Bubble Memory 6

A PRIMER ON MAGNETIC BUBBLE MEMORY

Magnetic bubble memory is a solid-state technology with high reliability, ruggedness, small size, light weight, and limited power dissipation. It has applications in telecommunications, data acquisition, industrial control, terminals, and small business computers. Yet many potential users remain unsure of the nature of a bubble memory. This primer is intended to introduce these users to the technology.

What a Magnetic Bubble Memory Is

A magnetic bubble memory stores data in the form of cylindrical magnetic domains in a thin film of magnetic material. The presence of a domain (a bubble) is interpreted as a binary 1, and absence of a domain is a 0. Bubbles are created from electrical signals by a bubble generator within the memory, and reconverted to electrical signals by an internal detector. Externally the memory is TTL-compatible.

An external rotating magnetic field propels these cylindrical domain bubbles through the film. Metallic patterns or chevrons deposited on the film steer the domains in the desired directions. Transfer rates, once started, are in the tens of thousands of bits per second, but because the data circulates past a pickup point at which it becomes available to the outside world, there is a latency averaging tens of milliseconds before data transfer can begin. In these respects, magnetic bubble memories are serial high-density storage devices like electromechanical disk memories. However, in a disk, the stored bits are stationary on a moving medium, whereas in the magnetic bubble memory the medium is stationary and the bits move.

Advantages of Magnetic Bubble Memories

The principal advantage of magnetic bubble memories are their non-volatility—that is, if power fails, the stored data is retained. Products incorporating bubble memories therefore do not require battery backups. Magnetic bubble memories share this feature with read only memories (ROMs), erasable PROMs (EPROMS), and electrically erasable PROMs (E^2PROMS). However, unlike any of these technologies, magnetic bubble memories can have data written into them at any time, at speeds comparable to those at which data is read. Furthermore, unlike disk memories, bubble memories are quiet and very reliable, because they have no moving parts. They are compact, and they dissipate very little power. Their support circuits are compatible with microprocessor systems. With a million or more bits per device, a bubble memory can store 16 to 64 times the amount of data of alternative semiconductor memories, providing very high storage capability in a compact space. Bubble memory has a wide variety of applications, some of which are listed in Table 1.

Table 1. **Bubble Memory Applications**

Numerical control	Robotics
Process control	Oil exploration
Aircraft navigation	Data acquisition
Cable television	Portable instruments
Telecommunications terminals	Avionics
Point-of-sale terminals	Gasoline pumps
Private branch telephone exchanges	Personal computers
Word processors	Office equipment
Flight-line test equipment	Automatic test equipment
Data encryption	

How Bubbles are Formed

Magnetic domains are found in all kinds of magnetic materials—iron bars, the coating on magnetic tape, ferrite toroids (the most common form of computer memory in the 1960s). Each domain is a group of atoms with parallel magnetic orientations. When the material in bulk is unmagnetized, the domains are oriented at random in three dimensions. When the material is magnetically saturated, most of the domains have the same orientation. Magnetization to a level less than saturation orients some of the domains to a common direction, but leaves many of them randomly oriented. When a domain orientation changes, usually by imposing an external magnetic field, the domain itself does not physically move, but boundaries between domains that have different orientations move or disappear altogether.

In an extremely thin film, less than 0.001 inch thick, the domain orientations may be constrained to two dimensions. In some kinds of material (orthoferrites and garnets), with proper crystallographic orientation, the domain orientations are always perpendicular to the film. When these materials are not in a magnetic field, some domains are oriented upward and some downward (north magnetic poles of some domains are on top of the film, and those of other domains are on the bottom). In these materials, the magnetic domains tend to be long and snakelike in the absence of an external field (Figure 1). When a weak magnetic field is applied perpendicular to the film, the domains that are oriented opposite to the applied field become substantially narrower. As the applied field, called a *bias field*, is made stronger, the length continues to decrease, until it becomes approximately the same as the width. Each domain is now cylindrical, magnetized oppositely to the applied field, and immersed in a much larger domain that is magnetized in the same direction as the field.

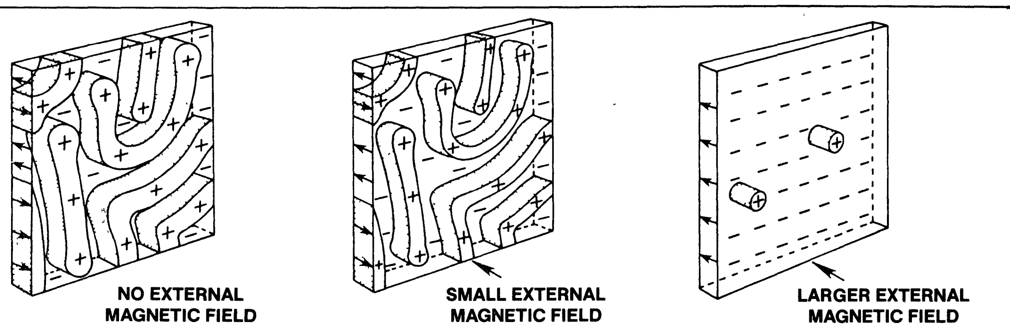


Figure 1. Magnetic Domains in Thin Film Under Increasing Magnetic Bias Field.

These small domains are the bubbles, generally less than 3 micrometers (1/10,000 inch) in diameter (Figure 2). When they are viewed from above, only the round shape is apparent, giving the domains the appearance of bubbles. If the bias field were to be made still stronger, all the bubbles would shrink and then disappear altogether; the entire film would be magnetized in the same direction as the bias field. The effect is reversible—that is, if the bias is removed, the domains return to a snakelike form.

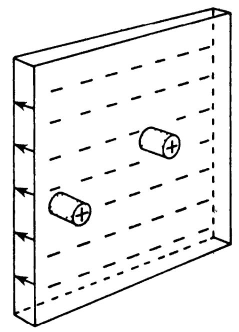


Figure 2. Magnetic Bubbles in a Thin Film

Why a Bubble Moves

Magnetic bubbles will move if they are in a magnetic field gradient. For instance, it will move from a region of lesser magnetic field strength to a region of greater strength. This is similar to the way a nail is pulled to the end of a bar magnet when it gets close the magnet.

In a bubble memory a magnetic film pattern is overlaid on the layer of bubbles. When this layer is magnetized it pulls the bubbles to the points of greatest field strength (or poles) as in Figure 3. The bubbles could then be moved if the pattern elements were moved.

A more easily controlled magnetic field is generated by two coils wrapped around the bubble layer and magnetic film pattern. With appropriate specification of the current in two coils positioned at right angles, the direction of the poles on the stationary elements can be changed in a controlled manner.

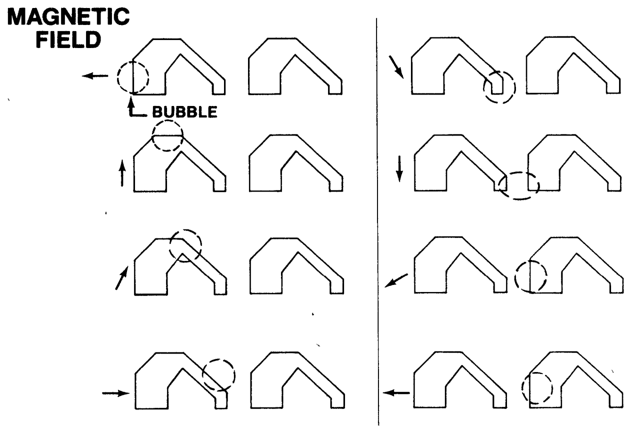


Figure 3. Bubble Propagation Under Asymmetric Chevrons

Various shapes for these metallic patterns have been used by different manufacturers to control the movement of the bubbles. At Intel asymmetric chevrons are used (Figure 3).

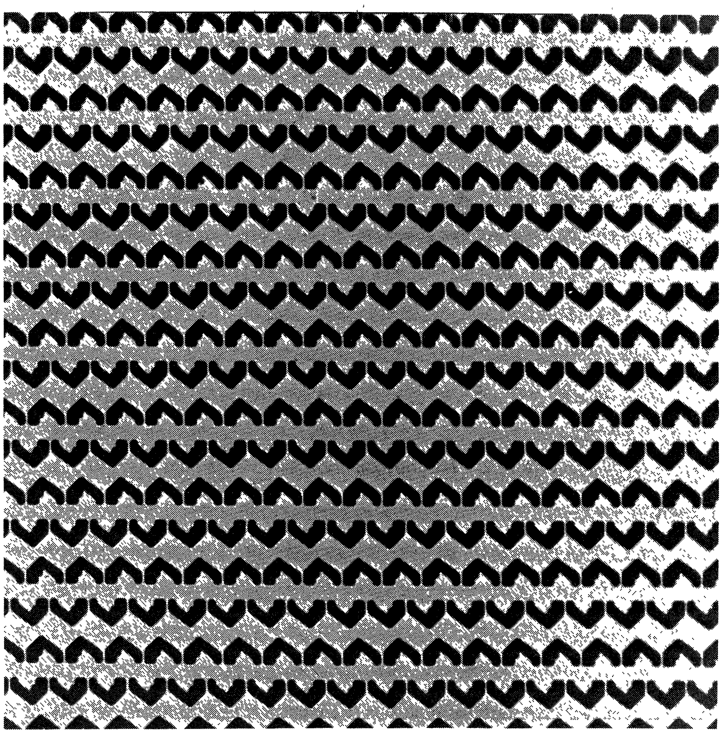


Photo 1. Asymmetric Chevrons Deposited on a Thin Film

Why Magnetic Bubbles are Non-Volatile

In a magnetic bubble memory system, the bias field in which the bubbles exist is generated by a pair of *permanent* magnets. The substrate bearing the thin film and its bubbles is mounted between these magnets and is therefore continuously subject to the bias field.

The rotating field that propels the bubbles through the film is generated by currents in two coils wrapped around the substrate at right angles to each other. These currents are generated by electronic circuits that are part of the magnetic bubble memory system. No mechanical motion is involved.

If power fails, the circuits stop operating, the rotating field disappears, and the bubbles stop moving. But the bias field, generated by the permanent magnet, is not affected. Therefore the bubbles and the data that they represent are maintained in the film. When power is restored the data is again accesible.

BUBBLE MEMORY MANUFACTURING TECHNOLOGY

Bubble memories are produced in a process that resembles semiconductor manufacturing in many ways (Figure 4). Manufacturing begins with a nonmagnetic garnet wafer on which a magnetic film is deposited, using conventional techniques. An ion implantation process alters the magnetization of the top surface of the film, discouraging the formation of abnormal bubbles with undesirable dynamic properties. Then nonmagnetic conductors, bubble-steering patterns of magnetic metal, insulation, passivation, and bonding pads are deposited in much the same way as successive layers on semiconductor integrated circuits. Patterns in each layer are defined photolithographically, just as with semiconductors.

Magnetic bubble technology differs from semiconductor technology in the materials used and in the complexity of the process. Semiconductor circuits use eight or more layers of silicon doped with various materials that affect its electrical characteristics, compared to about three layers of essentially pure metallic and insulating material in bubble technology. These materials are chosen for their magnetic rather than their electrical properties.

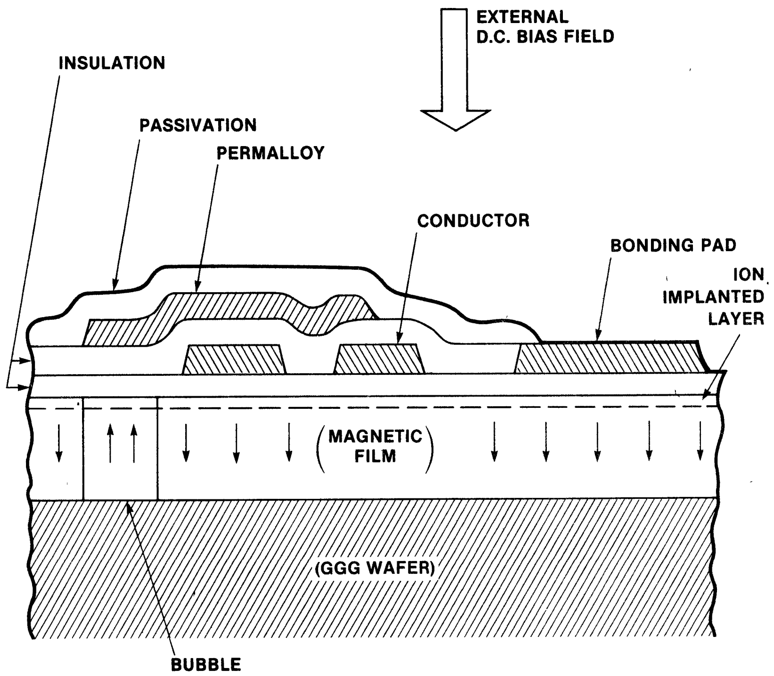


Figure 4. Magnetic Bubble Chip Cross Section

Bubble Memory Functional Description

The Intel 7110 magnetic bubble memory unit contains the bubble chip, the coils that generate the rotating field, two permanent magnets for the bias field, and a magnetic shield that prevents disturbances by external fields and forms a return path for the bias field around the bubble chip (Figure 5).

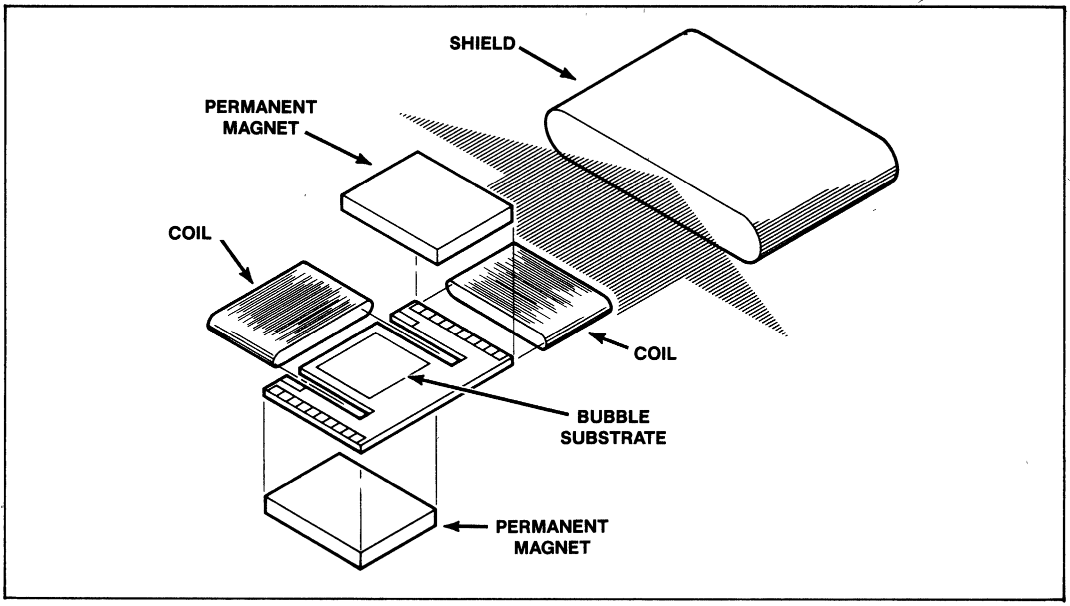


Figure 5. Magnetic Bubble Unit Assembly—Exploded View

Bubble Memory Architecture

Data is stored in the bubble memory unit with a block-replicate architecture (Figure 6). This architecture consists of a number of endless storage loops around which corresponding bits of successive pages continuously circulate, and two tracks, designated input and output, through which the controller writes and reads data in the storage loops. Exchange or replication of data between the tracks and the loops occurs in all loops simultaneously—the key idea in this architecture.

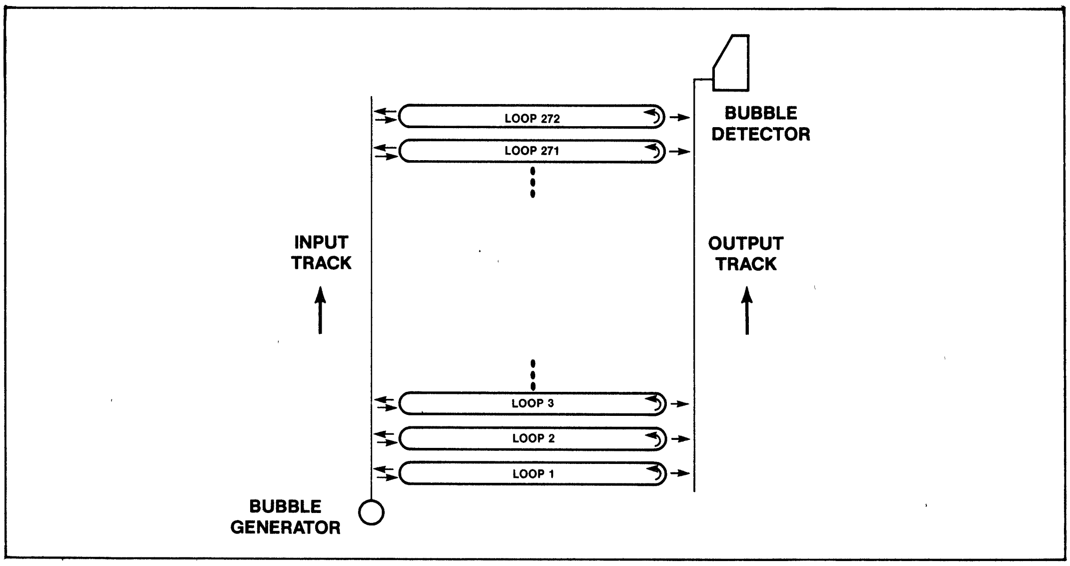


Figure 6. Block-Replicate Architecture

WRITING DATA INTO THE BUBBLE MEMORY

Seed Bubble

The seed bubble, at the beginning of the input track, is generated by an electric current pulse in a hairpin-shaped loop of conductive material. The pulse is strong enough to reverse the bias field locally and thus allow a bubble domain to be created. Once having been created, the seed bubble remains in existence as long as the external bias field is maintained.

The seed circulates under a permalloy patch, driven by the rotating field that propagates bubbles elsewhere in the memory. This bubble is constrained to a kidney shape by interaction of the bias and rotating field with the metal patch (Figure 7). The seed is split in two by a current pulse in the hairpin-shaped conductor. One of them remains under the patch as the seed, quickly regaining its original size; the other one, driven by the rotating field, is transferred to the input track section of the chip. The current pulse that splits the seed is generated to store a binary 1 in the memory; to store a 0, the pulse is omitted, and no bubble is generated.

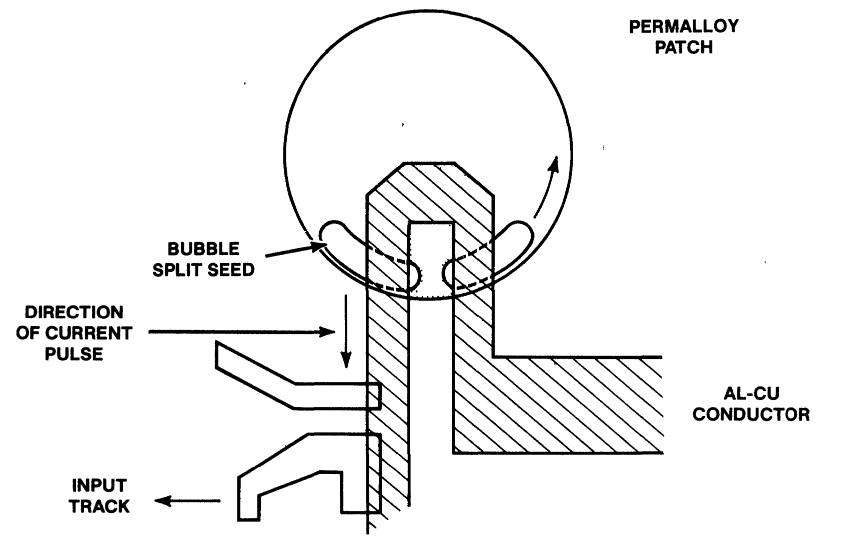


Figure 7. Seed Bubble and Bubble Generation

A seed bubble is maintained at one end of the input track. Bubbles corresponding to binary 1's in the input word are split from the seed and propagate along the input track. When the input track contains exactly one page (64 bytes) then the bubbles exchange places with old bubbles previously circulating in the loops. This is accomplished by an operation called swapping. Thereafter the new bubbles circulate, while the old bubbles now in the track propagate to the end and are destroyed.

