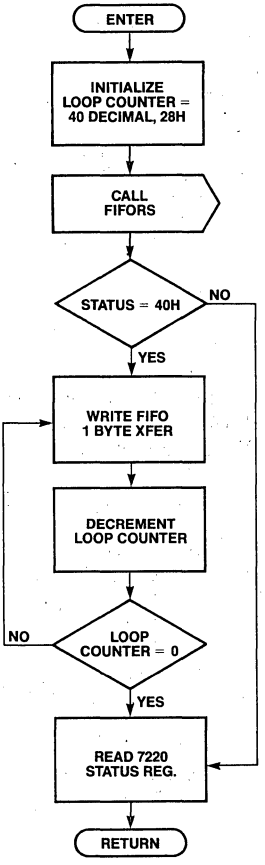


Figure 22. WRFIFO

```
ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0        BPK72   PAGE   23


 LOC  OBJ        LINE       SOURCE STATEMENT

                  820 ;*************************************************************************
                  821 ;
                  822 ; FUNCTION: RDFIFO
                  823 ; INPUTS:   D-E REGS STARTING ADDRESS IN RAM
                  824 ;           BPK72 STATUS REG
                  825 ;           READ 40 BYTES OF DATA FROM BPK72 FIFO DATA BUFFER
                  826 ; OUTPUTS:  TRANSFER FIFO DATA BUFFER TO RAM
                  827 ;           A REG= BPK72 STATUS REG
                  828 ; CALLS:    NONE
                  829 ; DESTROYS: A, F/FS
                  830 ;
                  831 ; DESCRIPTION: READ 7220 FIFO DATA BUFFER
                  832 ;       THE  D-E  REGS  CONTAIN THE ADDRESS TO THE  FIRST OF 40 CONTIGUOUS
                  833 ;       BYTES IN RAM THAT  WILL BE LOADED  WITH THE CONTENTS OF THE  BPK72
                  834 ;       FIFO DATA BUFFER. RDFIFO WILL TRANSFER THE DATA FROM THE FIFO DATA
                  835 ;       BUFFER TO RAM.  RDFIFO RETURNS THE VALUE OF  THE BPK72 STATUS  REG
                  836 ;       TO THE CALLING ROUTINE VIA THE 8085'S A REG.  ONLY A STATUS OF 40H
                  837 ;       OR 42H INDICATES A SUCCESSFUL EXECUTION OF RDFIFO.
                  838 ;
                  839         PUBLIC  RDFIFO  ; DECLARE PUBLIC FUNCTION
  0A8A C5         840 RDFIFO: PUSH    B       ; SAVE B-C REGS
  0A8B D5         841         PUSH    D       ; SAVE D-E REGS
  0A8C 0E28       842         MVI     C,28H   ; LOAD C REG= 28H, INITIALIZE LOOP COUNTER
  0A8E DBFE       843 OUTFIF: IN      PRTA00  ; LOAD A REG WITH ONE BYTE FROM FIFO DATA BUFFER
  0A90 12         844         STAX    D       ; LOAD A REG IN D-E REG ADDRESS
  0A91 13         845         INX     D       ; INCREMENT D-E REGS TO NEXT ADDRESS
  0A92 0D         846         DCR     C       ; DECREMENT LOOP COUNTER
  0A93 C28E0A     847         JNZ     OUTFIF  ; IF LOOP COUNTER NOT ZERO, JMP OUTFIF
  0A96 D1         848         POP     D       ; RESTORE D-E REGS
  0A97 C1         849         POP     B       ; RESTORE B-C REGS
  0A98 DBFF       850         IN      PRTA01  ; READ STATUS REG
  0A9A C9         851         RET             ; RETURN TO CALL
                  852 ;
                  853 ;
                  854 $EJECT
```

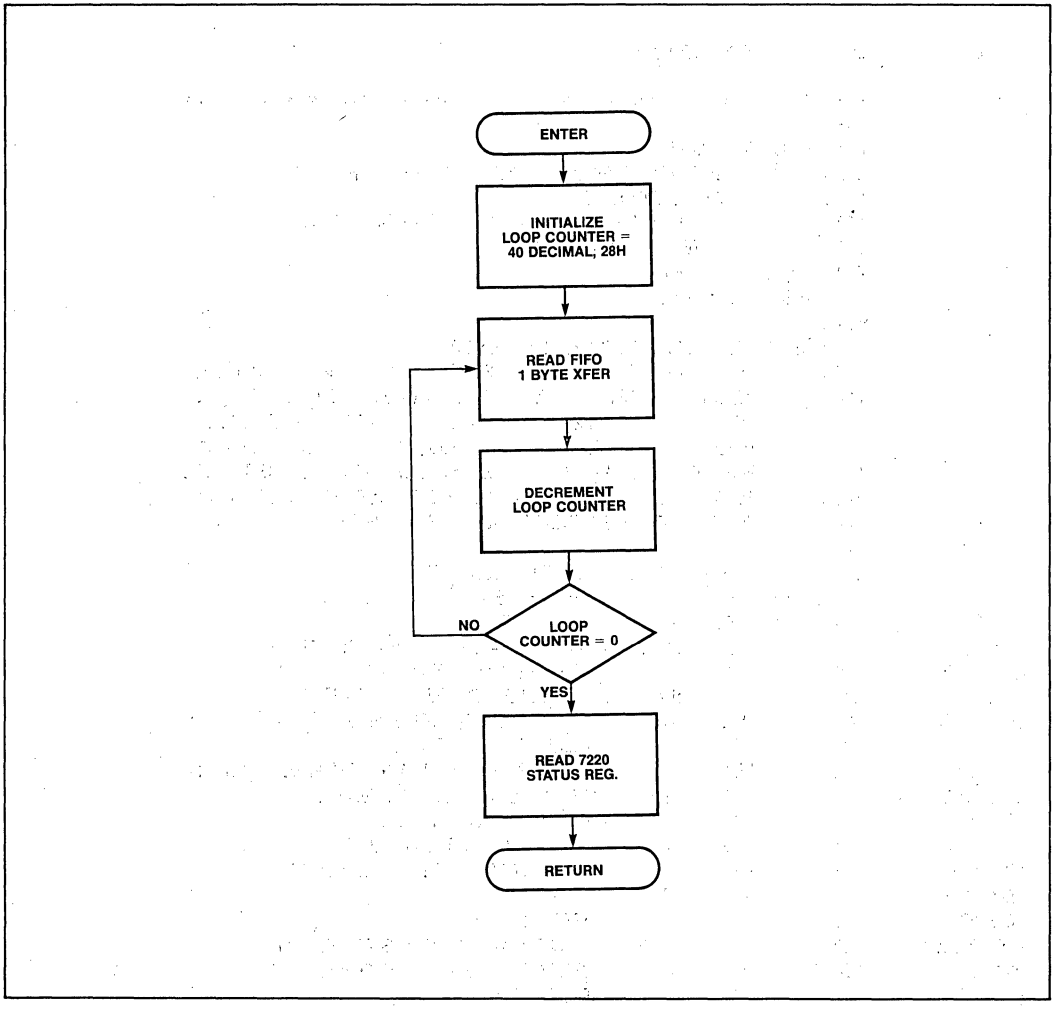
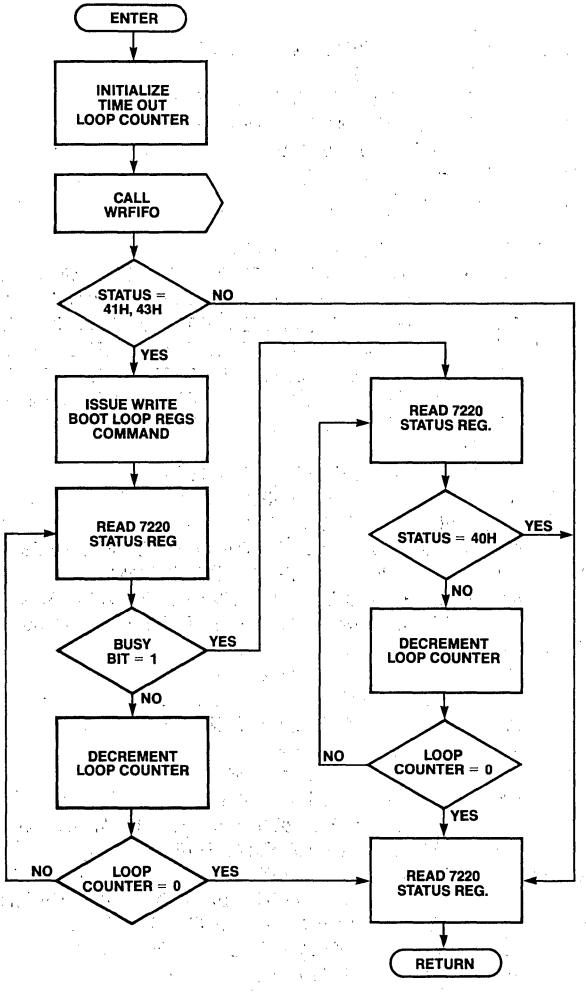