Swapping

Transfer of data from the input track to a storage loop involves a swap, bringing the old data onto the input track for destruction at the end of the line, while the new data takes its place in the loop. This is done when a current pulse in an associated conductor under the chevrons causes a bubble to jump from the input track to the storage loop and vice versa. The swap pulse is essentially rectangular, preserving the bubble without cutting it in two.

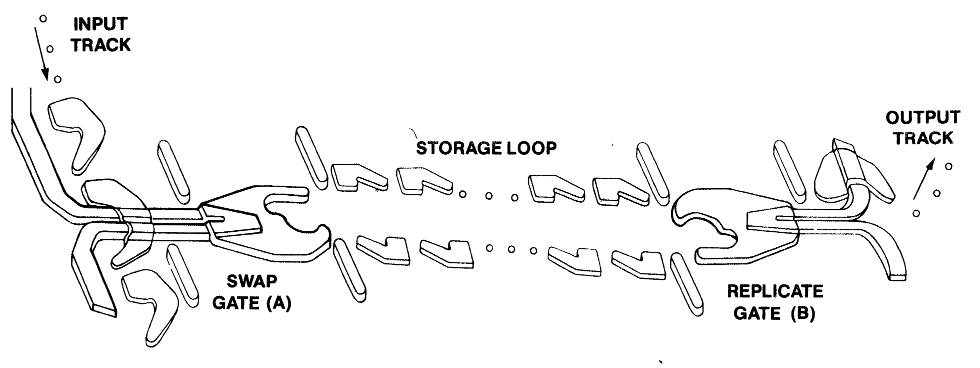


Figure 8. Swapping and Replication Configuration in Bubble Memory

READING DATA STORED IN THE BUBBLE MEMORY

To read the stored data, the circulating bubbles are replicated, one bubble or one unoccupied bubble site from each loop, onto the output track, after which they propagate to a bubble detector at its far end. After detection, these output bubbles are also destroyed. Meanwhile, the data in the loops continues to circulate, permitting a particular page to be read out repeatedly without regeneration, and protecting the stored data if power fails.

Replication

Data is transferred from the storage loop to the output track by replication, continuing to circulate in the loop after having been read out.

For replication, the bubble is propagated under a large element where it is stretched out. As it passes under a hairpin shaped conductor loop it is cut by a current pulse just as in bubble generation.
The replicating current pulse waveshape has a high, narrow leading spike for cutting the original bubble in two, and a lower and wider trailing portion during which the new bubble moves under the output track. The entire pulse lasts about one-quarter of a cycle of the rotating field. In this manner the data in the storage loops is replicated onto the output track, and yet retained in the storage loops in case of a sudden power failure.

Near the end of the output track is a bubble detector—essentially a magnetoresistive bridge formed by interconnecting the permalloy chevrons to make a continuous electrical path of maximum length (Figure 9). As bubbles pass under the bridge, the resistance changes slightly, modulating the currents through the bridge and creating an output voltage of several millivolts. Bubbles are stretched at right angles to the direction of propagation by adding parallel rows of chevrons; these stretched bubbles generate larger output signals at the detector. Beyond the detector, the output track runs the bubbles into the guard rail and destroys them.

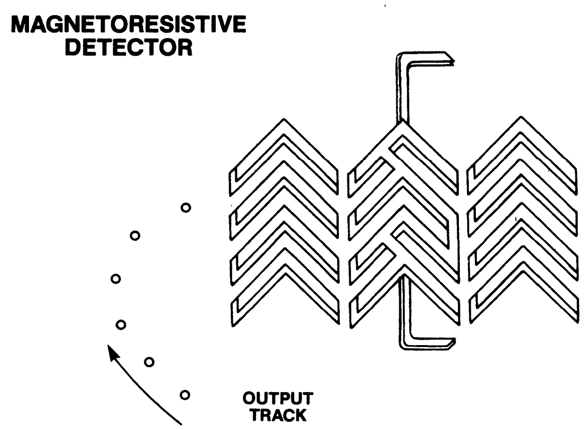


Figure 9. Bubble Detection

Redundancy

The Intel magnetic bubble memory unit physically stores data in 320 storage loops, with capacities of 4,096 bits each. Of the 320 loops, 272 are actually used (active) and 48 are spares (inactive); the boot loop records which loops are used.

Boot Loop

Some of the loops of an individual memory are set aside as spares. The decision as to which loops are to be used (active) and which are not to be used (inactive) is made after the memory unit has been assembled and is undergoing tests at the factory. The outcome of this decision is stored in an extra loop included in each memory chip, in the form of a 12 bit code for each "active" and "inactive" loop.

Whenever power is turned on in the memory system, the system must be initialized before it can be used. Part of the initialization process includes reading the contents of this extra loop, called the boot loop, and placing this information in a bootloop register in the formatter/sense amplifier. From then on, as long as power is on, this register identifies the "active" loops for both reading and writing; "inactive" loops are ignored. The formatter does not attempt to store data in "inactive" loops, and the sense amplifier ignores any data that appears from these loops.

Data Storage—External Appearance

Data is stored logically as 2,048 pages of 512 data bits each. 256 data bits plus 14 error-correction check bits and 2 unused bits are stored in each half of the bubble chip. If automatic error correction is not used, these 16 bits are available for data storage.

Error Correction

Error detection and correction can be performed in the formatter/sense amplifier, which includes a 14-bit cyclic redundancy code that corrects a single burst error of up to five bits in each 270-bit block including the code itself. These code bits are appended to the end of each 256-bit data block when writing into the cell, and checked when the block is read. The error correction feature can be used or not at the user's discretion, by properly setting a register in the bubble memory controller chip. If it is not used, the loops occupied by the code bits become available for additional data.

Access Time and Data Rate

Bubbles circulate at a rate of 50 kilohertz (the rotating field makes 50,000 complete revolutions per second). Average access time to the first bit of the first page is about 41 milliseconds—half the length of time required for a bubble to make one complete circuit of the loop, plus the time to shift a bubble along the length of the output track.

The 320 active and spare loops are actually in four "quads" of 80 loops each (Figure 10). This arrangement shortens the input and output tracks and thus reduces the read and write cycle times. The quads are separately addressable in pairs; in each pair the quads store odd-numbered and even-numbered bits of a word respectively. There are four seed bubbles and four input tracks, and four output tracks. The four output tracks share two detector bridges in such a way that there can never be bubbles from two tracks in a single detector simultaneously. By this means the four streams of output bubbles are interleaved into two bit streams that are stored in two registers in the sense amplifier. The data in these registers is interleaved again into a single stream transmitted serially to the controller.

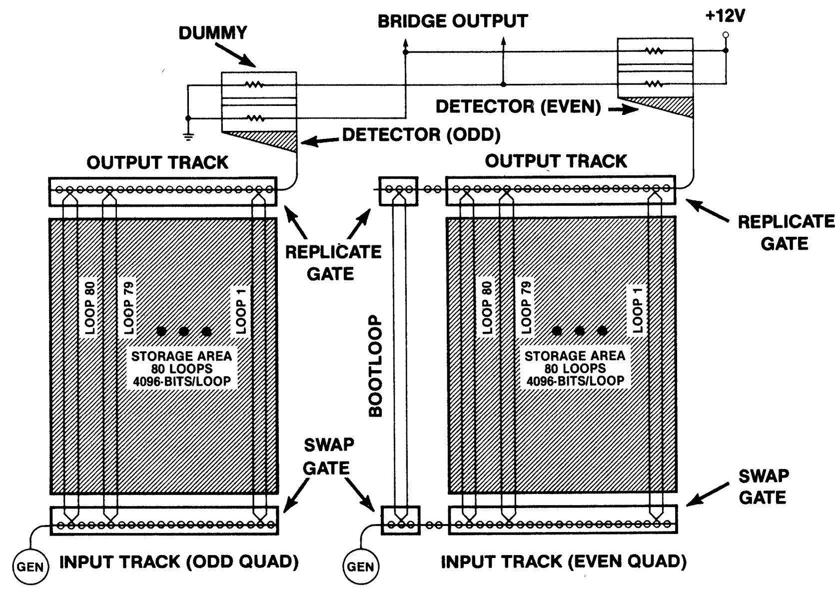


Figure 10. Organization of Bubble Memory (One-Half Chip)

SPECIFIC STRUCTURES OF A MAGNETIC BUBBLE MEMORY

A magnetic bubble memory system consists of a controller and up to eight 1-megabit magnetic bubble subsystems. A minimum system has a controller and one subsystem. The subsystem comprises one magnetic bubble unit in which the data is actually stored, and the peripheral units listed in Table 2 and diagrammed in Figure 11. These circuits are described later in this primer.

Table 2. Components of Intel Bubble Memory System

CONTROLLER	SUBSYSTEM
7220-1 Bubble Memory Controller (for 1 to 8 subsystems)	Memory 7110 Magnetic Bubble Unit Peripheral Units 7242 Formatter/Sense Amplifier 7230 Current Pulse Generator 7250 Coil Predriver 7254 Drive Transistor Assembly (2 required per subsystem)

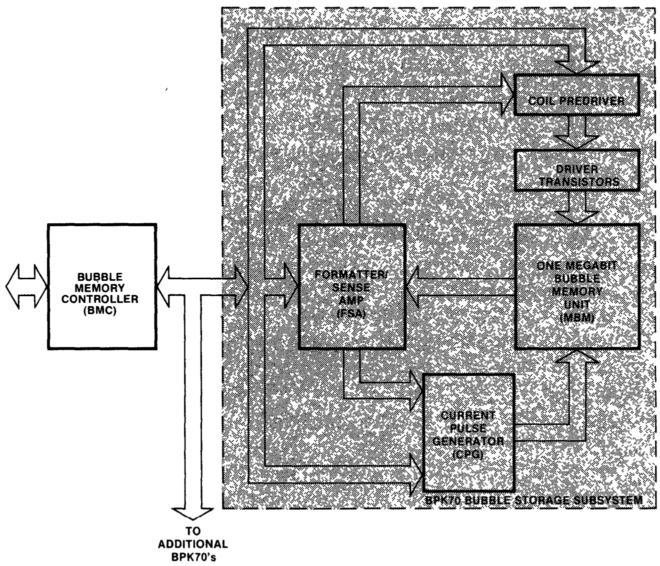


Figure 11. Minimum Magnetic Bubble Memory System, Shaded Portion is Bubble Subsystem

SUPPORT CHIPS

Five semiconductor integrated circuits are necessary to support each bubble chip. These components are described in some detail in the following paragraphs. In addition, each bubble memory system requires a controller, a separate integrated circuit described later.

Formatter/Sense Amplifier (FSA)

Serial data to be stored in or read from the bubble memory passes through the FSA. The FSA keeps track of which loops in the bubble memory are spares, executes the error correction coding and decoding if it is implemented, and shifts data to the bubble memory input tracks or from the output tracks, amplifying the output signals from the memory.

The FSA has a chip-select input, which is normally grounded (permanently enabled). However, each FSA drives the chip-select input of other circuits associated with the same bubble chip, so they are all enabled at the proper time.

Current Pulse Generator (CPG)

All signals except those that control the rotating field originate in the CPG. This device is the source of a current pulse that cuts a new bubble from the seed bubble whenever the FSA has a binary 1 to be stored. Later, when this bubble reaches the loop in which it is to reside, the CPG issues the signal that swaps it with the bubble or non-bubble previously stored in that location of the loop. When data is to be read, the bubble is replicated on the output track by still another signal from the CPG.

Coil Predriver (CPD)

Four digital signals (positive and negative versions of both X and Y waveforms) are sent to the CPD from the controller with appropriate durations and phases to control the rotating field that moves the bubbles in the memory. The CPD combines and inverts these to form eight pulsed outputs that are amplified in a separate transistor package to drive the coils surrounding the bubble chip with a triangular current waveform.

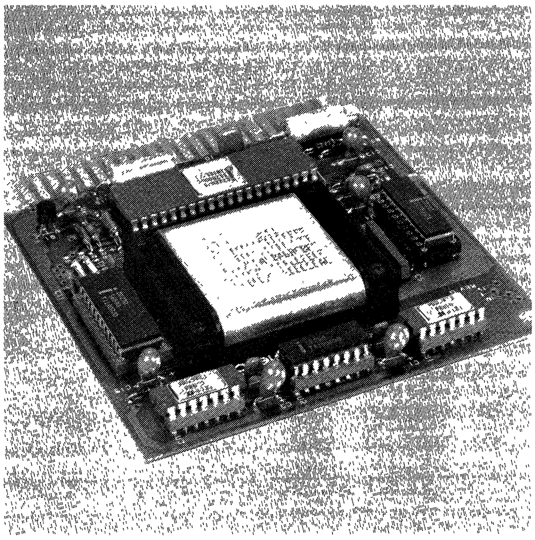


Photo 2. The Minimum Magnetic Bubble Memory System Including Controller

CONTROLLER

The bubble memory controller is the interface between the memory system and the equipment it serves. It converts serial data to parallel and parallel data to serial, and generates all timing signals required by the other support circuits in the bubble memory system. It can control up to eight bubble subsystems, for a total of a megabyte of memory.

Internal storage on the controller includes a first-in-first-out buffer with a capacity of 40 bytes. This buffer stores data to be sent serially to the FSA or just received from the FSA on one side, and data to or from the parallel bus served by the bubble memory on the other. It also serves as a speed matching device between the user at the parallel bus and the FSA which must transfer data to and from the bubble device at exactly the rotating field ratio in each channnel.

GLOSSARY

Bias field—a magnetic field perpendicular to a magnetic thin film that maintains conditions necessary to support formation of magnetic bubbles in the film.

Boot loop—in a magnetic bubble memory with serial/parallel/serial architecture and redundant loops, a special loop containing information that identifies which loops are active and which are inactive, as determined by factory test. This loop also contains the information necessary to synchronize the bubble memory page locations with the controller after power up.

Bubble, magnetic—a cylindrical magnetic domain in a thin film of orthoferrite or garnet. When viewed from above, the cylindrical shape appears spherical, hence the name "bubble." A bubble represents a binary 1 in most magnetic bubble memories.

Chevron—one of many possible shapes for a magnetic pattern deposited on a thin film to steer bubbles in a desired direction. Asymmetric chevrons are used in Intel memories.

Detector—a means of distinguishing bubbles from non-bubbles (1s from 0s) when a word is read from the bubble memory.

Domain, magnetic—a small region of a ferromagnetic substance that contains many similarly oriented atoms, so that the region as a whole is magnetized in that direction.

E²PROM—an acronym for electrically erasable programmable read-only memory, which is a memory component that, though nominally read-only, can accept changes to any work stored in it by electrical means, but at substantially slower speed than that at which stored words are read.

EPROM—an acronym for erasable programmable read-only memory, which is a memory component that, though nominally read-only, can be completely erased, usually by exposure to ultraviolet light, and then reloaded with new information, but at substantially slower speed than that at which stored words are read.

Ferrite—any of several compounds of iron, oxygen, and another metal, with magnetic properties that are useful in certain microwave applications and in computer memories.

Field, magnetic—a region of space in which a magnetic force exists and can be measured.

Garnet—a naturally occurring silicate mineral sometimes used in jewelry. Synthetic garnets with the same crystal structure can be made of oxides of iron and yttrium or one of the rare earths. Garnet is the preferred material for the thin magnetic film in a bubble memory.

Input track—a series of magnetic metal patterns that control the movement of bubbles in a thin film, and thereby lead them from a bubble generator toward one or more storage patterns.

Ion implantation—a process involving accelerators, similar to the machines used by nuclear physicists, for depositing dopants on and just below the surface of an electronic component; used to alter the physical properties of the material.

Latency—a delay between a request to read or write data in a memory and the actual beginning of the operation, imposed by a requirement for the address to move physically (but not necessarily mechanically) to a point where the data transfer can take place.

Magnetization vector—an expression of the magnitude and direction of a magnetic field at a point in space.

Magnetoresistance—a change in electrical resistance due to the presence of a magnetic field.

Major loop—in a magnetic bubble memory, an endless loop containing a bubble generator, a bubble detector, and/or a bubble annihilator, through which data is read or written, and which transfers bubbles to or from one or more minor loops (q.v.) in which they are stored. In some designs the major loop is not endless, and all bubbles not transferred out of it collapse when they reach the end. In these cases the major loop becomes an input or output track (q.v.).

Minor loop—in a magnetic bubble memory, an endless loop in which bubbles are stored, having been transferred into it from a major loop or input track (q.v.) and accessible by transfer into a major loop or output track (q.v.).

Non-Volatility—a property of some memory technologies that retains the integrity of stored data when power is turned off.

Orthoferrite—one of several oxides of iron and either yttrium or a rare earth. The molecular structure is simpler than that of garnet (q.v.). Orthoferrites were the first materials used for the thin magnetic film in experimental bubble memories, but have yielded to garnets, which have more desirable properties—notably ease of preparation as thin films with the necessary magnetic characteristics.

Output track—a series of magnetic metal patterns that control the movement of bubbles in a thin film, and thereby lead them from one or more storage patterns toward a bubble detector.

Permalloy—an easily magnetized and demagnetized alloy of nickel and iron.

PROM—acronym for programmable read-only memory—a read-only memory whose content is loaded by the user after delivery, as opposed to read-only memories whose content is fixed during manufacture. Once loaded, the data in a PROM is not alterable.

Pseudo-random access—a property of some memory technologies in which the time of access to blocks of stored data is largely (but not necessarily entirely) independent of the position of the block in the storage medium, but in which the time of access to bits, words or other entities depends on the position of that entity within the block.

Random access—a property of some memory technologies in which the time of access to any stored bit, word, or other entity is wholly independent of that entity's position in the storage medium.

Saturation—a state of magnetization of a material by a field such that, if the field is increased, the magnetization of the material does not increase and the magnetic flux density increases in proportion to the field (having increased much more rapidly in weaker fields).

Seed—a permanent bubble in a magnetic bubble memory, from which other bubbles are cut to represent stored binary 1s.

Serial access—a property of some memory technologies in which the time of access to any stored bit, word, or other entity depends strongly on that entity's position in the storage medium.

Thin film—any film of material deposited on a suitable substrate to take advantage of the material's special properties when dispersed as a film. Thickness ranges usually from about 10-9 to 10-6 meter, and occasionally to 10-5 meter or more, as in bubble memories.

T-I bar—one of several possible shapes for a magnetic pattern deposited on a thin film to steer bubbles in a desired direction, consisting of shapes like the letter T and the letter I alternately along a track. This pattern was used extensively in early bubble memory designs, but is no longer generally employed.