Figure 23. RDFIFO

```
ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0        BPK72    PAGE   24


 LOC  OBJ        LINE       SOURCE STATEMENT

                  855 ;*************************************************************************
                  856 ;
                  857 ; FUNCTION: WRBLRS
                  858 ; INPUTS:   D-E REGS, STARTING ADDRESS OF DATA IN RAM
                  859 ;           BPK72 STATUS REG
                  860 ; OUTPUTS:  WRITE BUBBLE MEMORY BOOT LOOP REGISTERS COMMAND
                  861 ;           A REG= BPK72 STATUS REG
                  862 ; CALLS:    WRFIFO
                  863 ; DESTROYS: A, F/FS
                  864 ;
                  865 ; DESCRIPTION: WRITE 7242 BOOT LOOP REGISTERS
                  866 ;       THE D-E REGS PROVIDE THE ADDRESS TO THE FIRST OF 40 CONTIGUOUS
                  867 ;       MEMORY  LOCATIONS  IN RAM THAT  CONTAIN DATA TO BE LOADED INTO
                  868 ;       THE 7242,  FORMATTER  SENSE  AMPLIFIER,  BOOT  LOOP REGISTERS.
                  869 ;       WRBLRS WILL  TRANSFER  THE DATA  FROM  RAM  TO  THE  BOOT LOOP
                  870 ;       REGISTERS.  WRBLRS  RETURNS THE VALUE OF THE BPK72  STATUS REG
                  871 ;       TO THE CALLING  ROUTINE VIA THE 8085'S A REG.  ONLY A STATUS OF
                  872 ;       40H INDICATES A SUCCESSFUL EXECUTION OF WRBLRS.
                  873 ;
                  874         PUBLIC  WRBLRS  ; DECLARE PUBLIC FUNCTION
  0A9B C5         875 WRBLRS: PUSH    B       ; SAVE B-C REGS
  0A9C E5         876         PUSH    H       ; SAVE H-L REGS
  0A9D 0641       877         MVI     B,41H   ; LOAD B REG= 41H, OP-COMPLETE, FIFO FULL
  0A9F 0EFD       878         MVI     C,0FDH  ; MASK, MASK OUT PARITY BIT
  0AA1 21FFFF     879         LXI     H,0FFFFH; INITIALIZE TIME OUT LOOP COUNTER
  0AA4 CD700A     880         CALL    WRFIFO  ; CALL WRITE FIFO DATA BUFFER
  0AA7 A1         881         ANA     C       ; RESET BIT 1, PARITY BIT
  0AA8 A8         882         XRA     B       ; TEST STATUS= 41H OR 43H, OP-COMPLETE, FIFO FULL
  0AA9 C2CE0A     883         JNZ     RETWBL  ; IF NOT ZERO, ERROR, JMP RETWBL
  0AAC 05         884         DCR     B       ; B REG= 40H, OP-COMPLETE
  0AAD 3E16       885         MVI     A,16H   ; LOAD A REG= WRITE BOOT LOOP REG COMMAND
  0AAF D3FF       886         OUT     PRTA01  ; WRITE, WRITE BOOT LOOP REG COMMAND
  0AB1 DBFF       887 BSYWBL: IN      PRTA01  ; READ STATUS REG
  0AB3 07         888         RLC             ; TEST BUSY BIT= 1
  0AB4 DAC10A     889         JC      POLWBL  ; IF BUSY= 1, POLL STATUS REG FOR 40H
  0AB7 2B         890         DCX     H       ; DECREMENT TIME OUT LOOP COUNTER
  0AB8 AF         891         XRA     A       ; CLEAR A REG
  0AB9 B4         892         ORA     H       ; TEST H REG= 00H
  0ABA B5         893         ORA     L       ; TEST L REG= 00H
  0ABB C2B10A     894         JNZ     BSYWBL  ; IF NOT ZERO, CONTINUE POLLING WRBLR COMMAND
  0ABE C3CE0A     895         JMP     RETWBL  ; TIME OUT ERROR, RETURN
  0AC1 DBFF       896 POLWBL: IN      PRTA01  ; READ STATUS REG
  0AC3 A8         897         XRA     B       ; TEST STATUS REG= 40H, OP-COMPLETE
  0AC4 CACE0A     898         JZ      RETWBL  ; IF ZERO, OP-COMPLETE, JMP RETWBL
  0AC7 2B         899         DCX     H       ; DECREMENT TIME OUT LOOP COUNTER
  0AC8 AF         900         XRA     A       ; CLEAR A REG
  0AC9 B4         901         ORA     H       ; TEST H REG= 00H
  0ACA B5         902         ORA     L       ; TEST L REG= 00H
  0ACB C2C10A     903         JNZ     POLWBL  ; IF NOT ZERO, CONTINUE POLLING WRBLR COMMAND
  0ACE E1         904 RETWBL: POP     H       ; RESTORE H-L REGS
  0ACF C1         905         POP     B       ; RESTORE B-C REGS
  0AD0 DBFF       906         IN      PRTA01  ; READ STATUS REG
  0AD2 C9         907         RET             ; RETURN TO CALL
                  908 $EJECT
```