APPLICATION NOTE

AP-119

December 1982

Microprocessor Interface for the BPK 72

Paul Wells
Application Engineer

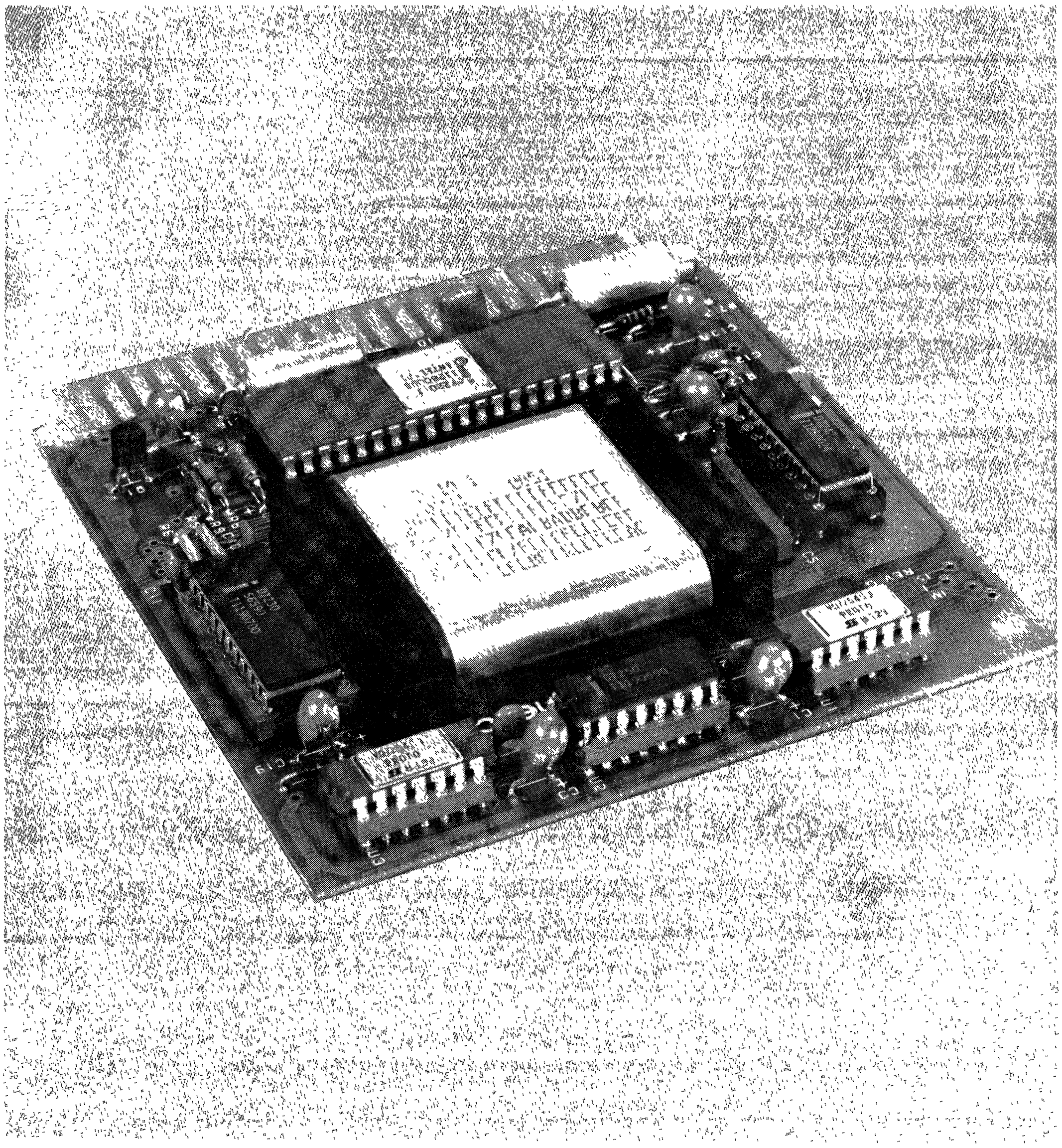
ORDER NUMBER: 210367-002

INTRODUCTION

To date, a major obstacle in the implementation of bubble memories in systems has been the inherently complex control requirements imposed by the bubble memory devices themselves. With the advent of Intel's BPK 72 bubble memory prototype kit, a design engineer can immediately realize the benefits of non-volatility, form factor, density and reliability without the complex control concerns. This application note provides additional background on the operating characteristics of the BPK 72 and is intended to further ease the design effort required in the implementation of bubble memory systems.

OVERVIEW

This application note provides an example of Bubble Memory system implementation using the BPK 72 and an Intel 8086 microprocessor. Before looking at this example, some explanation is necessary as to how this implementation was attained and how a user can take advantage of the principles involved.



As an introduction, the basic architecture of the BPK 72 is reviewed followed by an explanation of the operating characteristics of the BPK 72 kit as a whole and of the 7220 Bubble Memory Controller. Once the building blocks are in place, a detailed account of the implementation of a bubble memory kit is offered. The final section, which involves the actual implementation of the BPK 72 and an SDK-86, completes the application note.

BUBBLE SYSTEM OVERVIEW

A block diagram of the Intel Magnetics 128K-byte system is shown in Figure 1. The support circuitry used with one 7110 magnetic bubble memory (MBM) in the BPK 72 kit consists of the following integrated circuit components: one 7250 Coil Predriver, two 7254 Quad VMOS Drive Transistor packs, one 7230 Current Pulse Generator, and one 7242 Formatter/Sense Amplifier. The 7220 Bubble Memory Controller (BMC) completes the basic system.

The 7250 and the two 7254s supply the drive currents for the in-plane rotating magnetic field (X and Y coils) that move the magnetic bubbles within the MBM. The 7230 supplies the current pulses that generate the magnetic bubbles and transfer the bubbles into and out of the storage loops of the MBM.

The 7242 accepts signals from the bubble detectors in the MBM during read operations, buffers the signals and performs data formatting tasks that include the transparent handling of bootloop information. During write operations, the 7242 enables the current pulses of the 7230 that cause the bubbles to be generated in the 7110 MBM. Automatic error detection and correction of the data can be performed by the 7242.

The 7220 provides the user interface, performs serial-to-parallel and parallel-to-serial data conversions, and generates all timing signals necessary for the proper operation of the MBM support circuitry.

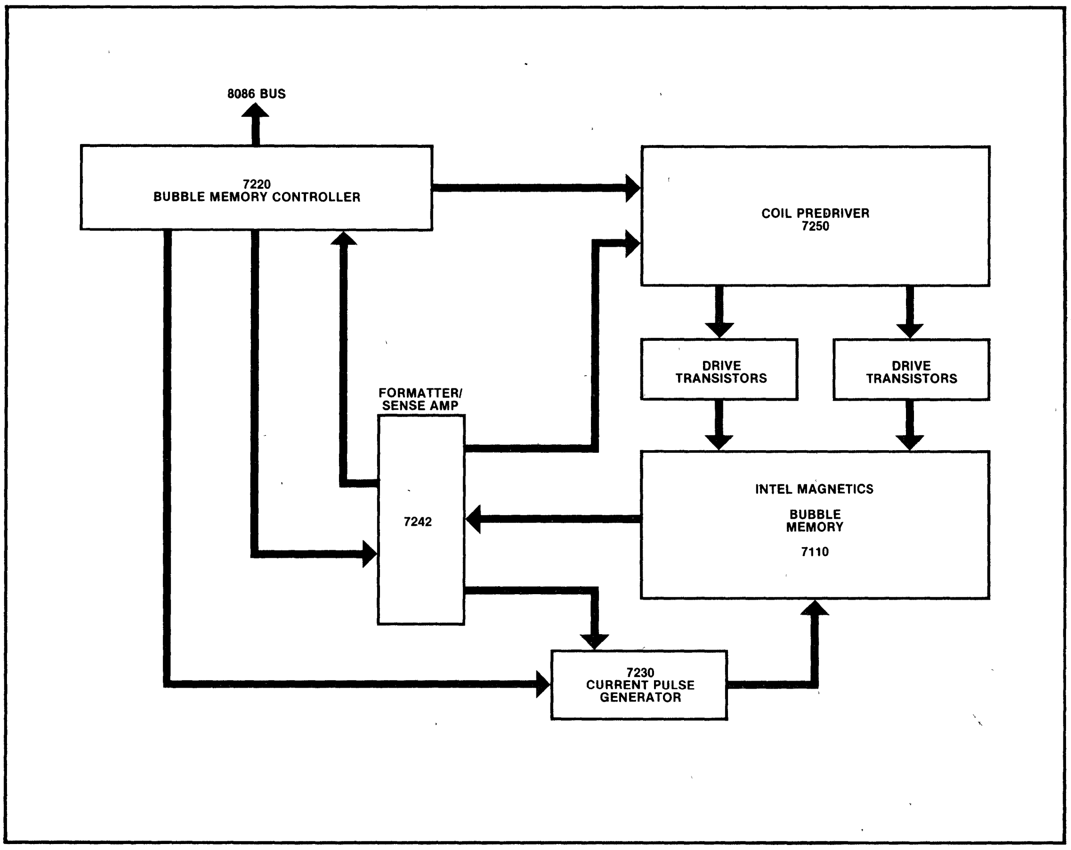


Figure 1. Block Diagram of the 128K Byte Magnetic Bubble Memory System

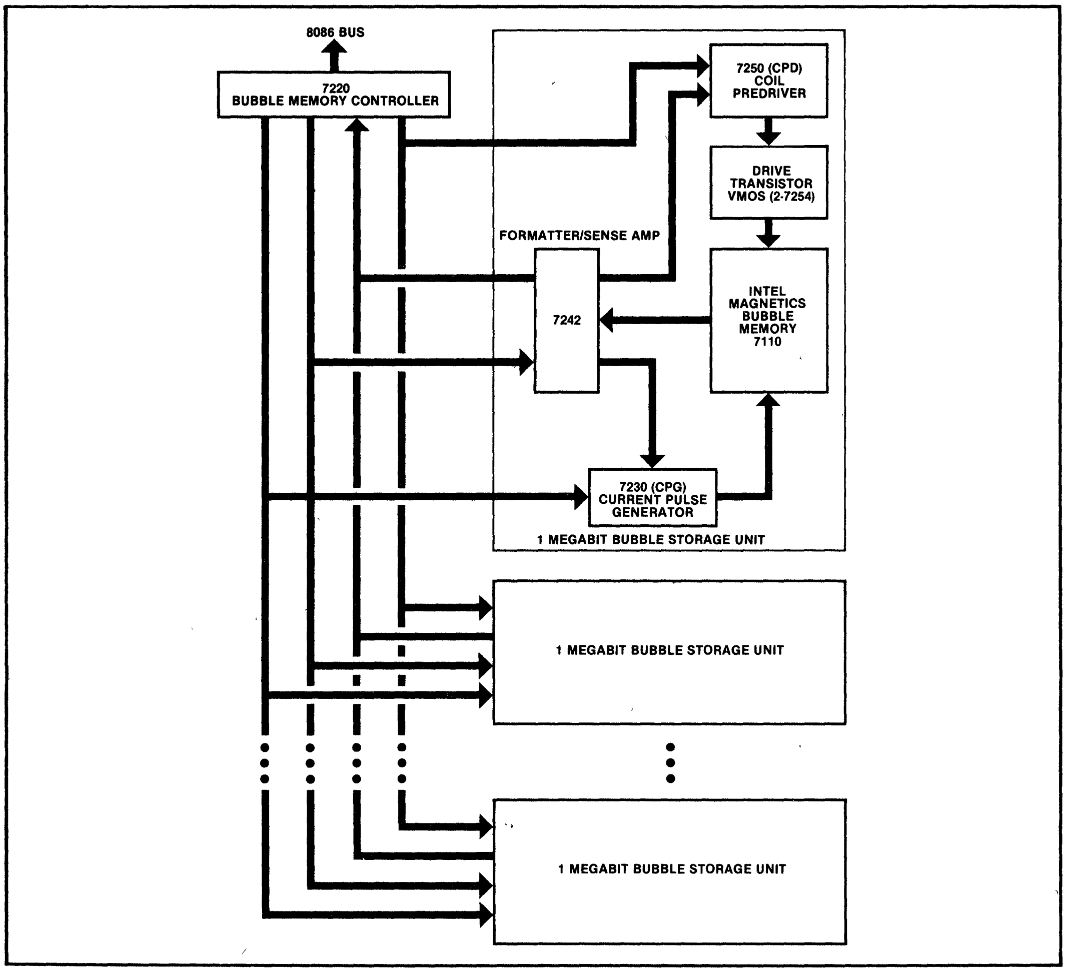


Figure 2. Bubble Memory System Expansion up to One Megabyte

Figure 2 shows how larger systems can be built from the basic components. A Bubble Storage Unit consists of one 128K-byte MBM and the five support chips shown. The components needed for one MBM cell are available as the BPK 70 kit. Larger systems can be constructed from the components supplied with one BPK 72 kit (which includes the 7220 controller) and one or more BPK 70 kits. For example, a one megabyte system can be assembled from one BPK 72 kit and seven BPK 70 kits. No additional TTL parts are required when building multibubble systems with up to eight MBMs.

One 7220 is capable of controlling up to eight Bubble Storage Units simultaneously. Larger systems can be configured with multiple 7220's and additional Bubble Storage Units.

Functional Organization of the 7110 Bubble Memory

The Intel Magnetics 7110 Bubble Memory utilizes a "major track/minor loop" architecture. With this architecture, if a binary 1 is to be written, a "seed bubble," always present in the 7110, is split in two. One bubble remains at the generator as the seed, and the other is propagated down the input (major) track. If a 0 is to be written, the seed bubble is not duplicated. The data generated is sent down the input track, in serial, until it is aligned with the "swap" gates at the minor loops of the device. The new data is then swapped into the minor loops in parallel at the same time the old data is swapped out to the major track.

To read data from the 7110, data is rotated in the minor loops until it is positioned at the "replicate" gates opposite the output track. On receipt of a replicate signal, the data in the minor loops is duplicated by splitting the bubbles. The original data remains in the minor loops, and the duplicate data is clocked down the output track where the detector elements of the bubble memory operate to transform the presence or absence of a bubble into small electrical signals that are converted into digital '1' and '0' signals in the 7242 FSA.

With the 7110, the process of reading data from the minor loops by simultaneously splitting all of the bubbles in a page is known as "block replicate." The advantage of the block replicate architecture is that the data currently stored in the minor loops is not compromised during a read operation; the data to be read never leaves the minor loops. This architecture can be contrasted with earlier architectures that required the data to leave the minor loops, be detected and then returned to the minor loops. In the event of a power failure, bubble systems not utilizing the block replicate architecture could suffer a loss of data during a read operation; the data being sensed would not be returned from the major loop to the minor loops.

With the 7110 MBM, there are 2048 positions for the data within a minor loop. To move the bubbles in the MBM, a magnetic field is induced and rotated in the plane of the 7110. As the field is rotated 360 degrees, every bubble is moved ahead one position, and all of the bubbles maintain the same position relative to one another. All of the bubbles in similar positions in the loops are referred to as a "page."

By way of illustration, suppose the bubble is made of five minor loops (a,b,c,d,e) capable of holding nine pages of data (Table 1). During four 360 degree "rotations" of the in-plane magnetic field, the nine pages of data shift four positions (1.1, 1.2, 1.3, 1.4).

Table 1. 7110 Loop Operation

abcde	abcde	abcde	abcde
00000	00011	00000	00000
00011	00000	00000	11111
00000	00000	11111	00000
00000	11111	00000	00000
11111	00000	00000	00000
00000	00000	00000	10110*
00000	00000	10110*	00000
00000	10110*	00000	00011
10110*	00000	00011	00000
1.1	1.2	1.3	1.4
* = page zero			

The 7110 MBM actually contains 320 minor loops, of which 272 must be good. The additional 48 loops provide 15% redundancy. This redundancy factor allows some of the loops in the 7110 to be bad while maintaining a completely functional one megabit device. A map of the good and bad loops is placed on the label of the 7110 and is also

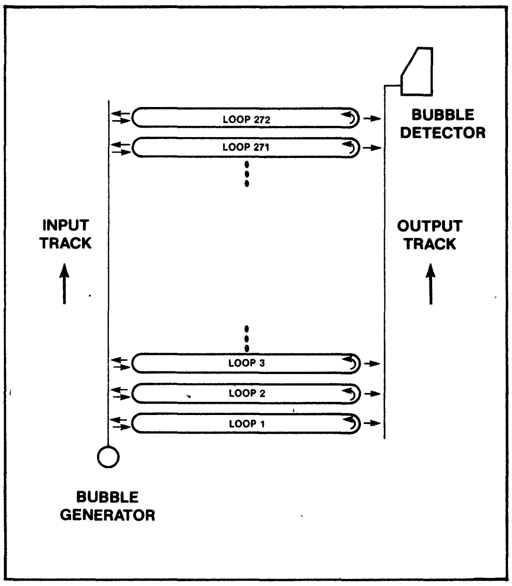


Figure 3. Functional Organization of the 7110

encoded and placed in the boot loop of the device as it is tested. This map, the bootloop, consists of forty bytes of data. Each good loop in the 7110 is represented by a one, each bad loop by a zero. When the system is initialized, the 7220 BMC reads the bootloop from the 7110 and decodes it. The bootloop is then automatically placed in the bootloop register of the 7242. The bootloop register serves as a working 'map' of the 7110 for read and write operations.

With the pages of data rotating around the minor loops, there must be a mechanism to orient the device and to assign a starting address to a page. The mechanism used to identify page zero involves the bootloop that resides on the 7110. Page zero (or address zero) is defined as the position of the 7110 after the bootloop has been read by the 7220 controller. Thus, each time the host CPU sends an "initialize" command, the bootloop is read by the 7220, and the 7110 is queued at page zero. From this point, any desired page in the bubble can be obtained by the controller.

Data Flow Within the Bubble Memory System

To better understand the relationship between the 7110 MBM and its support circuitry, the data flow within the bubble system during a read operation is examined. During the read operation, bubbles from the storage loops are replicated onto an output track and then moved to a detector within the MBM. All movements within the MBM occur under the influence of a rotating magnetic field; the number of rotations and the rotation timing are under the control of the 7220 BMC. The detector outputs a differential voltage according to whether a bubble is present or absent in the detector at any given time. This voltage is fed to the detector input of the 7242 Formatter/Sense Amplifier (FSA).

The data path between the 7110 MBM and the 7242 FSA consists of two channels (channel A and channel B) connected to the two halves of the MBM. When data is written, the bit stream is divided with half of the data going to each side of the MBM. During a read operation, data from each half of the MBM goes to the corresponding channel of the FSA. In the FSA, the sense amplifier performs a sample-and-hold function on the detector input data, and produces a digital 0 or 1. The resulting data bit is then paired with the corresponding bit in the FSA bootloop register.

If an incoming data bit is found to be from a good loop (a corresponding "1" in the FSA bootloop register), it is stored in the FSA FIFO; otherwise, it is ignored. This process continues until both FSA FIFOs (channels A and B) are filled with 256 bits. Error detection and correction, if enabled by the user, is applied to each block of 256 bits at this point. If error correction is not enabled, 272 bits of data can be buffered in each FIFO.

As data leaves the 7242 FSA, the bit patterns buffered in each of the FSA FIFOs is interleaved and sent to the 7220 BMC in the form of a serial bit stream via a one-line bidirectional data bus (DIO line). In the 7220 BMC, the data undergoes a serial-to-parallel conversion and is assembled into bytes that are buffered in the 7220 FIFO. It is from this FIFO that the data is written onto the user interface.

COMMUNICATING WITH THE 7220

The CPU views the 7220 BMC as two input/output ports on the bus. When the least-significant bit of the address line is active (A0 = 1), the command/status port is selected. When the least-significant bit of the address line is inactive (A0 = 0), the bidirectional data port is selected. In order to define the operations on these ports, it is necessary to understand something of the internal organization of the 7220 Bubble Memory Controller.