COMMENT: MIN TIME OUT LOOP COUNTER = 1MS

Figure 24. WRBLRS

```
ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0        BPK72    PAGE   25


 LOC  OBJ        LINE       SOURCE STATEMENT

                  909 ;*****************************************************************************
                  910 ;
                  911 ; FUNCTION: RDBLRS
                  912 ; INPUTS:   D-E REGS, STARTING ADDRESS IN RAM
                  913 ;           BPK72 STATUS REG
                  914 ;           READ DATA FROM 7242 BOOT LOOP REGISTERS
                  915 ; OUTPUTS:  TRANSFER BOOT LOOP REGISTER DATA TO RAM
                  916 ;           A REG= BPK72 STATUS REG
                  917 ; CALLS:    RDFIFO
                  918 ; DESTROYS: A, F/FS
                  919 ;
                  920 ; DESCRIPTION: READ 7242 BOOT LOOP REGISTERS
                  921 ;       THE D-E REGS CONTAIN THE ADDRESS TO THE FIRST OF 40 CONTIGUOUS
                  922 ;       MEMORY LOCATIONS IN RAM TO BE  LOADED WITH THE CONTENTS OF THE
                  923 ;       7242, FORMATTER  SENSE AMPLIFIER, BOOT LOOP REGISTERS.  RDBLRS
                  924 ;       WILL COPY  THE CONTENTS  OF THE  BOOT  LOOP  REGISTERS TO RAM.
                  925 ;       RDBLRS  RETURNS THE  VALUE OF  THE  BPK72  STATUS  REG  TO THE
                  926 ;       CALLING ROUTINE  VIA THE 8085'S  A REG.  ONLY A  STATUS OF 40H
                  927 ;       INDICATES A SUCCESSFUL EXECUTION OF RDBLRS.
                  928 ;
                  929         PUBLIC  RDBLRS  ; DECLARE PUBLIC FUNCTION
  0AD3 C5         930 RDBLRS: PUSH    B       ; SAVE B-C REGS
  0AD4 E5         931         PUSH    H       ; SAVE H-L REGS
  0AD5 06C1       932         MVI     B,0C1H  ; LOAD B REG= C1H, OP-COMPLETE, FIFO FULL >22 BYTES (BUSY BIT=1)
  0AD7 21FFFF     933         LXI     H,0FFFFH; INTIALIZE TIME OUT LOOP COUNTER
  0ADA 3E15       934         MVI     A,15H   ; LOAD A REG= READ BOOT LOOP REGS COMMAND
  0ADC D3FF       935         OUT     PRTA01  ; WRITE THE READ BOOT LOOP REGS COMMAND
  0ADE DBFF       936 BSYRBL: IN      PRTA01  ; READ STATUS REG
  0AE0 07         937         RLC             ; TEST BUSY BIT= 1
  0AE1 DAEE0A     938         JC      POLRBL  ; IF BUSY= 1, POLL STATUS REG FOR C1H
  0AE4 2B         939         DCX     H       ; DECREMENT TIME OUT LOOP COUNTER
  0AE5 AF         940         XRA     A       ; CLEAR A REG
  0AE6 B4         941         ORA     H       ; TEST H REG= 00H
  0AE7 B5         942         ORA     L       ; TEST L REG= 00H
  0AE8 C2DE0A     943         JNZ     BSYRBL  ; IF NOT ZERO, CONTINUE POLLING READ BOOT LOOP REG COMMAND
  0AEB C3010B     944         JMP     RETRBL  ; TIME OUT ERROR, RETURN
  0AEE DBFF       945 POLRBL: IN      PRTA01  ; READ STATUS REG
  0AF0 A8         946         XRA     B       ; TEST STATUS= C1H, OP-COMPLETE, FIFO FULL
  0AF1 CAFE0A     947         JZ      CALLRD  ; IF ZERO, OP-COMPLETE, JMP CALLRD
  0AF4 2B         948         DCX     H       ; DECREMENT TIME OUT LOOP COUNTER
  0AF5 AF         949         XRA     A       ; CLEAR A REG
  0AF6 B4         950         ORA     H       ; TEST H REG= 00H
  0AF7 B5         951         ORA     L       ; TEST L REG= 00H
  0AF8 CA010B     952         JZ      RETRBL  ; IF ZERO, ERROR, JMP RETRBL
  0AFB C3EE0A     953         JMP     POLRBL  ; CONTINUE POLLING READ BOOT LOOP REG COMMAND
  0AFE CD8A0A     954 CALLRD: CALL    RDFIFO  ; CALL READ FIFO
  0B01 E1         955 RETRBL: POP     H       ; RESTORE H-L REGS
  0B02 C1         956         POP     B       ; RESTORE B-C REGS
  0B03 DBFF       957         IN      PRTA01  ; READ STATUS REG
  0B05 C9         958         RET             ; RETURN TO CALL
                  959 $EJECT
```