For simplicity, the user need only view the 7220 as containing a 40-byte FIFO and a collection of 8-bit registers. The FIFO is a buffer through which data passes on its way from the 7242 Formatter/Sense Amplifier (FSA) to the user, or from the user to the FSAs. The primary purpose of the FIFO is to reconcile differences in timing requirements between the user interface to the 7220 controller and the controller interface to the FSAs.

The six 8-bit registers internal to the 7220 are loaded by the user prior to any operation of the bubble system and contain information regarding the operating mode of the 7220. Loading the 7220 registers before any commands are sent is similar to passing parameters to a subroutine prior to invocation, hence, the registers are often referred to as "parametric registers."

Data transferred between the CPU and the 7220 FIFO and parametric registers takes place over an 8-bit data port. The choice as to whether the data is destined for the FIFO or the parametric registers, however, is made through the command/status port. In one case, the actual commands that cause some operation to take place, such as a read or write, consist of a 4-bit code sent by the CPU to select one of 16 possible commands. This 4-bit code occupies the low-order nibble (bits 0, 1, 2, and 3) of the command byte. The command byte must also have bit 4 set to indicate to the 7220 that a command is being sent. In the second case, another 4-bit code on the command port (bits 0, 1, 2, and 3) is used to select either one of the parametric registers or the 7220 FIFO. As shown in Table 2, if bit 4 of the command byte is set to zero, the value of the low-order nibble is taken to be a pointer value that specifies a parametric register or the 7220 FIFO. This pointer is referred to as the "Register Address Counter" (RAC).

Table 2. Command Port Function

FUNCTION	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
Command	0	0	0	1	C	C	C	C
RAC	0	0	0	0	R	R	R	R

RAC values that may be sent out on the command port and the corresponding register names are illustrated in Table 3. The RAC points to, or selects, six unique registers and the 7220 FIFO. Once a RAC value is sent by the CPU to the 7220 via the command port, the next read or write operation to the data port transmits data to or receives data from the register addressed. Notice that the six registers have values that are in ascending order starting at 0AH and that the FIFO has a value of 0.

The reason for this ordering is due to the auto-incrementing feature of the RAC; once the first register is selected, each subsequent byte of data on the data port causes the RAC to be automatically incremented and to point to the next register in the sequence. Once the most-significant byte of the Address Register has been loaded, the RAC value automatically rolls over from 0FH to 0 and points to the 7220 FIFO. The system is now in position to transfer data to or from the FIFO without the user code explicitly pointing to the FIFO.

Table 3. Register Address Counter Assignments

Register Name	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Read/ Write
Utility Register	0	0	0	0	1	0	1	0	R/W
Block Length Register (LSB)	0	0	0	0	1	0	1	1	W
Block Length Register (MSB)	0	0	0	0	1	1	0	0	W
Enable Register	0	0	0	0	1	1	0	1	W
Address Register (LSB)	0	0	0	0	1	1	1	0	R/W
Address Register (MSB)	0	0	0	0	1	1	1	1	R/W
7220 FIFO	0	0	0	0	0	0	0	0	R/W

Once the FIFO has been selected, the RAC stops incrementing and continues to point to the FIFO until changed by the user software. This sequence minimizes the number of instructions necessary for a given transaction and aids in establishing a protocol to ensure that all of the necessary information is sent to the controller. The user, however, is not bound to follow this automatic sequence. Each parametric register may be selected and loaded in any order; specific registers may be updated where needed, but in each case, the host software must explicitly name the register to be loaded. Until a user is familiar with the bubble system, it is recommended that the auto-incrementing feature be used.

It is important to remember that once a command has been given to the 7220 BMC, the parametric registers must not be updated until the Status byte indicates that the operation is complete. The parametric registers are, in effect, working registers for the controller during the execution of a command. For example, during a Read or Write operation, the Block Length Register, which contains the terminal page count for the operation, is decremented by the 7220. Similarly, the Starting Address Register, which initially contains the starting page for an operation, is incremented by the controller as each page is transferred. Attempting to modify these registers during the operation of a command causes the block count and address to be incorrect.

Addressing the Bubble Memory System

One of the interesting aspects of the Intel Bubble Memory System is its inherent addressing flexibility. The user may treat a 7220 BMC with eight bubbles as a collection of 16K pages of 64 bytes each (addressing each bubble in turn) or as collection of 2K pages of 512 bytes each (addressing eight bubbles in parallel). Of course, there are a variety of configurations in between these two extremes, each dictated by the user's need for speed, power consumption, address space, and cost. Control over the configuration is achieved at run time via two of the parametric registers: the Block Length Register and the Starting Address Register.

The Block Length Register (BLR) is a 16-bit value divided into two fields: the "terminal count" field and the "channel" field. The bit configuration for the BLR is as follows:

Table 4. Block Length Register

```
 channel              terminal count
C C C C X T T T   T T T T T T T T
     MSB                  LSB
```

The "terminal count" field ranges over eleven bits and defines the total number of pages requested for a read or write operation. With eleven bits in the field, a user may request from one to 2048 pages be transferred (eleven bits of zero indicate a 2048-page transfer). The width of the page is effectively defined in the "channel" field. This field specifies the number of FSA channels that are to be addressed. Recalling that each 7242 FSA has two channels to communicate with one 7110 bubble memory, the legal combinations in this field address one channel (one half of a 7110), two, four, eight, or 16 channels. These combinations translate into page sizes of 32, 64, 128, 256, or 512 bytes, respectively. (The one-channel mode of operation is usually reserved for diagnostic purposes, and examples of its use will be illustrated later.)

Table 5 shows the relationship between the "channel" field and the number of FSA channels selected. Notice that the channel field bits are encoded. A value of "0001" binary selects two FSA channels: 0 and 1.

Table 5. FSA Channel Select

	Channel field (BLR MSB bits 7, 6, 5,4)				
	0000	0001	0010	0100	1000
Number of channels selected:	0	0,1	0,1,2,3	0 to 7	0 to F

Thus, a BLR value of "0001" in the high-order four bits selects one bubble through channels 0 and 1 Similarly, a BLR value of "0010" selects two bubbles in parallel with a page size of 128 bytes. This, however, is not the complete story. For example, a value of "0100" in the BLR selects four bubbles in parallel through channels 0 to 7. Suppose, that there are eight bubbles in the system and that the user desires to arrange the eight bubbles as two sets of four. The mechanism to communicate through channels 0 to 7 and channels 8 to F resides with the Address Register (AR).

The Address Register contains a 16-bit value divided into two fields: a "starting address" field of eleven bits and a "magnetic bubble memory (MBM) select" field of four bits.

Table 6. Starting Address Register

```
   MBM Select     starting address
X M M M M A A A   A A A A A A A A
      MSB                 LSB
```

The eleven bits in the starting address field of the AR are set by the user to indicate to the 7220 BMC on which page of a bubble's 2048 pages the transfer is to start. For example, if a read operation is to start at page 1125 and is to continue for 16 pages, the starting address field contains 1125, and a value of 16 is placed in the terminal count field of the BLR. After each page is transferred, the starting address field is incremented and the terminal count is decremented by the controller.

Continuing with the example of two banks of four bubbles, notice in Table 7 that the MBM select field is needed to switch between the two banks. A value of "0000" in bits 3, 4, 5, and 6 of the high-order byte of the address register selects bank 0 or FSA channels 0 through 7; a value of "0001" selects bank 1 or FSA channels 8 through F. Each bank contains 2048 pages of 256 bytes.

To operate eight bubbles serially, a user needs only to specify a value of "0001" once in the channel field of the BLR and to begin with a value of "0000" in the MBM select field. As page 2048 is written in the first bubble, the AR, managed by the 7220 controller, rolls over to 0 and updates the MBM select field with no additional bit manipulation. In this case, the bubble system appears as 16K pages of 64 bytes each. Power consumption is one-eighth of that consumed by operating eight bubbles in parallel. However, the data rate is limited to the data rate of one bubble.

Table 7. FSA Channel Select/MBM Select

MBM SELECT AR MSB BITS (6, 5, 4, 3)	"CHANNEL FIELD" (BLR MSB bits 7, 6, 5, 4)				
	0000	0001	0010	0100	1000
0 0 0 0	0	0,1	0,1,2,3	0 to 7	0 to F
0 0 0 1	1	2,3	4,5,6,7	8 to F	
0 0 1 0	2	4,5	8,9,A,B		
0 0 1 1	3	6,7	C,D,E,F		
0 1 0 0	4	8,9			
0 1 0 1	5	A,B			
0 1 1 0	6	C,D			
0 1 1 1	7	E,F			
1 0 0 0	8				
1 0 0 1	9				
1 0 1 0	A				
1 0 1 1	B				
1 1 0 0	C				
1 1 0 1	D				
1 1 1 0	E				
1 1 1 1	F				

The Enable Register

The Enable register is the parametric register that defines the various modes of operation of the 7220 controller. The data transfer mode (polled, interrupt driven, or DMA operation) is selected by setting the appropriate bit in this register. Likewise, the type of error correction to be applied to the data is selected, based on the bits selected in this register.

While the function of each of the enable register fields is described in the BPK 72 manual, some of the finer points and implications are detailed here.

Note that it is possible to completely change the operating characteristics of the bubble system through software control. A system can go from the DMA mode with error correction enabled to a system operating in polled I/O with no error correction enabled by altering the value of the Enable register. Though most implementations will not take advantage of this degree of flexibility, there are cases where the Enable register is modified during system operation. For example, the normal interrupt and MFBTR bits can be modified between operations to change interrupt and read data rates, respectively. (If the error correction mode is changed, the CPU must issue an Initialize command to the 7220 controller).

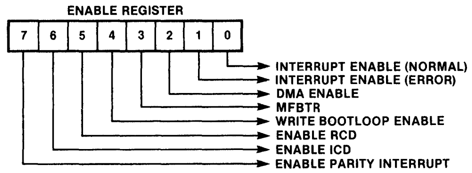


Figure 4. Enable Register Definition

The interrupt capabilties of the 7220 are reflected in the NORMAL, PARITY and ERROR INTERRUPT bits of the ENABLE register byte. The 7220 controller is capable of issuing interrupts to a CPU at the normal completion of an operation, if a parity error is encountered between the 7220 controller and the CPU, or if a data transfer error is found by the 7242 FSA. Any (or all) of these conditions are selected via the Enable register byte, and any resultant interrupts are sent to the CPU via a single INT line. At this point, the software must examine the status register to determine the cause of the interrupt. (An additional interrupt, the FIFO half-full interrupt, is issued on the DRQ pin and is not controlled by the Enable byte).

One of the more difficult aspects of the ENABLE register byte to understand is the operation of the ERROR INTERRUPT bit (bit 2). This bit normally is not used alone, but in conjunction with the ENABLE RCD and ENABLE ICD bits of this register. These three bits form combinations that gate selected 7242 error conditions to the CPU. For example, if, while operating under error correction, a user does not wish to be bothered by an interrupt that indicates an error has been corrected automatically by the system, a specific pattern of these three bits would be selected (100 or 010 from Table 8). If the user wishes to be notified of all errors, another pattern would be selected (011 or 101).

Table 8. Error Correction Combinations

Enable ICD	Enable RCD	Interrupt Enable (ERROR)	Interrupt Action
0	0	0	No interrupts due to errors
0	0	1	Interrupt on TE only
0	1	0	Interrupt on UCE or TE
0	1	1	Interrupt on UCE, CE or TE
1	0	0	Interrupt on UCE or TE
1	0	1	Interrupt on UCE, CE or TE
1	1	0	Not used
1	1	1	Not used

The purpose of the ERROR INTERRUPT bit is not to enable or disable error interrupts, but rather to aid in selecting the type of error interrupt received by the CPU. If any type of error correction is selected, interrupts are enabled automatically.

The ENABLE RCD (read corrected data) bit causes the error correction algorithm to be applied to the data being transferred from the 7110 MBM in an almost transparent manner. The RCD bit allows the 7220 controller to send its own commands to the 7242 FSA. These commands cause the FSA to automatically correct and transfer to the controller, any data that is found to be in error and that is considered correctable.

With only the RCD bit on, no interrupt is generated if a correctable error is found. However, the user is informed that a correctable error was encountered and corrected during the data transfer via the 7220 status byte at the end of the operation. Uncorrectable and timing errors cause an interrupt to which the CPU must respond. With both the RCD bit and ERROR INTERRUPT bit on, the CPU is notified via an interrupt whenever a correctable, uncorrectable or timing error is encountered.

The RCD mode of operation is suitable for transfers where a GO/NO GO termination is sufficient. For example, when loading executable code from the bubble to RAM, it is necessary to know that the transfer was good (with errors corrected) or aborted due to an uncorrectable error.

A retry of an uncorrectable page of data is accomplished by sending another Read command without modifying the parametric registers. It may be the case that the errors encountered were soft (read) errors that may not be present on a retry. Thus, what may have been detected as an uncorrectable error, may become a correctable error (or simply vanish) on a subsequent read of the offending page. In this case, the error correction ability of the system corrects the errors automatically without additional user intervention.

The advantage of the RCD mode of operation is that error correction can be applied transparently to the CPU except for uncorrectable conditions. The disadvantage is that a page of uncorrectable data is passed to the controller before the interrupt is sent. The software must have the ability to clear the 7220 FIFO prior to rereading the offending page from the bubble.

If a given page continues to show up as having a correctable error after a number of retrys, it is up to the user's protocol to determine the action to be taken. One protocol suitable for handling errors involves "scrubbing" the data. Suppose a page appears with an error and, on retry, the error is still present. If the error is correctable, the data should be corrected and written back to the bubble and then read back into RAM. The probability of encountering an uncorrectable error after the first retry is 1 in 10^{16}. Data scrubbing after one retry maintains this level of reliability.

The ENABLE ICD (internally correct data) bit also enables the error correction capability of the bubble system, but allows a slightly different interaction between the 7220 controller and the 7242 FSA than defined for the RCD mode. Error interrupt conditions are the same as defined for RCD operation. With the ICD bit on, correctable errors are handled automatically, but the operation halts for uncorrectable or timing errors. With both the ICD and ERROR INTERRUPT bits on, the operation halts for correctable, uncorrectable or timing errors. The ICD mode differs from the RCD mode in that when an operation halts due to an error, the offending page is held in the 7242 FSA and is not automatically transferred to the 7220 FIFO. Though the difference is subtle, the ICD mode of operation allows more flexibility in error logging and recovery. With data held in the 7242, the number of the bad page can be read for logging purposes, and the data can be recycled through the error correction network or reread from the bubble repeatedly. When the CPU is interrupted due to an error in the ICD mode, the user must look at the 7220 status byte to determine the type of error encountered. If the error is correctable, the user's software sends a Read Corrected Data command (0CH) to the controller. This command causes the controller to issue it's own commands to the 7242 to correct the error and to transfer the data to the 7220 FIFO. (Recall this action is done automatically when the RCD mode is selected; uncorrectable errors can be handled as described above).

As an example of how the ICD mode can be utilized, suppose that during a data transfer in the RCD mode, a correctable error consistently occurs. The error, of course, is automatically handled by the 7242, and the only indication that an error had been corrected is through the status byte at the end of the transfer. There is no information as to how many or in what page the error or errors appear. One way to diagnose the problem is to reread the entire data block in the ICD mode with the ERROR INTERRUPT bit on. The transfer stops at the appearance of any error, and the data remains in the 7242. The page number of the error can be found by reading the Address Register since this register is incremented automatically after each page is read if no error is detected.

The user should then issue an RCD command to the 7220 to allow the page to be corrected and transferred to the 7220. Once the transfer is complete, the enable register again is changed to disable all error correction, and the 7220 is reinitialized. The entire block is read again and compared with the corrected version. (Error correction bits are appended to the data and can be ignored.) If a bad loop is suspected, the bad loop location could be calculated and the bootloop modified.

It is unlikely that repeated correctable errors are sufficient motivation to modify the bootloop. Repeated uncorrectable errors, however, at the same location, might be sufficient reason. Note that modifying the bootloop is an extreme measure and should only be performed as a last resort and only if justified by test data.

The Status Register

The 7220's 8-bit Status register is accessed by reading the Command port (A0 = 1). This register provides information regarding error conditions, the termination of commands, and the readiness of the controller to transfer data or accept new commands.

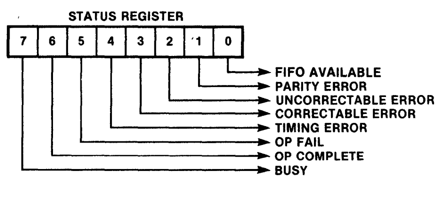


Figure 5. Status Register Definition

Values for the Uncorrectable Error and Correctable Error fields are generated when error correction is utilized as previously defined. The PARITY ERROR bit is set when a parity error is encountered on data sent to the controller on the D_0-D_7 lines. The TIMING ERROR bit is set for a number of conditions. The most frequent cause of a timing error is when the CPU fails to keep up with the rate at which the controller is filling or emptying the FIFO (an overflow or underflow condition). With one bubble in the system and the MFBTR bit of the Enable byte set to one, the controller moves data to or from the FIFO at a rate of about one byte every 80 microseconds. With eight bubbles operating in parallel, the rate is about one byte every 10 microseconds. (With the MFBTR bit set to 0, the data rate on a one page transfer or the last page of a multipage transfer is four times these rates.) Once a Read or Write command is issued, if the CPU cannot meet these transfer requirements, a timing error results.

Another way in which a timing error occurs is when the proper number of bits is not set in the bootloop register of the 7242 FSA. The 7242 must have 272 loops active to operate properly (270 with error correction enabled). If a mistake is made either when the bootloop of the 7110 is written or if the bootloop register is loaded incorrectly from RAM by the user, a timing error results. A timing error also occurs if the Write Bootloop command is issued to the 7220 controller and the WRITE BOOTLOOP ENABLE bit of the Enable byte is not on. Finally, a timing error is generated if the bootloop synch code is not found when a Read Bootloop or Initialize command is issued.

The OP FAIL and OP COMPLETE bits of the status register simply indicate the state of an operation after a command is executed. If an operation fails (OP FAIL = 1), the cause can be determined by looking at the other error bits of the status byte. When an operation (command) terminates successfully, the OP COMPLETE bit is set, and the status register shows a 40H.

The FIFO AVAILABLE bit of the status byte is more complex than the other bits since its meaning can change depending on the type of operation being performed as outlined below.

From an operational point of view, the FIFO AVAILABLE bit acts as a gate for the FIFO handling software. During a write operation, if the FIFO bit is set (1), there is room for more data; if the FIFO bit is clear (0), the FIFO is full. During a read operation, if the FIFO bit is set, data has been placed in the FIFO by the controller; if it is clear, the FIFO is empty.

Table 9. FIFO Available Bit Semantics

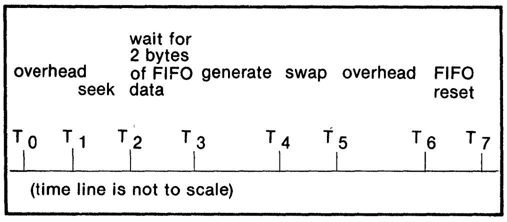
FIFO AVAIL BIT	BUSY = 1 & writing	BUSY = 1 & reading	BUSY = 0 & reading
1	room for data	data avail.	data avail.
0	no room for data	no data	no data

Note that it is possible to complete an operation with data still remaining in the FIFO (indicated by a 41H status value). This condition is quite legal; it is up to the software to remove the data or to issue a FIFO RESET command.

The BUSY bit indicates when the controller is in the process of executing a command. When a command is sent, the BUSY bit goes active within a few microseconds after the command is received and remains active until the operation either completes or fails. It is important to note that the BUSY bit remains active until all other bits in the status byte have been set. Thus it is possible to see logically-exclusive conditions such as BUSY and OP COMPLETE at the same time. The key to interpreting the status byte is to consider the status byte valid only after the BUSY bit returns to an inactive level. The single exception to this rule is the FIFO AVAILABLE bit.

The action of the controller during a write operation is one of the more complex sequences and serves as a good illustration of the behavior of the BUSY and FIFO AVAILABLE bits. Suppose a Write command is sent to transfer an arbitrary number of pages. Table 10 shows the activity of the controller at various steps in the sequence.

Table 10. Stages of a Write Command



Before the Write command is sent, the FIFO is in a general-purpose mode and remains in this mode until T_2. When the command is sent at T_0, the BUSY bit is low and, in fact, the BUSY bit must be low in order for the controller to accept a new command (except Abort). Sometime between T_0 and T_1, the BUSY bit goes high. Thus, between T_1 and T_2, the status byte will be 80H.

At T_2, the FIFO is internally placed in the "write mode," and FIFO AVAILABLE changes meaning from "FIFO has data" to "FIFO has room". For proper operation, the FIFO must be empty prior to issuing the WRITE command. This condition can be guaranteed by using the FIFO Reset command. Assuming the FIFO is empty, at T_2 the status byte changes from 80H to 81H. The status byte remains at 81H until T_6 (unless the CPU is able to fill the FIFO in which case, the FIFO AVAILABLE bit toggles between 0 and 1).

At T_7 (the completion of the command), the status byte should be 40H if the CPU did not load data between T_6 and T_7. If data was loaded during this interval, the status value is 41H.

Notice that if the FIFO contains data when the Write command is sent, the CPU can, by mistake, overflow the FIFO during the "seek" portion of the command. This condition results from the FIFO AVAILABLE bit being a "1" due to data present in the FIFO, not because there is room in the FIFO. While the following diagnostic routines take advantage of the "preloading" ability of the FIFO, the examples of operational software at the end of this application note do not preload the FIFO.

7220 Commands

The 7220 command set consists of 16 commands identified by a 4-bit command code. The function of most of the commands is obvious from the command name (e.g., Initialize, Abort, Read, Write). These commands are adequately described in the BPK 72 manual. There are, however, some commands and protocols that merit additional discussion (specific examples are covered later in this document).

Table 11. 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

In general, all commands sent to the 7220 controller must be preceded by the setting of the parametric registers. While there are some exceptions as with the Abort command, it is usually necessary to supply operating information to the controller via the parametric registers prior to issuing any command. Since many initial problems stem from failing to load the registers prior to issuing commands, the user software should never assume that the regsiters contain valid data.

After the bubble system has been powered up, the 7220 controller inhibits (or ignores) all commands except an Initialize or Abort command. One of these commands must be sent prior to issuing any other command. Normally, the first command issued after loading the parametric registers is the Initialize command. This complex command reads and decodes the bootloop information from each bubble in the system and places this information in the bootloop register of the corresponding 7242 FSA. Pointers internal to the 7220 automatically are prepared for normal operation. As described later, the combination of the Abort, MBM Purge and Write Bootloop Register commands is functionally similar to the Initialize command. (The only time the MBM Purge command is used is in conjunction with the Abort command).

Once the system has been initialized, the remainder of the command set can be selected. Assuming, for example, that a Read command is to be executed, the user selects the page number and length of the transfer via the parametric regisiters and then issues the Read command. If the system uses the polled mode, the CPU reads the status register and waits for the BUSY bit to go active and then for the FIFO READY bit to indicate that data is being sent to the FIFO. Data can be taken from the FIFO until the FIFO READY bit goes inactive.

If the page selected for the read operation is not in position to be read (i.e., the page is not at the replicate gates), additional time is required to execute the Read command as the proper page is rotated into position. In systems where faster response is desired, the Read Seek command can be used to place the page into position in order to free the CPU to perform other tasks. Once the page is in position, approximately eight milliseconds are required before the data is available to the CPU. This latency only occurs on the first page of a multipage transfer. Similarly, when a page is not in a position to be written, Write Seek can be used to position the page at the swap gates.

If there is any doubt regarding the state of the FIFO prior to a read or write operation, the user should issue a FIFO Reset command in order to clear the 7220's FIFO counter before initiating the data transfer. If a prior transfer is stopped with data remaining in the FIFO or if the FIFO is partially filled, the 7220's internal FIFO counter is not zero, and there is a danger that the subsequent transfer count may be incorrect. If the FIFO is reset properly, execution of a FIFO Reset command is redundant.

Although the 7220 FIFO may be treated as a 40-byte RAM buffer, the temptation to "pre-load" the FIFO with 40 bytes of data and then to issue a Write command should be avoided due to the danger of overflowing the FIFO. Prior to issuing a Write command, a FIFO Reset command should be sent, and the parametric registers should be loaded. Following the Write command, the CPU should monitor the status byte and wait for the BUSY and FIFO AVAILABLE bits to go active. When this status condition occurs, the user software should then send the proper number of bytes to the 7220. The FIFO AVAILABLE bit of the status byte should be polled prior to sending each byte.

An exception to not preloading the FIFO is when a Write Bootloop, Write Bootloop Register, or Write Bootloop Register Masked command is used. Prior to issuing any of these commands, a FIFO Reset command must be sent before preloading the bootloop data into the FIFO. When one of the bootloop-related commands is issued, the 7220 controller immediately begins taking data from the FIFO. If the FIFO is not preloaded, incorrect data may be transferred. The operation of the normal Write command differs from the bootloop-related commands in that, after a Write command is issued, the 7220 waits for at least two bytes to be present in the FIFO before beginning to transfer data to the bubble.

If the FSA encounters an error condition during a read or write operation, the status of the FSA is reflected in the 7220 status byte. If the user system decodes the error and decides to continue, the error flags in the 7220 controller and FSA first must be cleared. To clear the status bytes, the software can issue an Initialize command. However, this command resets all of the current operating parameters in the 7220 controller. To continue processing without resetting the system, the software can use the Software Reset command. This command resets any error flags and clears the FIFO, but does not affect the parametric register fields that define the system configuration (e.g., number of FSA channels selected).

INSTALLING THE BPK 72 BUBBLE MEMORY KIT

This section examines the individual components of the Bubble Memory System and how each component can be analyzed. All elements of the bubble system need not be working before any meaningful diagnostics can be effected. In general, a user first establishes communication between the host CPU and the 7220 controller. Next, communication with the 7242 formatter/sense-amplifier is verified via the 7220 controller. Finally, the operation of the 7110 Bubble Memory is checked. The software that exercises each of these phases of implementation should be small, well-defined device drivers that can be controlled through a system monitor.

The procedures that follow are applicable to most startup problems. The procedures are organized in chronological fashion and address each step of the installation process as it would normally occur. Software drivers in 8086 assembly language are provided to illustrate the basic functions supported by the device drivers.

Powering Up for the First Time

With power removed from the IMB-72 board, insert all of the supporting integrated circuits with the exception of the 7110 Bubble Memory Module. Insert the "dummy module" included in the BPK 72 kit in place of the 7110. The dummy module is electrically equivalent to the 7110 module and allows the circuits of the BPK 72 kit to be tested without the possibility of damaging the bubble. With both the +5V and +12V power supplies turned off, insert the IMB 72 with the dummy module into the edge connector. As power is applied to the system, monitor the RESET.OUT/pin of the 7220 controller and verify that the signal goes from low to high after power is applied. The low-to-high transition indicates that the power-up sequence has been completed successfully.

Communicating With the 7220 Bubble Memory Controller

The first step in communicating with the 7220 is to write initial values to the parametric registers using the code sequence in Table 15. When the registers have been set, the code shown in Table 12 can be used to examine the 7220 status byte.

The status value returned in Table 12 should be 40H. The user should not continue until the proper status value can be obtained repeatedly after performing the power-up sequence. Reading back the correct status indicates that the host CPU and the 7220 are communicating and that the power-up sequence is being performed by the 7220.

Table 12. Reading 7220 Controller Status

```
RDSTAT:
  ; THIS PROGRAM READS THE 7220
  ; STATUS BYTE
  ; TO READ STATUS, THE HOST CPU MUST
  ; READ FROM THE 7220 WITH A0 = 1.
    IN     AL, 49H        ; COMMANDS/STATUS
                          ; PORT ADDRESS OF
                          ; 7220
    MOV    STATUS, AL     ; MOVE AL REGISTER
                          ; TO STATUS
    RET
```

Once the power-up sequence is complete and the 7220 status register has been read, the 7220 FIFO can be accessed. The software drivers that write and read the FIFO are shown in Tables 13 and 14. Notice that these code sequences do not send commmands to the 7220; only data is transferred to and from the controller. The purpose here is to test the bus interface and timing between the CPU and the 7220 controller. In this case, the 7220 FIFO is used as a general purpose RAM. Any data can be written to the FIFO, but it is best to use an easily indentifiable sequence (e.g., an incrementing pattern) for easy recognition.

Table 13. Writing the 7220 FIFO

```
WTFIFO:
  ; THIS PROGRAM WRITES 40 BYTES FOR
  ; MEMORY TO THE 7220 FIFO.
  ; DATA IS ASSUMED TO BE ATBUFADR.
    MOVE   SI,BUFADR      ; LOAD BUFFER
                          ; POINTER
    MOV    CX,40          ; LOAD COUNT
WRT1:
    LODSB                 ; PUT BYTE AT SI
                          ; INTO AL, AUTO INCR
                          ; SI
    OUT    48H, AL        ; OUTPUT BYTE TO
                          ; DATA PORT
    LOOP   WRT1           ; DECREMENT COUNT,
                          ; LOOP IF NOT 0
    RET
```

Once forty bytes have been written to the FIFO, the 7220 status byte should be read. The status value should be "41H" (indicating that data is in the FIFO). Other status values such as "parity error" can be ignored. While status values give some indication of the CPU-7220 interaction, the integrity of the data is more important here. If the data read back is not the same as the data sent, a fundamental timing and/or interface problem between the CPU and the 7220 is indicated.

To verify that data is being transmitted to the 7220, the code sequence shown in Table 14 can be used to read back the FIFO data into user RAM space for direct comparison with the original pattern.

Table 14. Reading the 7220 FIFO

```
RDFIFO:
  ; THE PROGRAM READS 40 BYTES FROM
  ; THE 7220 FIFO INTO MEMORY.
    MOV     DI, BUFADR  ; LOAD BUFFER AD-
                        ; DRESS INTO DI
    MOV     CX,40       ; LOAD COUNT INTO
                        ; CX
RD1:
    IN      AL,48H      ; INPUT FROM DATA
                        ; PORT
    STOSB               ; STORE AL AT ADDR
                        ; IN DI, AUTO INCR. DI
    LOOP    RD1         ; DECREMENT COUNT
                        ; IN CX, LOOP IF NOT 0
    RET
```

After reading the FIFO, the status byte should be read (a value of "40H" or "42H," indicating that the FIFO has no data, should be obtained). The user should not proceed until the FIFO can be written and read correctly and until the FIFO status indicates the amount of data in the FIFO (not empty or empty). These steps verify that the CPU can communicate with the 7220. Note that no data has been transferred to or from the 7242 Formatter/Sense Amplifier or the 7110 bubble device (or dummy module).

Communicating With the 7242 Formatter/Sense Amplifier

The next step in verifying the BPK 72 is to ensure that the 7220 is driving the 7242 Formatter/Sense Amplifier properly by first setting up the 7220 for interaction with the 7242 and then sending commands to the 7220 to exercise the 7242 functions that can be verified easily.

Under normal operating conditions an Initialize command is the second command sent to the system. However, the Initialize command assumes that the 7110 Bubble Memory is installed and attempts to read bootloop information. Since the dummy module is installed at this time, timing errors result from the attempted Initialize command. Although no harm results from using the Initialize command, an Abort command followed by an MBM-Purge command can be used in place of the Initialize command to eliminate timing errors. The Abort command is sent by executing the code sequence at label "CMND9" in Table 16. When Abort command execution is complete, the user should read the status byte and check for an op-complete indication (40H).

Table 15. Write Register Sequence for Two FSA Channels

```
WTREG2:;                          WRITE REGISTERS
; 2 FSA CHANNELS SELECTED.
; THIS IS USED FOR DEBUG TO WRITE/READ THE
; BOOTLOOP REGISTERS AND CHECK FOR MISSING SEEDS, ETC.
; THE FOLLOWING VALUES INTO THE 7220 REGISTERS
;       B = 01H      : 1 PAGE TRANSFER
;       C = 10H      : SELECT 2 CHANNELS (WHOLE BUBBLE)
;       D = 08H      : STANDARD TRANSFER RATE
;       E = 00H      : PAGE 0
;       F = 00H      : FIRST BUBBLE

    MOV         AL, 0BH         ; SELECT B REGISTER
    OUT         49H, AL
    MOV         AL, 01H         ; ONE PAGE TRANSFERS
    OUT         48H, AL
    MOV         AL, 10H         ; WHOLE BUBBLE (2 FSA CHANNELS)
    OUT         48H, AL
    MOV         AL, 08H         ; LOW FREQ
    OUT         48H, AL
    MOV         AL, 00H         ; START ADDRESS = 0000H
    OUT         48H, AL
    MOV         AL, 00H         ; FIRST BUBBLE
    OUT         48H, AL
    RET
```

Once the op-complete status is received, the MBM-Purge command is issued by executing the routine labeled "CMNDE" in Table 16. This command, as described in the BPK 72 manual, clears all of the controller registers, counters and address RAM (except the block length register), the NFC bits, the FSA present counter and the high-order four bits of the address register. After the command is complete, the user again should receive an operation complete indication on reading the status byte.

After the Abort and MBM-Purge commands are executed and is status verified, additional commands may be sent to the 7220 BMC. Since the purpose of this section is to verify the interaction of the 7242 and 7220, manually loading and reading the 7242 bootloop registers can be used for the verification. Two additional commands are required to load and read the bootloop registers: the Write Bootloop Register command and the Read Bootloop Register command. These commands transfer data between the 7242 bootloop registers and the 7220 FIFO. Since the ability to transfer data between user RAM and the 7220

Table 16. 7220 Controller Commands

```
CMNDS:                  ; 7220 COMMANDS
     ; THESE 16 ROUTINES EACH SEND A SINGLE COMMAND TO THE 7220.
     ; FOR EXAMPLE, THE "INITIALIZE COMMAND" WILL WRITE 11H
     ; TO THE 7220 WITH A0 = 1. THESE ARE THE 7220 COMMANDS LISTED
     ; IN THE BPK-72 USERS MANUAL.
CMND0:
     MOV        AL, 10H         ; WRITE BOOTLOOP REGISTER MASKED COMMAND
     OUT        49H, AL
     RET
CMND1:
     MOV        AL, 11H         ; INITIALIZE COMMAND
     OUT        49H, AL
     RET
CMND2:
     MOV        AL, 12H         ; READ COMMAND
     OUT        49H, AL
     RET
CMND3:
     MOV        AL, 13H         ; WRITE COMMAND
     OUT        49H, AL
     RET
CMND4:
     MOV        AL, 14H         ; READ SEEK COMMAND.
     OUT        49H, AL
     RET
CMND5:
     MOV        AL, 15H         ; READ BOOTLOOP REGISTER COMMAND
     OUT        49H, AL
     RET
CMND6:
     MOV        AL, 16H         ; WRITE BOOTLOOP REGISTER COMMAND
     OUT        49H, AL
     RET
CMND7:
     MOV        AL, 17H         ; WRITE BOOTLOOP COMMAND
     OUT        49H, AL
     RET
CMND8:
     MOV        AL, 18H         ; READ FSA STATUS COMMAND
     OUT        49H, AL
     RET
CMND9:
     MOV        AL, 19H         ; ABORT COMMAND
     OUT        49H, AL
     RET
CMNDA:
     MOV        AL, 1AH         ; WRITE SEEK COMMAND.
     OUT        49H, AL
     RET
CMNDB:
```

Table 16. 7220 Controller Commands (cont.)

```
        MOV     AL, 1BH     ; READ BOOTLOOP COMMAND
        OUT     49H, AL
        RET
CMNDC:
        MOV     AL, 1CH     ; READ CORRECTED DATA COMMAND
        OUT     49H, AL
        RET
CMNDD:
        MOV     AL, 1DH     ; FIFO RESET COMMAND
        OUT     49H, AL
        RET
CMNDE:
        MOV     AL, 1EH     ; MBM PURGE COMMAND
        OUT     49H, AL
        RET
CMNDF:
        MOV     AL, 1FH     ; SOFTWARE RESET COMMAND
        OUT     49H, AL
        RET
```

FIFO has been verified previously, these two additional commands verify the system's ability to transfer between user RAM and the 7242 FSA.

The 7220 parametric registers must be loaded prior to sending the Write Bootloop Register command. The sequence of operations is important; loading the parametric registers destroys the first byte of data in the 7220 FIFO. If valid bootloop information is placed in the FIFO before the parametric registers are loaded, the first byte of bootloop register information is invalid. Accordingly, the sequence of operations must be as follows:

(1) load the 7220 parametric registers
(2) load bootloop data into the 7220 FIFO
(3) send the Write Bootloop Register command.

As a point of interest, if a user wishes to maintain the system bootloop in EPROM rather than to allow automatic handling by the system, the Initialize command would not be used and would be replaced by a sequence similar to the one described.

After the 7220 parametric registers are loaded, the CPU next must load the 7220 FIFO with 40 bytes of bootloop register data using the "write FIFO" sequence from Table 13. This sequence then is followed by the code sequence to issue the Write Bootloop Register command. The data pattern written to the bootloop register should be an easily identified sequence of bytes such as an incrementing pattern. Under operational conditions, the data written to the bootloop registers represents "loop map" information that is written on the label of the 7110 device. Under these test conditions, it only is necessary to ensure that the 40 bytes sent out are the same 40 bytes read back.

Once the Write Bootloop Register command has been sent, the status byte is read (when the BUSY bit goes low) and an operation-complete status is verified. Any parity error indication may be ignored. Valid status at this point indicates that communication with the 7242 has been established. To verify that the data has been transferred properly, the contents of the bootloop register are read into the 7220's FIFO. The CPU then must transfer the data to user RAM in order to compare the data with the original pattern. To read the bootloop register, it only is necessary to issue the Read Bootloop Register command. This command places the contents of the 7242's bootloop register into the 7220's FIFO. The user then must execute the "read FIFO" sequence from Table 14 in order to transfer the data from the 7220 FIFO to RAM. Comparing the loop map written into the bootloop register and the loop map read from the bootloop register should show the loop maps to be equal.

Installing the 7110 MBM

Reading and writing the 7110 bubble memory requires the application of specific control signals at the appropriate times within the read or write cycles. These control signals originate from the 7254 and 7230 integrated circuits and are generated under the control of the 7220 BMC. Prior to installing the 7110, the presence of the control signals should be verified. While it is unlikely that the 7110 can be seriously damaged, it is possible for the "seeds" and bootloop established at the factory to be lost if there are problems with the 7254 or 7330 control signals and, if lost, would require additional steps on the part of the user to regenerate the seeds and bootloop data. With the dummy module installed, the required control signals can be verified directly on the bubble socket, and the possibility of damaging the bubble can be avoided.