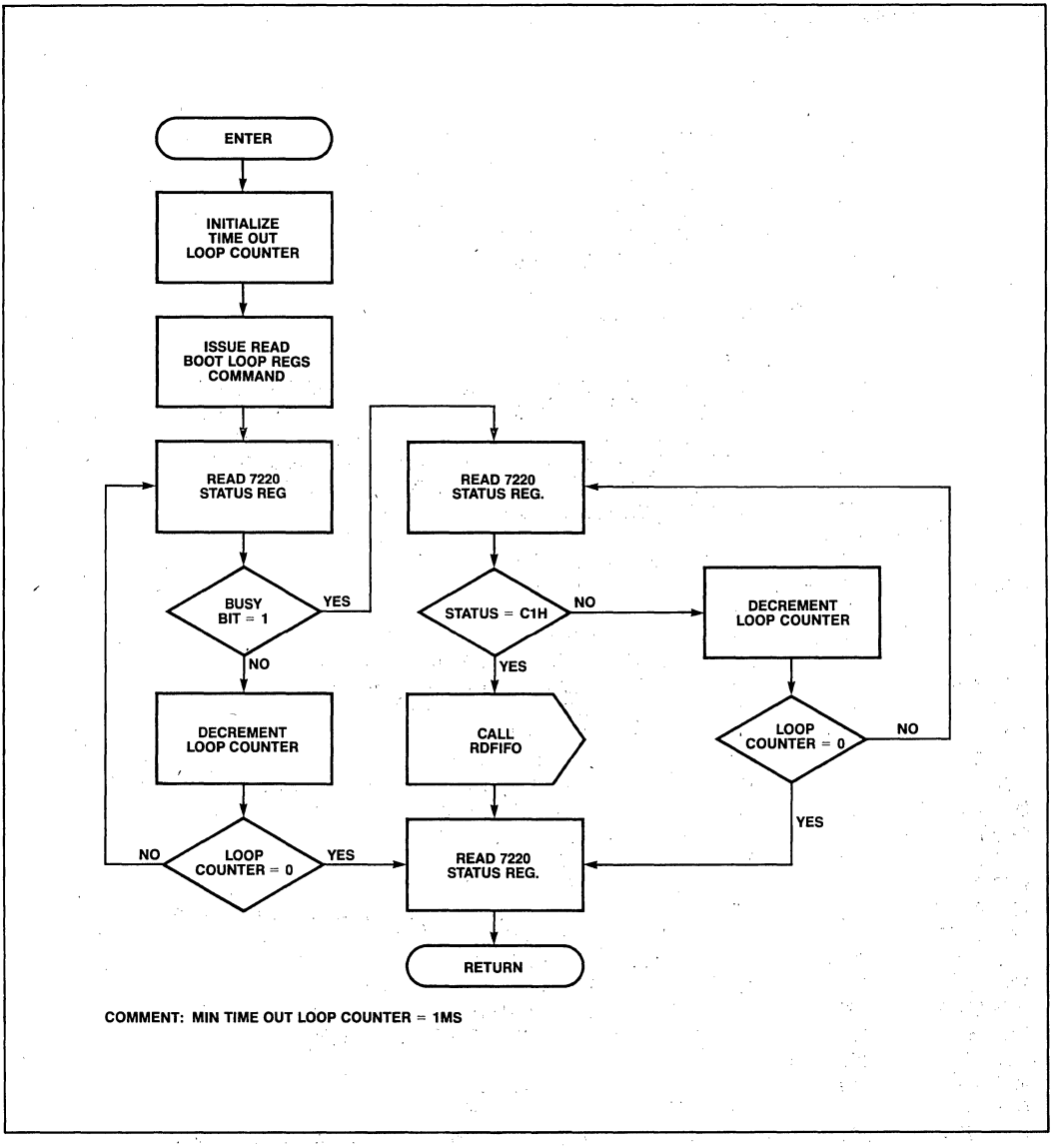
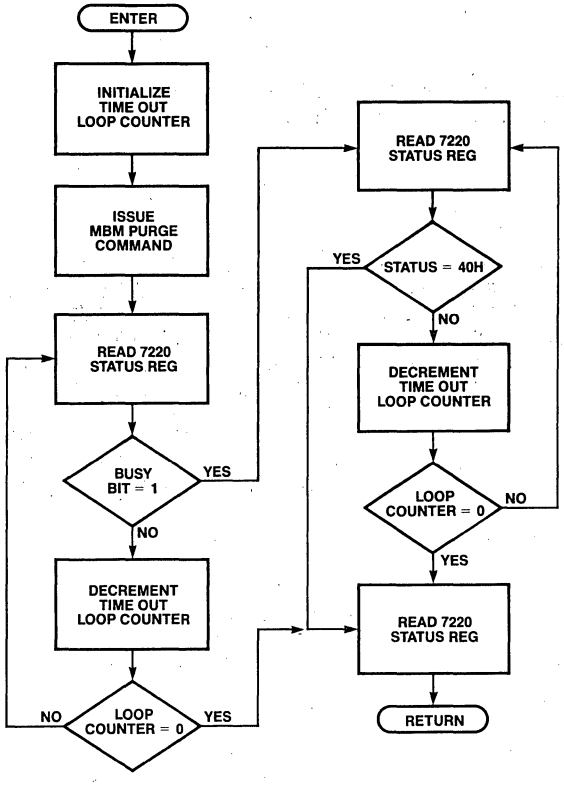


Figure 25. RDBLRS

```
ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0        BPK72    PAGE   26


 LOC  OBJ        LINE       SOURCE STATEMENT

                  960 ;*************************************************************************
                  961 ;
                  962 ; FUNCTION: MBMPRG
                  963 ; INPUTS:   BPK72 STATUS REG
                  964 ; OUTPUTS:  ISSUE MBM PURGE COMMAND
                  965 ;          A REG= BPK72 STATUS REG
                  966 ; CALLS:    NONE
                  967 ; DESTROYS: A, F/FS
                  968 ;
                  969 ; DESCRIPTION: MBM PURGE COMMAND
                  970 ;       AN MBM PURGE COMMAND IS ISSUED TO THE BPK72.  AFTER ISSUING THE
                  971 ;       COMMAND,  THE BPK72  STATUS REG IS POLLED UNTIL AN OP-COMPLETE
                  972 ;       40H,  HAS BEEN  READ OR THE  TIME OUT LOOP  COUNTER DECREMENTS
                  973 ;       TO ZERO.  MBMPRG  RETURNS THE VALUE OF THE BPK72 STATUS REG TO
                  974 ;       THE CALLING ROUTINE VIA THE 8085'S A REG.  ONLY A STATUS OF 40H
                  975 ;       INDICATES A SUCCESSFUL EXECUTION OF MBMPRG.
                  976 ;
                  977         PUBLIC  MBMPRG  ; DECLARE PUBLIC FUNCTION
  0B06 D5         978 MBMPRG: PUSH    D       ; SAVE D-E REGS
  0B07 C5         979         PUSH    B       ; SAVE B-C REGS
  0B08 0640       980         MVI     B,40H   ; LOAD B REG= 40H, OP-COMPLETE
  0B0A 11FFFF     981         LXI     D,0FFFFH; INITIALIZE TIME OUT LOOP COUNTER
  0B0D 3E1E       982         MVI     A,1EH   ; LOAD A REG= MBM PURGE COMMAND
  0B0F D3FF       983         OUT     PRTA01  ; WRITE MBM PURGE COMMAND
  0B11 DBFF       984 BSYMBM: IN      PRTA01  ; READ STATUS REG
  0B13 07         985         RLC             ; TEST BUSY BIT= 1
  0B14 DA210B     986         JC      POLMBM  ; IF BUSY= 1, POLL STATUS REG FOR 40H
  0B17 1B         987         DCX     D       ; DECREMENT TIME OUT LOOP COUNTER
  0B18 AF         988         XRA     A       ; CLEAR A REG
  0B19 B2         989         ORA     D       ; TEST D REG= 00H
  0B1A B3         990         ORA     E       ; TEST E REG= 00H
  0B1B C2110B     991         JNZ     BSYMBM  ; IF NOT ZERO, CONTINUE POLLING THE MBMPRG COMMAND
  0B1E C32E0B     992         JMP     RETMBM  ; TIME OUT ERROR, RETURN
  0B21 DBFF       993 POLMBM: IN      PRTA01  ; READ STATUS REG
  0B23 A8         994         XRA     B       ; TEST STATUS= 40H, OP-COMPLETE
  0B24 CA2E0B     995         JZ      RETMBM  ; IF OP-COMPLETE, JMP RETMBM
  0B27 1B         996         DCX     D       ; DECREMENT TIME OUT LOOP COUNTER
  0B28 AF         997         XRA     A       ; CLEAR A REG
  0B29 B2         998         ORA     D       ; TEST D REG= 00H
  0B2A B3         999         ORA     E       ; TEST E REG= 00H
  0B2B C2210B    1000         JNZ     POLMBM  ; IF NOT ZERO, CONTINUE POLLING MBM PURGE COMMAND
  0B2E C1        1001 RETMBM: POP     B       ; RESTORE B-C REGS
  0B2F D1        1002         POP     D       ; RESTORE D-E REGS
  0B30 DBFF      1003         IN      PRTA01  ; READ STATUS REG
  0B32 C9        1004         RET             ; RETURN TO CALL
                 1005 $EJECT
```