The first control signal waveform to check is the coil drive on pins 9, 10, 11, and 12 of the 7110 socket. The drive current can be verified by ensuring that the voltage waveform on these pins (or on pins 1 and 7 of the 7254) conforms to Figure 6A when the drive field is being rotated. To rotate the drive field, the following code sequence can be used:

1. Write the parametric registers.
2. Send the Read command.

Next, the "cut and transfer" pulses generated during a read operation should be checked. The waveforms on pins 2 and 3 of the 7110 socket (REPLICATE.A and REPLICATE.B), should appear as shown in Figure 6B.

The cut and transfer pulses that occur during a write operation should now be verified. The waveforms on pins 7 and 8 of the 7110 socket (GENERATE. A and GENERATE. B) should appear as shown in Figure 6C. Since a write operation is required, a new code sequence must be used for this test:

1. Write the parametric registers.
2. Write data (any patten) to the FIFO.
3. Send the Write command.

bootloop register of the 7242 first must be loaded to allow data to be written. A Write Bootloop Register Masked command can be used to write a bootloop register pattern of all ones; it is only necessary to write the bootloop register once.

Finally, the SWAP pin is tested for proper operation during a write operation. The waveforms on pins 13 and 14 of the 7110 (SWAP.A and SWAP.B) should appear as shown in Figure 6D. The code sequence described for a write operation may be used.

One additional check of the system should be made prior to installing the 7110 device to determine if valid status values are received after a Read or Write command is issued to the 7220 BMC. Since the bubble is not yet installed, no data actually is transferred; the system should, however, execute the Read or Write command, and valid status should be received. Since a new command cannot be issued to the 7220 while a command is in progress, an Abort command is sent to cancel any command that may be pending from the last test performed. Next, a FIFO Reset command is sent to clear any data remaining in the FIFO. The status byte received should indicate an OP-COMPLETE and FIFO AVAILABLE status condition. The 7220 now is ready to execute a Read or Write command.

First, the 7220 parametric registers are loaded using the modified "diagnostic" driver shown in Table 17. This routine selects one FSA channel (half of a bubble) and, with ECC disabled, requires the loading of only 34 bytes in the 7220 FIFO. By limiting the FIFO to less than 40 bytes, FIFO underflow/overflow conditions are eliminated, and timing errors are avoided in the status byte. After, the 7220 FIFO is preloaded with 34 bytes of data (any pattern), a Write command is issued to the 7220 BMC. The 7220 status value received following command execution should reflect OP-COMPLETE since the 7220 transferred the data from its FIFO to the 7242 and executed the Write command as though the bubble were in place.

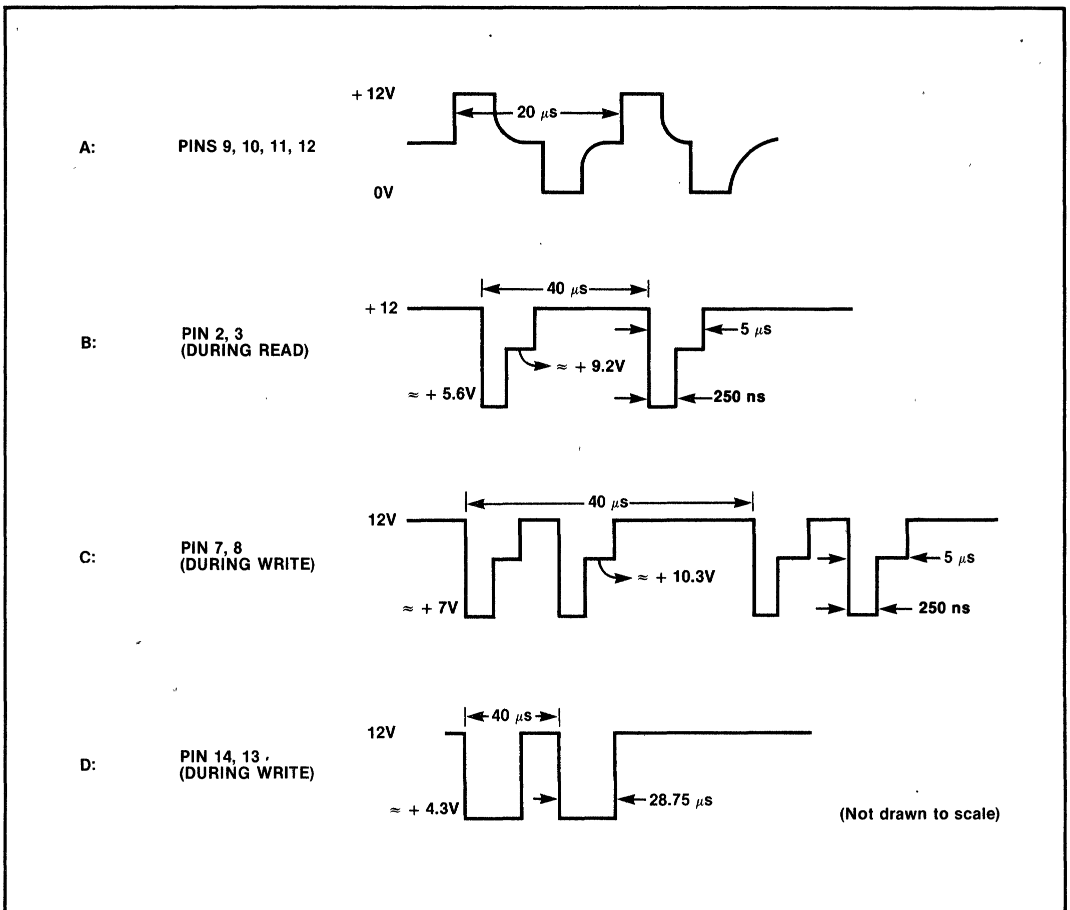


Figure 6. Control Signal Waveforms

To test the system in the read mode, the 7220 parametric registers are reloaded and a Read command is issued to the 7220. The user software must now read 34 bytes of "data" from the 7220's FIFO. Note that the data read will consist of all zeroes since no bubble is in place.

When the system completes all of the previous tests successfully, the 7110 bubble memory device may be inserted. Before proceeding, REMOVE POWER FROM THE SYSTEM.

Installing the 7110 is no different from installing any other device. Remove the dummy module in the 7110 socket and insert the 7110 Bubble Memory. Note that the 7110 is keyed to prevent the device from being inserted incorrectly. When power is applied, the system should execute its power-up sequence as described for the dummy module, and the 7220 status byte should return OP-COMPLETE after the parametric registers have been loaded.

Table 17. Write Register Sequence for One FSA Channel

```
WTREG1:;            WRITE REGISTERS (ONE HALF BUBBLE)

    ; THIS PROGRAM WRITES THE 7220 REGISTERS "B" THROUGH "F".
    ; DIAGNOSTIC ROUTINE WITH ONE FSA CHANNEL SELECTED
    ; THE FOLLOWING VALUES ARE WRITTEN TO THE 7220 REGISTERS.
    ;                B = 01H  :        1 PAGE TRANSFER
    ;                C = 00H  :        SELECT 1 CHANNEL (HALF BUBBLE)
    ;                D = 08H  :        LOW FREQ
    ;                E = 00H  :        PAGE 0
    ;                F = 00H  :        FIRST BUBBLE
      MOV      AL, 0BH       ; SET REGISTER ADDRESS COUNTER (RAC) TO B REGISTER
      OUT      49H, AL       ; PROT ADDRESS OF 7220 WITH A0 = 1
      MOV      AL, 01H       ; SET B REGISTER TO 01H (ONE PAGE TRANSFER)
      OUT      48H, AL       ; PORT ADDRESS OF 7220 WITH A0 = 0
      MOV      AL, 0H        ; SELECT HALF BUBBLE (1 FSA CHANNEL)
      OUT      48H, AL
      MOV      AL, 08H       ; SELECT LOW FREQ (NO ERROR CORRECTION)
      OUT      48H, AL
      MOV      AL, 0H        ; START ADDRESS = 000H
      OUT      48H, AL
      MOV      AL, 0H        ; SELECT THE FIRST BUBBLE
      OUT      48H, AL
      RET
```

Normal Read and Write Operations

Under normal operating conditions, a user sends an Initialize command and then proceeds to access the bubble. The Initialize command automatically purges the RAM area of the 7220, reads and decodes the bootloop on the 7110, fills the 7242 bootloop registers, and places the 7110 at page 0. This very important command is the next command to be tested before reading and writing data.

To verify the Initialize command, load the 7220 parametric registers to select both FSA channels for one bubble and then send the Initialize command. Status following execution of this command should be 40H, OP-COMPLETE. Once the 7220 is initialized, data can be transferred to and from the bubble. For a first attempt, it is recommeded that the operations be kept simple. That is, avoid error correction, DMA, or interrupts and only attempt single page transactions until reasonably familiar with the basic operations.

Prior to issuing the Write command, a FIFO Reset command is sent and then the parametric registers are loaded to select the page address and number of FSA channels. After the Write command is sent, the data should be output to the 7220 FIFO. When the proper number of bytes have been transferred, the 7220 status byte should reflect OP-COMPLETE and FIFO AVAILABLE to indicate that the data has been written into the 7110 bubble memory and can now be read. To read back the data written, issue a FIFO Reset command and reload the parametric registers to select the same page address in which the data was written. Issue the Read command to move the data from the 7110 to the 7220 FIFO and then use the "read FIFO" routine to transfer the data to user RAM. As always, the 7220 status byte should be checked after the operation.

AN IMPLEMENTATION EXAMPLE

To illustrate the ease with which Intel's bubble memory solution may be implemented, an MCS®-86 System Design Kit (SDK-86) is used as a vehicle to control a single BPK 72 bubble memory kit.

The bus interface between the 8086 CPU and the 7220 bubble memory controller requires seven integrated circuits and consists of four sections: address decode, data bus decode and buffering, a clock circuit, and miscellaneous control logic. The system requires power supply voltages of + 12V, + 5V, and, if a CRT is used, − 12V.

The 8086 bus is expanded through two 50-pin, wirewrap connectors, and the BPK 72 is connected to the SDK-86 by a flat cable into a 40-pin connector located on the SDK-86. The following interface diagram shows how the signals required by the bubble system are derived from the 8086. Detailed diagrams of the address, data, clock and control logic are in the appendix.

Either the SDK-86's Keypad or Serial monitor may be used to write and debug the necessary software drivers to control the BPK 72. There is, however, an EPROM-based monitor (BMDSDK) explicity designed for the BPK 72 and is available from the Intel Insite Library. Some of the bubble-specific portions of this monitor are discussed in the following text.

Monitor Software

The BMDSDK Bubble Monitor is a highly-modular program that is written in 8086 assembly language and that resides in two 2716 EPROMs. This monitor implements, at the console level, most of the standard SDK-86 monitor functions (display/change memory, etc.) and all of the 7220 commands. The current version of the monitor utilizes only polled I/O protocol; implementing an interrupt-driven system on the SDK-86 is possible using the principles outlined in this application note. The DMA mode of operation is not available with the hardware described.

The BPK 72 driver routines are confined to one module; a listing of this module is included in the appendix. To provide some feeling for the elements of "operational" software as opposed to the test drivers discussed earlier, the write function implemented in BMDSDK monitor is examined. The flow chart in Figure 9 shows how the routine is constructed on a functional basis. Note that the subroutine reflects a very "safe" approach in that the FIFO Reset command always is sent prior to issuing the Write command. While the FIFO Reset command is not mandatory, if there is any a doubt regarding the state of the FIFO prior to a read or write operation, resetting the FIFO is a good idea. Note also that a running byte count is maintained and that the routine exits when the count goes to zero. Such a counter is not actually necessary; the FIFO AVAILABLE bit alone can be used to gate the data to the 7220.

The calling program supplies the BMWRIT routine with the total number of bytes to be transferred in the CX register. The total number of bytes written is sent to the console at the end of the operation as a monitor function. BMWRIT also returns the value of the status byte to the calling program.

Note that at label WRIT01, the routine does not progress after the Write command is sent unless both the BUSY and FIFO AVAILABLE bits are set by the controller. Once these values are set, the code issues a byte of data to the controller only if the FIFO AVAILABLE bit indicates there is room. The remainder of the code in BMWRIT is concerned with processing special write requests for the bootloop and bootloop register commands.