COMMENT: MINIMUM TIME OUT LOOP COUNTER = 150μS.

Figure 26. MBMPRG

```
ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0        BPK72    PAGE   27


  LOC  OBJ         LINE        SOURCE STATEMENT

                   1006 ;
                   1007         END


PUBLIC SYMBOLS
ABORT  A 08DE    BOOTUP A 099C    FIFORS A 0813    INBUBL A 0961    MBMPRG A 0B06    RDBLRS A 0AD3    RDBOOT A 0A2C
RDBUBL A 0936    RDFIFO A 0A8A    WRBLRS A 0A9B    WRBUBL A 090B    WRFIFO A 0A70

EXTERNAL SYMBOLS


USER SYMBOLS
ABORT  A 08DE    ALLFFS A 09BB    BLCODE A 09F1    BOOTUP A 099C    BSYMBM A 0B11    BSYRBL A 0ADE    BSYWBL A 0AB1
BTLPRD A 0A52    BUSYA  A 08E9    BUSYB  A 09C8    BUSYBL A 0A05    BUSYFR A 081E    BUSYIN A 097A    BUSYRB A 0A42
BUSYRD A 08AD    BUSYWR A 0873    BYTCNT A 0840    CALLRD A 0AFE    CONT   A 09E8    DONE   A 0867    FIFORD A 0A62
FIFORS A 0813    FINSHR A 08DB    FINSHW A 08A1    INBUBL A 0961    INFIFO A 0A7D    INTPAR A 0800    LOAD   A 0808
LOOPRD A 094F    LOOPWR A 0924    MBMPRG A 0B06    MULT   A 0852    MULT1  A 0862    MULTO  A 0856    OUTFIF A 0A8E
POLLA  A 08F9    POLLBL A 0A15    POLLBR A 09D8    POLLFR A 082E    POLLIN A 098A    POLLRD A 08BA    POLLWR A 0880
POLMBM A 0B21    POLRBL A 0AEE    POLWBL A 0AC1    PRTA00 A 00FE    PRTA01 A 00FF    RDBLRS A 0AD3    RDBOOT A 0A2C
RDBUBL A 0936    RDFIFO A 0A8A    READ   A 08A4    RETA   A 0906    RETBT  A 0A23    RETFR  A 083B    RETIN  A 0997
RETMBM A 0B2E    RETRBL A 0B01    RETRD  A 095C    RETRDB A 0A6A    RETWBL A 0ACE    RETWF  A 0A85    RETWR  A 0931
RFIFO  A 08D0    WFIFO  A 0896    WRBLRS A 0A9B    WRBUBL A 090B    WRFIFO A 0A70    WRITE  A 086A

ASSEMBLY COMPLETE,  NO ERRORS
```

APPENDIX B
SERVICE INFORMATION

SERVICE INFORMATION

Typically, a Bubble Memory System will never require any special service throughout its useful life. The sequence of program flow presented in Appendix B is not required for normal read/write operation. However, power supply failure, socket contact problems, or component failures may inadvertently produce a BPK 72 system failure.

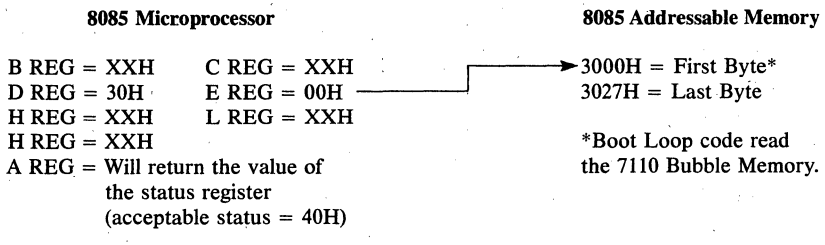
Note: Power supply failure is defined as any violation of the power supply specifications listed in the section titled, "Power Supply Requirements."

A figure titled, "BPK 72 Failure Recovery" is included in Appendix C to illustrate the sequence of events necessary to remedy a Bubble Memory System failure. The flowchart is intended as a guide for handling a Bubble Memory System failure. A system failure is defined as continued attempts that fail to read and write data correctly. Upon detection of a BPK 72 system failure, the first course of action is to verify the existence of the seeds within the 7110 Bubble Memory module. Four replicating Bubble Memory generators reside in the 7110. Each generator requires one seed from which all other bubbles are created. Under extreme circumstances such as power supply failure, one or all of the seeds can be destroyed making it impossible to write data into the 7110's storage loops. The "BPK 72 Failure Recovery" flowchart requests a call to the "seed verification procedure." The "seed verification procedure" should be followed closely to determine if any of the seeds are missing.

In the unlikely event that some or all of the seeds are lost, the "BPK 72 Failure Recovery" figure instructs the reader to perform the "procedure to reseed a 7110 Bubble Memory." The seed replacement procedure will create a seed in each of the four generators. After completing the seed replacement procedure, the "seed verification procedure" should be performed again to confirm that all four seeds are present in the 7110.

The next step in diagnosing a BPK 72 system failure is to verify the accuracy of the boot loop code within the 7110. The boot loop is a map containing information about the active and inactive storage loops. The 7110 is designed with a 15% storage loop redundancy to improve the product yield during manufacture. A diagnostic subroutine named RDBOOT can be called to read the boot loop from the 7110. It is the responsibility of the calling routine to verify that the boot loop code read from the 7110 matches byte for byte with the code found on the label attached to the case of the Bubble Memory module.

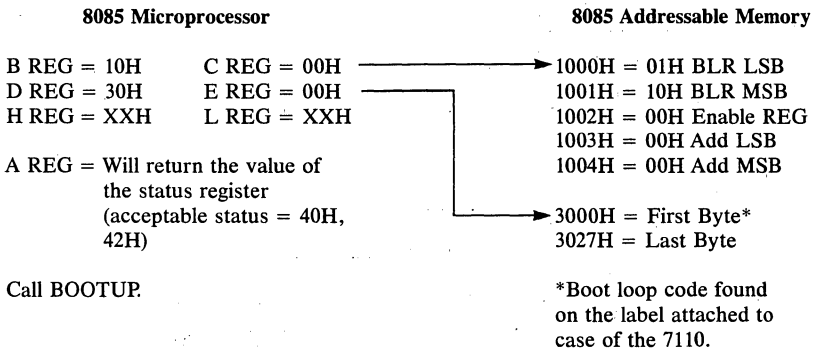
The following is an example of how to use the read Bubble Memory boot loop subroutine, RDBOOT:



Additional detail regarding the use of the read Bubble Memory boot loop subroutine, RDBOOT, may be found in the software listing presented in Appendix A.

If the boot loop is incorrect, a subroutine called BOOTUP is provided for writing the boot loop into the 7110.

The following is an example of how to use BOOTUP to write the boot loop code into the 7110:



Additional detail regarding the use of the write Bubble Memory boot loop subroutine, BOOTUP, may also be found in the software listing presented in Appendix A.

After the seeds and boot loop have been examined and replaced as necessary, the remaining step is to call the initialization subroutine, INBUBL. See the section titled, "Initalizing the Bubble" for a description of how to call the initialization subroutine. If the initialization subroutine returns a status of 40H, the BPK 72 is ready to be put back into service.

Contact the local Intel field sales office in the unlikely event that the BPK 72 system failure guidelines do not eliminate the problem.