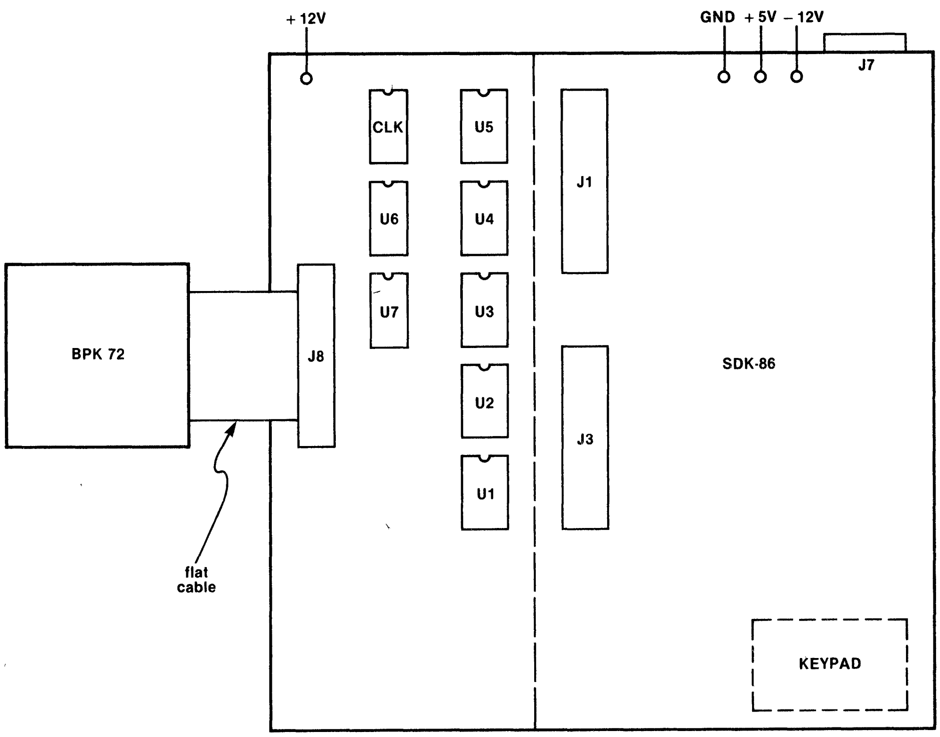


Figure 7. SDK-86/BPK 72 Implementation

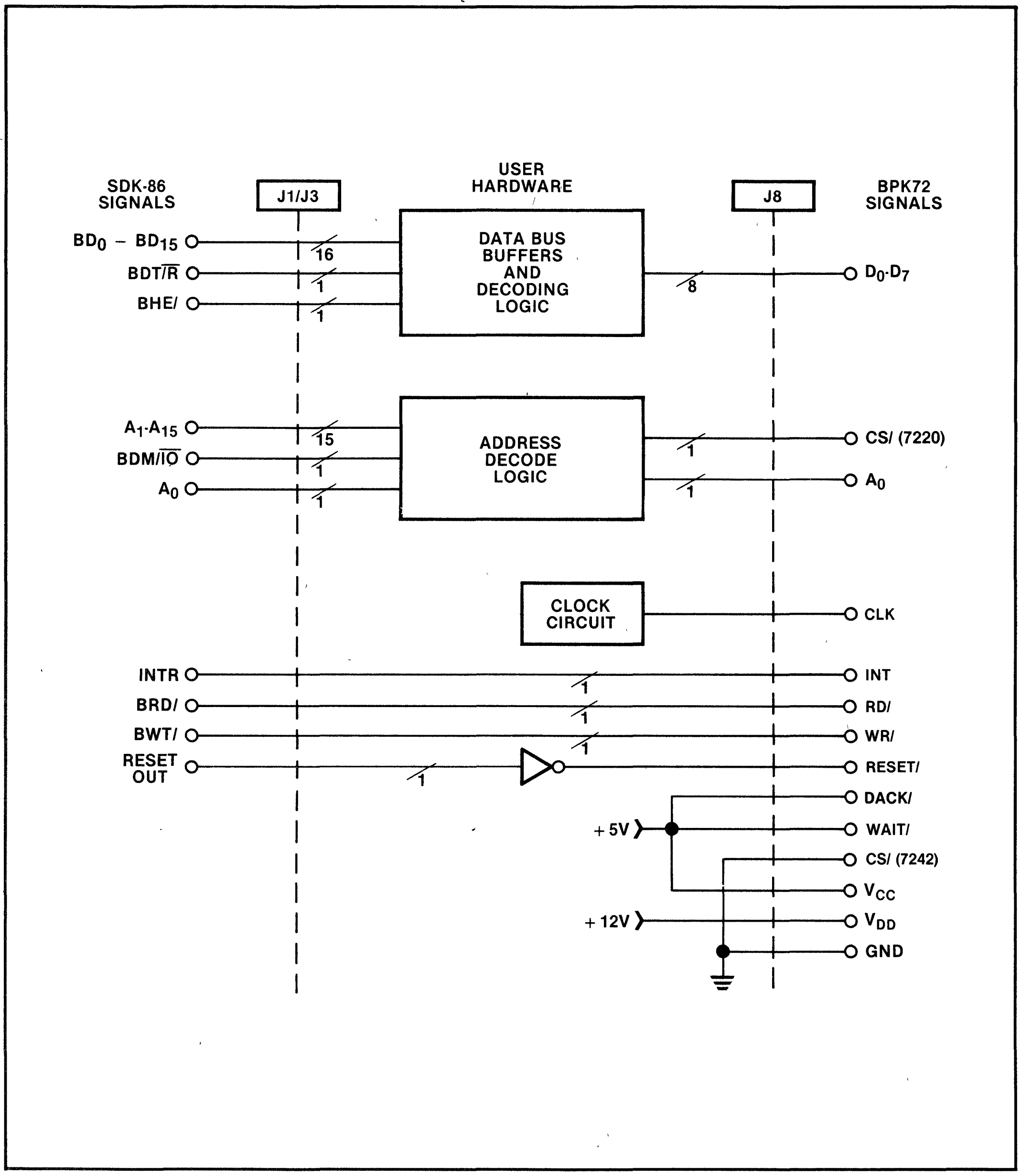


Figure 8. SDK-86/BPK 72 Interface Diagram

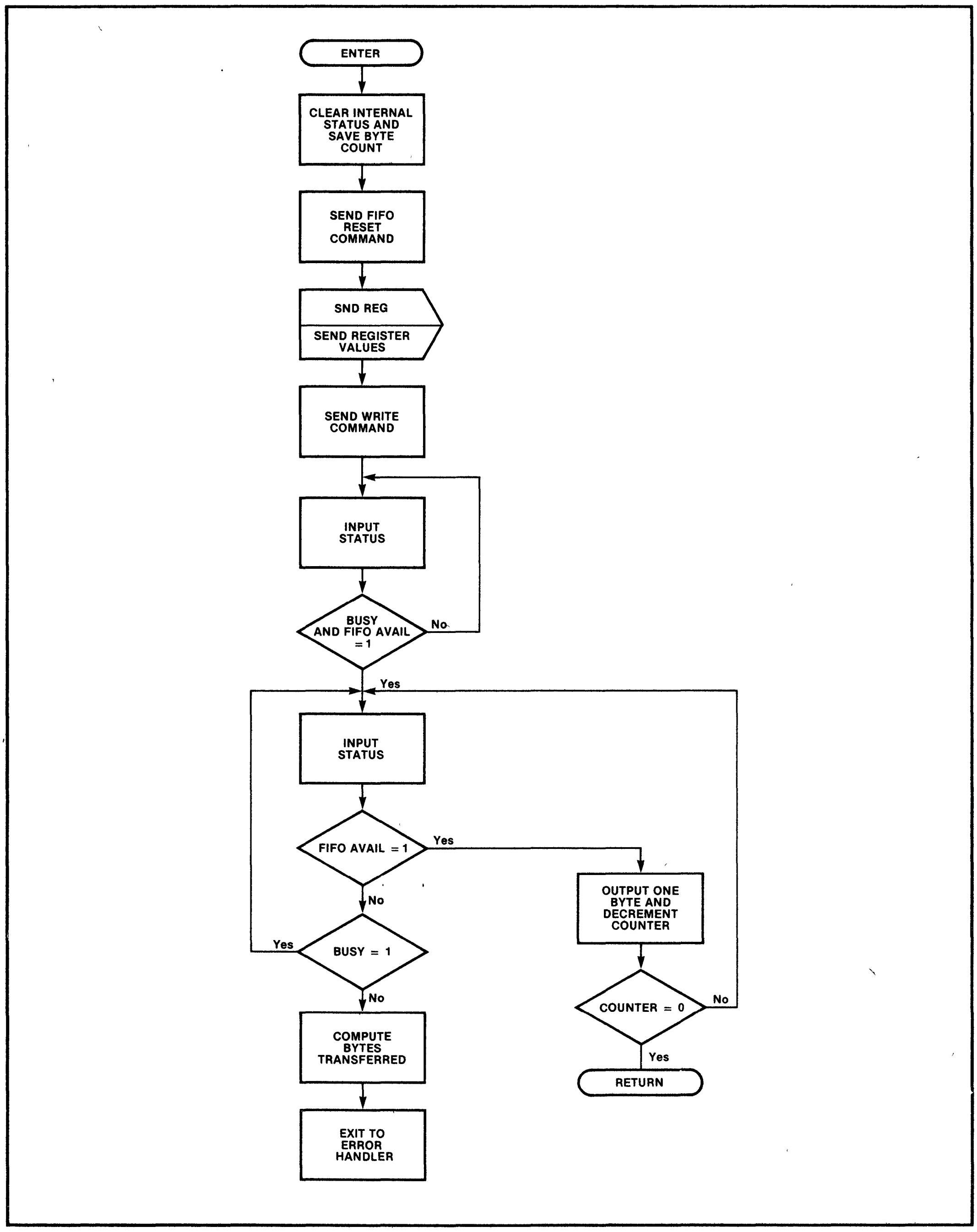


Figure 9. BMWRIT Flowchart

Table 18. BMWRIT Procedure for the SDK-86

```
;
; FUNCTION: BMWRIT - WRITE BUBBLE MEMORY DATA.
; INPUTS: CX = # OF BYTES TO WRITE.
; OUTPUTS: A = STATUS: F/F(C=1: ERROR OCCURED) BX=# OF BYTES WRITTEN.
; CALLS: SNDREG, BMWAIT.
; DESTROYS: ALL.
; DESCRIPTION:   THIS PROCEDURE PERFORMS A BUBBLE MEMORY WRITE OPERATION.
;                AN ERROR WILL OCCUR IF THE NUMBER OF BYTES GIVEN FOR THE
;                WRITE OPERATION EXCEED THE NUMBER THAT THE BMC EXPECTS
;                (DERIVED FROM COMMAND, BLOCK LENGTH AND NUMBER OF FSA
;                CHANNELS), OR IF THE NUMBER OF BYTES IS LESS THAN THAT
;                WHICH THE BMC EXPECTS.
;
BMWRIT:
        XOR     AL, AL          ; A = 0
        MOV     STATUS, AL      ; CLEAR STATUS
        MOV     BX, CX
        MOV     AL, CFR
        OUT     BMSTAT, AL      ; FIFO RESET
        CALL    SNDREG          ; SEND REGISTERS TO BMC.
        MOV     SI, BUFADR      ; SET UP SRC BFR PTR (IN DATA SEG)
        MOV     AL, BMCMD       ; GET COMMAND
        OUT     BMSTAT, AL      ; ISSUE IT.
WRIT01:
        IN      AL, BMSTAT
        TEST    AL, BUSYBT      ; WAIT FOR BUSY. . .
        JZ      WRIT01
        TEST    AL, FIFOBT      ; AND FIFO READY
        JZ      WRIT01
;
; KEEP STUFFING DATA INTO FIFO UNTIL DONE OR AN ERROR OCCURS.
; (NOTE: BMC GOING NOT BUSY IS AN ERROR).
;
WRIT03:
        IN      AL, BMSTAT      ; GET STATUS
        TEST    AL, FIFOBT      ; FIFO READY?
        JZ      WRIT04          ; NO, WAIT FOR IT
        LODSB                   ; YES, GET DATA FOR IT
        OUT     BMDATA, AL      ; GIVE IT TO BMC
        LOOP    WRIT03          ; LOOP UNTIL DONE.
        JMP     BMWAIT          ; XFER DONE, WAIT FOR A GOOD STATUS
WRIT04:
        TEST    AL, BUSYBT
        JNZ     WRIT03          ; OK IF STILL BUSY
        SUB     BX, CX          ; BX:# OF BYTES XFERED
        JMP     CTRL99          ; ERROR IF NOT BUSY AND CX NOT ZERO
; SPECIAL WRITE FOR BOOTLOOP AND BOOTLOOP REG CMNDS
;
BMWRTB:
        XOR     AL, AL          ; A = 0
        MOV     STATUS, AL      ; CLEAR STATUS
        MOV     BX, CX
        MOV     AL, CFR
        OUT     BMSTAT, AL      ; FIFO RESET
        CALL    SNDREG          ; SEND REGISTERS TO BMC.
        MOV     SI, BUFADR      ; SET UP SRC BFR PTR (IN DATA SEG)

  ; FILL FIFO WITH 20/40/41 BYTES
```

Table 19. BMWRIT Procedure for the SDK-86 (cont.)

```
;
;
;
WRTB01:
        LODSB
        OUT     BMDATA, AL      ; STICK IN FIFO.
        LOOP    WRTB01          ; LOOP UNTIL FILL COUNT=0.
        IN      AL, BMSTAT      ; GET BMC STATUS
        TEST    AL, BUSYBT      ; CHECK BUSY BIT.
        JZ      SHORT WAITEX    ; NOT BUSY, ALREADY DONE.
        MOV     CX, OFFFFH      ; JUST IN CASE. . .
WAITPO:                         ; POLLED WAIT MODE
        IN      AL, BMSTAT      ; GET STATUS
        TEST    AL, BUSYBT      ; CHECK BUSY BIT
        LOOPNZ  WAITPO          ; LOOP IF STILL BUSY
        JCXZ    CTRL99          ; PROBABLY AN ERROR IF CX=0
WAITE:
        MOV     STATUS, AL      ; A = STATUS
        RET
```

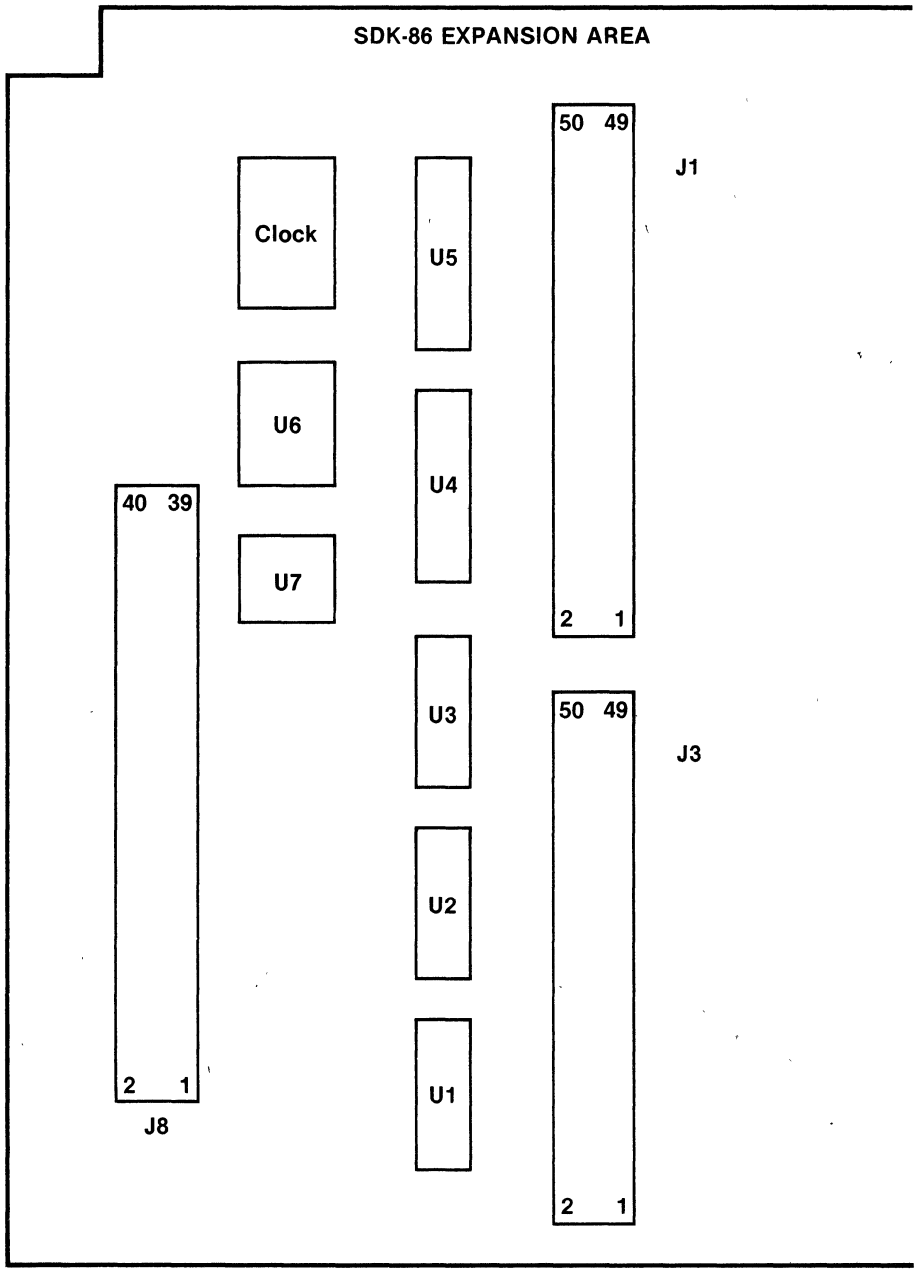
SUMMARY

The purpose of this application note is to provide a more clear understanding of the functions and characteristics of the BPK 72 one-megabit bubble memory kit. This kit has been designed specifically to relieve the user of the design effort that historically is associated with implementing a bubble memory system, and to provide a simple interface that is compatible with a broad range of microprocessor systems.

The BPK 72 is a subsystem in itself that should be viewed as simply one more component on the system bus. This component-level approach, plus the inherent flexibility of the kit, provides the user with maximum utility and functionality. By understanding how each of the subsystem parts fits together and by approaching the implementation of the kit in a methodical fashion as described in this note, the development of a working system is facilitated.