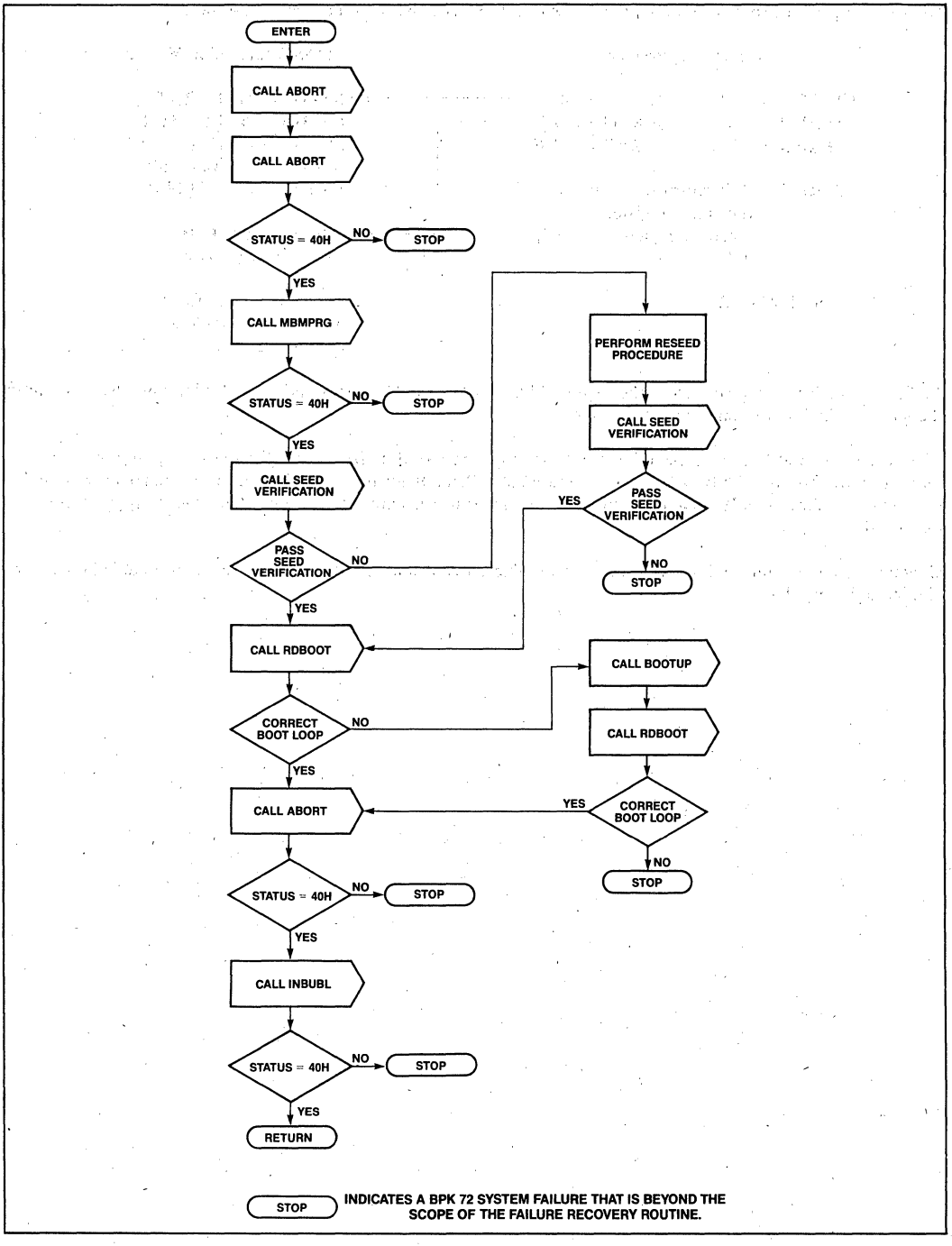
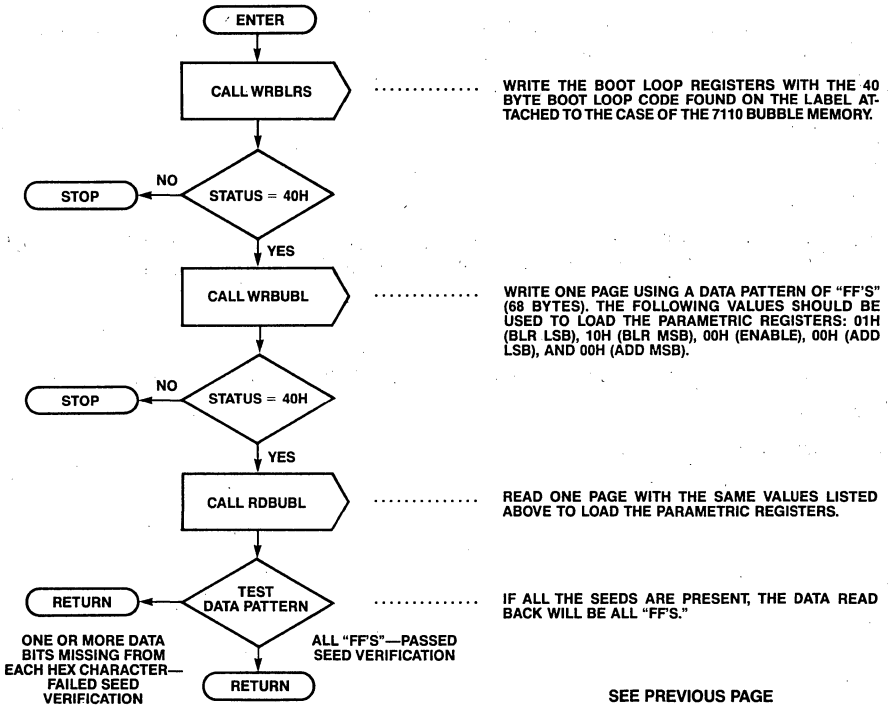


Figure 28. BPK 72 Failure Recovery



If one or more seeds are missing, the data read back will be a pattern with one or more bits missing from each hex character. One example of several possible patterns is shown below. Each pattern will typically contain a dominant pair of hex characters (i.e., "88's" or "AA's"). In any case, if seeds are missing no "FF's" will be read using the subroutine, RDBUBL.

88	88	88	88	88	88	88	88	88	00	08	88	88	88	88	88
88	88	88	88	88	08	80	88	88	88	88	88	88	88	88	88
88	A8	88	80	08	88	88	88	88	80	88	88	A8	88	8A	88
A8	88	8A	88	88	88	88	A8	88	AA	88	88	88	8A	88	88

Do not attempt to use the seed verification procedure without first performing the program sequence described in Figure 28, "BPK 72 Failure Recovery."

Figure 29. Seed Verification Procedure

PROCEDURE TO RESEED A 7110 BUBBLE MEMORY

1. Remove power from circuit.
2. Remove the 7230 current pulse generator from its socket, and install the 7230 in the socket provided on the seed module. Be careful to note the orientation of Pin 1.
3. Install the seed module (with the 7230 installed) in the 7230 socket.
4. Apply power to the circuit.
5. Call ABORT.
6. Call MBMPRG.
7. Call WRBUBL (1 page transfer, any location, data pattern is not important). Parametric register values; 01H (BLR LSB), 10H (BLR MSB), 00H (ENABLE), 00H (add LSB), and 00H (add MSB).
8. Remove power from circuit.
9. Remove the seed module from the 7230 socket.
10. Remove the 7230 from the seed module and reinstall the 7230 in its socket on the IMB-72 board.
11. Apply power to the circuit.
12. Reseed procedure is now complete.