APPENDIX A

SDK-86/BPK 72
HARDWARE INTERFACE



Top View User Hardware Section

Figure 10. Parts Layout

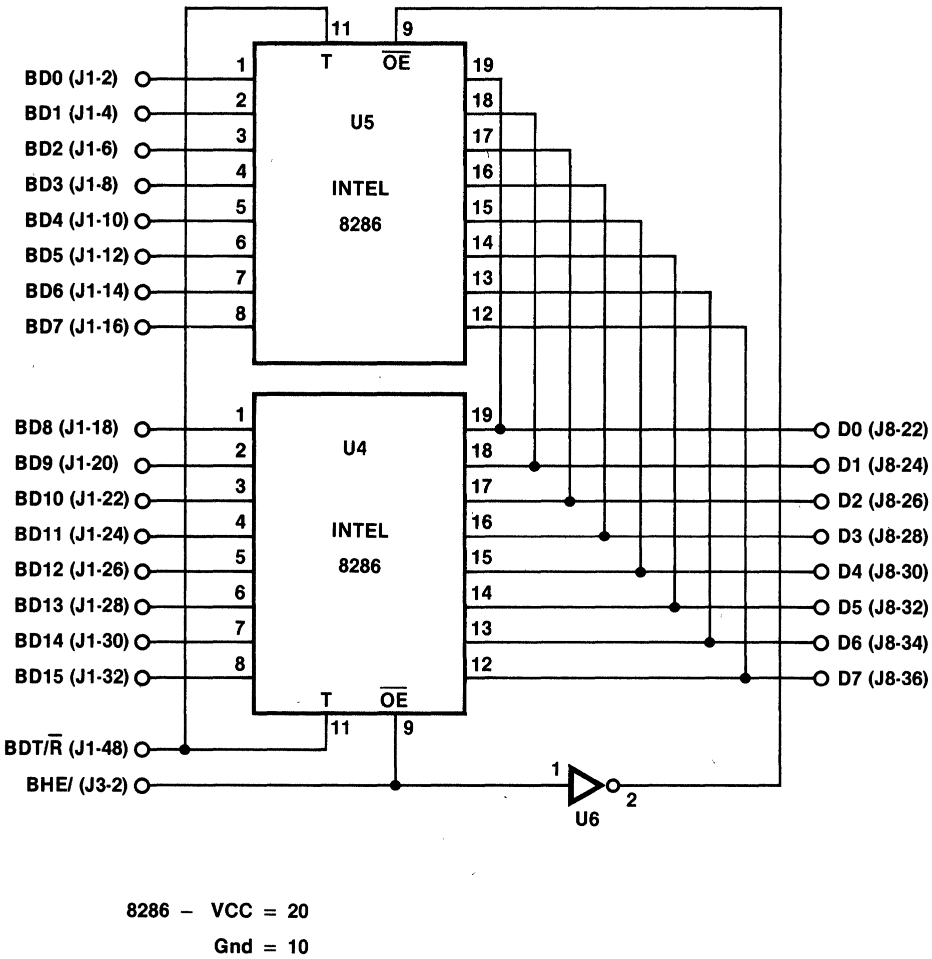


Figure 11. Data Bus Buffer and Decoding Logic

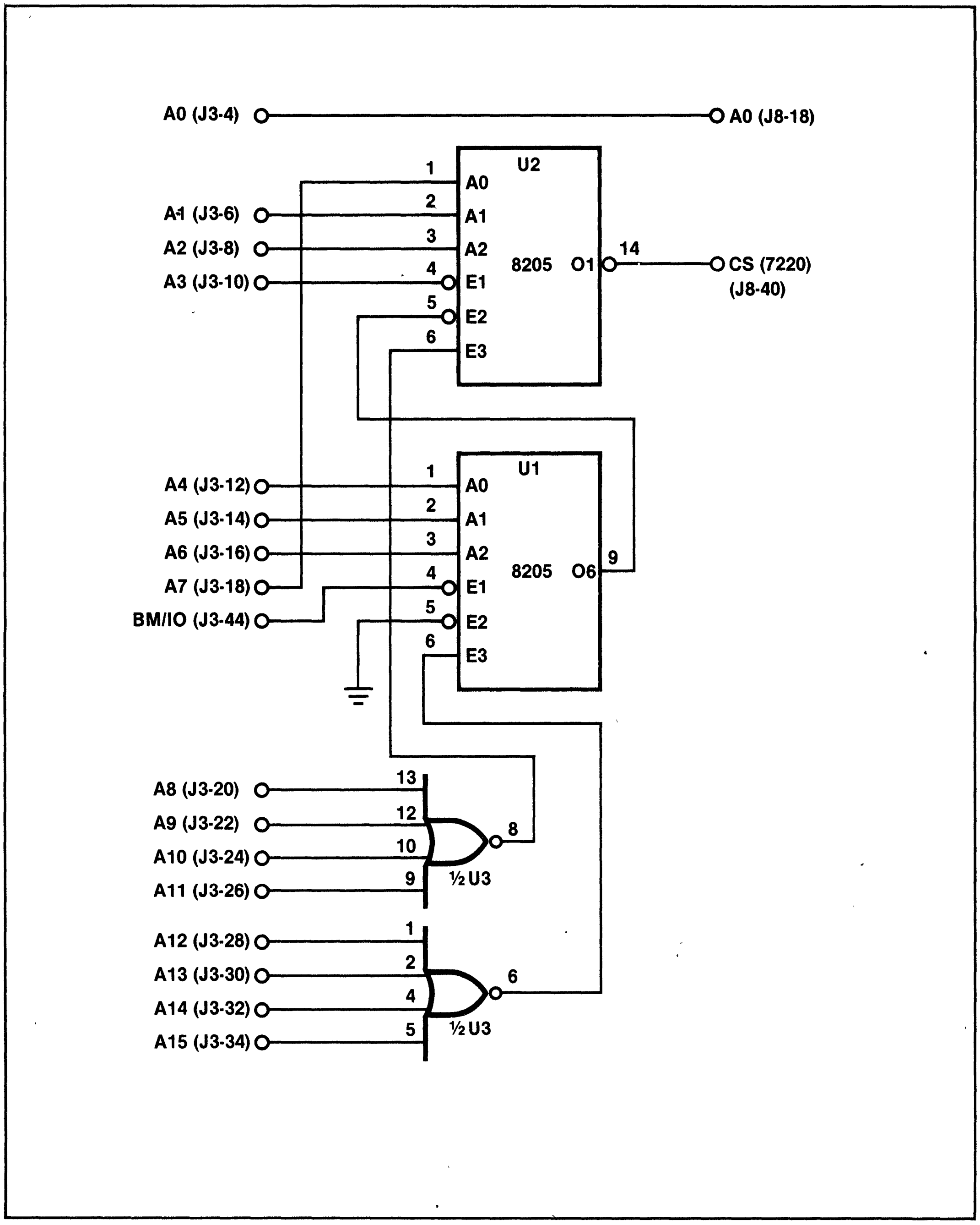


Figure 12. Address Decode Logic

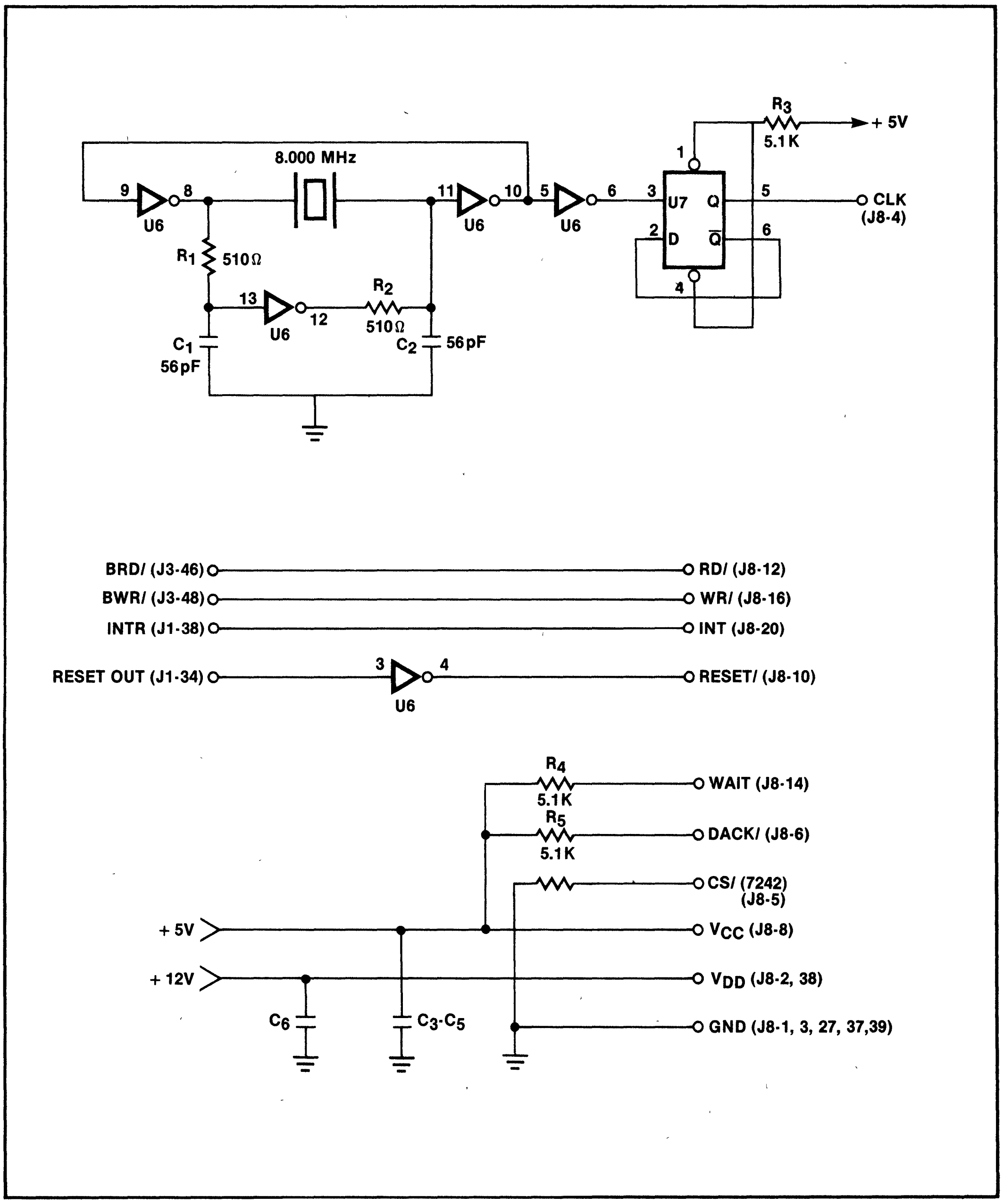


Figure 13. Clock Circuit and Control Signals

Table 20. SDK-86 Pinout

Pin	J1/J2	J3/J4	J5	J6
2	BD0	BHE/	P2C1	—
4	BD1	A0	P2C2	P1B3
6	BD2	A1	P2C3	P1B4
8	BD3	A2	P2B7	P1B2
10	BD4	A3	P2B0	P1B5
12	BD5	A4	P2B6	P1B1
14	BD6	A5	P2B3	P1B6
16	BD7	A6	P2B4	P1B0
18	BD8	A7	P2B2	P1B7
20	BD9	A8	P2B5	P1C3
22	BD10	A9	P2B1	P1C2
24	BD11	A10	P2C0	P1C1
26	BD12	A11	P2C4	P1C0
28	BD13	A12	P2C5	P1C4
30	BD14	A13	P2C6	P1C5
32	BD15	A14	P2C7	P1C6
34	RESET OUT	A15	P2A0	P1C7
36	PCLK/	A16	P2A7	P1A0
38	INTR	A17	P2A1	P1A7
40	TEST	A18	P2A6	P1A1
42	HOLD	A19	P2A2	P1A6
44	BHLDA	BM/IO/	P2A5	P1A2
46	BDEN/	BRD/	P2A3	P1A5
48	BDT/R/	BWR/	P2A4	P1A3
50	BALE	BINTA/	—	P1A4

All Odd Pins are Ground except as follows:

	J2
41	CSX/ (FD000-FDFFF)
43	CSY/ (FC000-FCFFF)
45	BS3
47	BS4
49	BS5

Table 21. SDK-86/BPK 72 Cable Wiring

Signal	J8	P1
+12v	2, 38	B, X
+5v	8	F
Ground	1, 3, 27, 37, 39	1, A, P, 22, Z
D0	22	11
D1	24	12
D2	26	13
D3	28	14
D4	30	15
D5	32	16
D6	34	17
D7	36	18
CS/ (7220)	40	Y
A0	18	10
RD/	12	J
WR/	16	K
INT	20	N
RESET/	10	H
CS/ (7242)	5	E
WAIT/	14	8
CLK	4	4
DACK/	6	L

Cable is standard 40 conductor Flat Cable.
All Odd Conductors are grounded at J8.

Table 22. SDK-86/BPK 72 Parts List

Item	Description	QT	Ref	
1	IC-8205 - Bindry Decoder	2	U1, U2	Intel (TI-74LS13)
2	IC-8286 - Octal Bus Tranciever	2	U4, U5	Intel
3	IC-746525 - Dual 4 Input M	1	U3	Any
4	IC-74H04 - Inverter	1	U6	Any
5	Resistor 510Q 1/4w	2	R1, R2	Any
6	Capacitor, 56pF 25V	2	C1,C2	Any
7	Capacitor, .1pF 25V	4	C3-C6	Any
8	Crystal, 8.000MHz Serie Res.	1	Y1	Any
9	Connector, 50 pin wirewrap	2	J1, J3	3M # 3433
10	Connector, 40 pin wirewrap	1	J8 (M)	3M # 3432
11	Connector, 40 pin	1	J8 (F)	3M # 3417
12	Connector, 44 pin Edge w/w	1	P1	Any
13	IC Socket, 20 pin w/w	2		Any (Augat)
14	IC Socket, 16 pin w/w	3		Any
15	IC Socket, 14 pin w/w	3		Any
16	Adapter Plug Assembly, 16 pin	1		Augat # 616-CE1
17	Flat Cable, 40 Conductor, 1 Ft.	1		3M # 3365
18	IC-74LS74 - Dual D Flip-Flop	1	07	Any
19	Resistor 5.1K 1/4W ±5%	3	R3, R4, R5 R5	Any
20	IC-74LS32 - OR Gate	1	U8	Any

APPENDIX B

SDK-86/BPK 72
SOFTWARE DRIVER

```
ISIS-II MCS-86 MACRO ASSEMBLER V2.1 ASSEMBLY OF MODULE DRIVER
OBJECT MODULE PLACED IN :F1:DRIVER.OBJ
ASSEMBLER INVOKED BY:  asm86 :f1:DRIVER.a86 xref print(:f1:DRIVER.lst) debug WORKFILES(:F0:,:F0:)

LOC  OBJ                 LINE     SOURCE

                                  $TITLE(                       BPK-72 DRIVER ROUTINES.)
                                          NAME    DRIVER
                              +1  $INCLUDE(:F1:RAMDEF.EXT)
                        =1        ;
                        =1        ;       publics from module RAMDEF, file RAMDEF.A86
                        =1        ;
----                    =1        STACK   SEGMENT STACK
                        =1                EXTRN   BMSTAK:NEAR
----                    =1        STACK   ENDS
                        =1        ;
----                    =1        DATA    SEGMENT PUBLIC
                        =1                EXTRN   RAM:BYTE,SCRBUF:BYTE,MYBUF:BYTE
                        =1                EXTRN   DEFADR:WORD,DEFBUB:BYTE,DEFNFC:BYTE,DEFENA:BYTE
                        =1                EXTRN   DEFMOD:BYTE,DEFPAG:WORD,DEFBLK:WORD
                        =1                EXTRN   BUFADR:WORD,BLKLEN:WORD,ENABLE:BYTE,PAGENO:WORD
                        =1                EXTRN   BBLNUM:BYTE,NFC:BYTE,MODE:BYTE,STATUS:BYTE,BMCMD:BYTE
                        =1                EXTRN   INBUF:BYTE,INBUFP:WORD,INBUFC:BYTE
                        =1                EXTRN   INBUFA:WORD,INBUFL:BYTE
                        =1                EXTRN   OUTBUF:BYTE,OUTBFP:WORD,OUTBFC:BYTE
                        =1                EXTRN   OUTBFA:WORD,OUTBFL:BYTE
                        =1                EXTRN   RDLEN:WORD,WRLEN:WORD
                        =1                EXTRN   PROMPT:BYTE,LEVMSK:BYTE
                        =1                EXTRN   BPADR:WORD,USERRG:WORD
                        =1                EXTRN   POPREGS:WORD,PUSHREGS:WORD
                        =1                EXTRN   USERBX:WORD,USERDS:WORD,USERBP:WORD,USERSS:WORD
                        =1                EXTRN   USERSP:WORD,USERIP:WORD,USERCS:WORD,USERFL:WORD
                        =1                EXTRN   USERPC:WORD
----                    =1        DATA    ENDS
                        =1        ;
                              +1  $INCLUDE(:F1:BMC.EQU)
                        =1        ;
                        =1        ; THESE ARE THE COMMAND EQUATES FOR BMDS
                        =1        ;
  0010                  =1        CWBRM   EQU     10H                     ; WRITE BOOTLOOP WITH MASK.
  0011                  =1        CIZ     EQU     11H                     ;INITIALIZE
  0012                  =1        CRD     EQU     12H                     ;READ
  0013                  =1        CWD_    EQU     13H                     ;WRITE
  0014                  =1        CRS     EQU     14H                     ;READ SEEK
  0015                  =1        CRBR    EQU     15H                     ;READ BOOTLOOP REGISTER
  0016                  =1        CWBR    EQU     16H                     ;WRITE BOOTLOOP REGISTER
  0017                  =1        CWB     EQU     17H                     ;WRITE BOOTLOOP
  0018                  =1        CRFS    EQU     18H                     ;READ FIFO STATUS
  0019                  =1        CAB     EQU     19H                     ;ABORT
  001A                  =1        CWRS    EQU     1AH                     ; WRITE SEEK.
  001B                  =1        CRB     EQU     1BH                     ;READ BOOTLOOP
  001C                  =1        CRCD    EQU     1CH                     ;READ CORRECTED DATA
  001D                  =1        CFR     EQU     1DH                     ;FIFO RESET
  001E                  =1        CPURG   EQU     1EH                     ; MBM PURGE COMMAND.
  001F                  =1        CSR     EQU     1FH                     ;SOFTWARE RESET
                        =1        ;
```

```
LOC  OBJ            LINE      SOURCE

                =1            ; I/O PORT ADDRESSES.
                =1            ;
 00E1           =1            BMSTAT  EQU     0E1H          ; BUBBLE MEMORY DEVICE STATUS PORT.
 00E0           =1            BMDATA  EQU     0E0H          ; BUBBLE MEMORY DEVICE DATA PORT.
                =1            ;
                =1            ; STATUS WORD BITS
                =1            ;
 0001           =1            FIFOBT  EQU     01H           ; FIRST BIT IS FIFO STATUS
 0002           =1            PARERR  EQU     02H           ; SECOND BIT IS PARITY ERROR.
 0004           =1            UNCERR  EQU     04H           ; THIRD BIT IS UNCORRECTABLE ERROR BIT.
 0008           =1            CORERR  EQU     08H           ; FOURTH BIT IS CORRECTABLE ERROR BIT.
 0010           =1            TIMERR  EQU     10H           ; FIFTH BIT IS TIMING ERROR BIT.
 0020           =1            OPFAIL  EQU     20H           ; OPERATION FAIL BIT.
 0040           =1            OPDONE  EQU     40H           ; OPERATION COMPLETE BIT.
 0080           =1            BUSYBT  EQU     80H           ; BUSY BIT.
                =1            ;
                =1            ; ENABLE REG BITS
                =1            ;
 0001           =1            INTENA  EQU     01H           ; INTERRUPT NORMAL
 0002           =1            IERENA  EQU     02H           ; INTERRUPT ERROR
 0004           =1            DMAENA  EQU     04H           ; DMA
 0008           =1            RSVD1   EQU     08H
 0010           =1            WBLENA  EQU     10H           ; WRITE BOOTLOOP
 0020           =1            RCDENA  EQU     20H           ; READ CORRECTED DATA
 0040           =1            ICDENA  EQU     40H           ; INTERNALLY CORRECTED DATA
 0080           =1            RSVD2   EQU     80H
                         +1   $EJECT
```

```
LOC  OBJ            LINE    SOURCE

                            CODE    SEGMENT PUBLIC
                                    ASSUME  DS:DATA,CS:CODE,SS:STACK
                            ;*************************************************************
                            ;                                                            *
                            ;              BPK72 DRIVER routines                         *
                            ;              ===== ====== ========                         *
                            ;                                                            *
                            ;  The routines in this module constitute the routines       *
                            ;  needed to directly drive the BPK72 bubble memory          *
                            ;  development board.  This module is designed to be self    *
                            ;  contained, and may be called by ANY user procedures.      *
                            ;                                                            *
                            ;       The procedures in this module are                    *
                            ;                                                            *
                            ;  BMCTRL - Perform non-data transfer BMC operations.        *
                            ;  BMREAD - Perform data read BMC operations.                *
                            ;  BMWRIT - Perform data write BMC operations.               *
                            ;                                                            *
                            ;  ZAPREG - Set internal registers to an acceptable value    *
                            ;                                                            *
                            ;       Parameter passing                                    *
                            ;       ========= =======                                    *
                            ;                                                            *
                            ;  All parameters are passed to the BMC driver routines via  *
                            ;  common (PUBLIC) variables.  These variables are           *
                            ;                                                            *
                            ;  BUFADR - The memory address of the input/output buffer    *
                            ;           to be used for data transfer operations.         *
                            ;  ENABLE - The enable byte to be passed to the BMC before   *
                            ;           every operation.                                 *
                            ;  PAGENO - The starting block number to be passed to the    *
                            ;           BMC before every operation.  (NOTE: This field   *
                            ;           has no meaning for control operations).          *
                            ;  BLKLEN - The number of pages to be transfered by the BMC.*
                            ;           (NOTE: This field has no meaning for control     *
                            ;           operations).                                     *
                            ;  BBLNUM - The bubble select to be transfered to the BMC    *
                            ;           before every operation.(NOTE: This field has     *
                            ;           no meaning for SOME control operations).         *
                            ;  NFC    - The number of FSA channels passed to the BMC     *
                            ;           before every operation. (NOTE: This field has    *
                            ;           no meaning for SOME of the control operatioNs).  *
                            ;                                                            *
                            ;  For a detailed definition of the ENABLE,PAGENO.BLKLEN,    *
                            ;   BBLNUM. and NFC fields, refer to the BPK-72 USER MANUAL  *
                            ;   or the Bubble Memory Design Handbook.                    *
                            ;                                                            *
                            ;*************************************************************
                        +1  $EJECT
```

```
LOC  OBJ          SOURCE

                  ;******************************
                  ;
                  ; ENTRY POINTS
                  ;
                          PUBLIC  ZAPREG,BMCTRL,BMWAIT,BMREAD,BMWRIT,BMWRTB
                  ;
                  ;*********************
                  ;
                  ; MISC EQUATES
                  ;
 000B             REG1    EQU     0BH                  ; FIRST BMC REGISTER TO USE IS BLOCK LENGTH
 003C             STATER  EQU     3CH                  ; STATUS WORD ERROR MASK
                                                       ; IGNORE PARITY ERR, REV D OF BMC
              +1  $EJECT
```

```
LOC  OBJ          SOURCE

                  ;******************************
                  ;
                  ;    MODE BYTE DEFINITION
                  ;    ==== ==== ==========
                  ;
                  ; The bits in the MODE BYTE specify the type of the data transmission
                  ; TO USE, AND WHETHER TO PRINT STATUS AFTER EACH OPERATION.
                  ; If interrupts are enabled in the MODE BYTE, they must also be selected
                  ; in the ENABLE BYTE for desired operation to occur.
                  ;
 0001             INTMOD  EQU    01H          ; FIRST BIT IN MODE WORD IT INTERRUPT SELECT.
 0002             DMAMOD  EQU    02H          ; SECOND BIT IN MODE WORD IS DMA SELECT.
 0080             DBGMOD  EQU    80H          ; DEBUG BIT OF MODE WORD
              +1  $EJECT
```

```
LOC  OBJ                 LINE      SOURCE

                                   ;******************************
                                   ;
                                   ; FUNCTION: BMCTRL - PERFORM BMC CONTROL OPERATIONS (NON-DATA TRANSFER).
                                   ; INPUTS: NONE
                                   ; OUTPUTS: A=STATUS;F/F(C=1: AN ERROR OCCURED).
                                   ; CALLS: SNDREG,BMWAIT
                                   ; DESTROYS: ALL
                                   ; DESCRIPTION: THIS PROCEDURE IS USED TO PERFORM NON-DATA TRANSFER
                                   ;              BMC OPERATIONS.
                                   ;
0000                               BMCTRL:
0000 E8D700                                CALL    SNDREG                  ; LOAD BMC REGISTERS.
0003 A00000          E                     MOV     AL,BMCMD                ; GET COMMAND.
0006 E6E1                                  OUT     BMSTAT,AL               ; INITIATE COMMAND.
0008 E80E00                                CALL    BMWAIT                  ; WAIT FOR COMPLETION.
000B 243C                                  AND     AL,STAIER               ; DO WE HAVE AN ERROR?
000D A00000          E                     MOV     AL,STATUS               ; LOAD STATUS INTO 'A' FOR EXIT
0010 7502                                  JNZ     SHORT CTRL99            ; ERROR, RETURN WITH FLAG SET.
0012 F8                                    CLC                             ; CLEAR CARRY(ERROR FLAG)
0013 C3                                    RET                             ; AND RETURN
                                   ;
                                   ; WE HAD AN ERROR, RETURN WITH ERROR FLAG(CARRY FLAG) SET.
                                   ;         THIS IS THE GENERAL ERROR EXIT
                                   ;
0014                               CTRL99:
0014 A20000          E                     MOV     STATUS,AL
0017 F9                                    STC                             ; SET ERROR FLAG (CARRY FLAG)
0018 C3                                    RET                             ; AND RETURN.
                                   ;******************************
                                   ;
                                   ; FUNCTION: BMWAIT
                                   ; INPUTS: NONE
                                   ; OUTPUTS: STATUS IN A
                                   ; CALLS: NOTHING
                                   ; DESTROYS: A,F/F
                                   ; DESCRIPTION: THIS PROCEDURE WILL WAIT UNTIL THE CURRENT BMC
                                   ;              OPERATION COMPLETES.
                                   ;
0019                               BMWAIT:
                                   ;
                                   ; CHECK CURRENT STATUS (GOOD ONLY IF RAC=0 AND BSY=0)
                                   ;
0019 E4E1                                  IN      AL,BMSTAT               ; GET BMC STATUS
001B A880                                  TEST    AL,BUSYBT               ; CHECK BUSY BIT.
001D 740B                                  JZ      SHORT WAITEX            ; NOT BUSY, ALREADY DONE.
001F B9FFFF                                MOV     CX,0FFFFH               ; JUST IN CASE...
0022                               WAITPO:                                 ; POLLED WAIT MODE
0022 E4E1                                  IN      AL,BMSTAT               ; GET STATUS
0024 A880                                  TEST    AL,BUSYBT               ; CHECK BUSY BIT
0026 E0FA                                  LOOPNZ  WAITPO                  ; LOOP IF STILL BUSY
0028 E3EA                                  JCXZ    CTRL99                  ; PROBABLY AN ERROR IF CX=0
002A                               WAITEX:                                 ; CORRECT STATUS AND RETURN.
002A A20000          E                     MOV     STATUS,AL               ; A = STATUS
002D C3                                    RET
                          +1       $EJECT
```

```
LOC  OBJ                 SOURCE

                         ;****************************
                         ;
                         ; FUNCTION: BMREAD
                         ; INPUTS: CX = NUMBER OF BYTES TO READ, ES SET TO DS
                         ; OUTPUTS: A = STATUS; F/F(C=1: ERROR OCCURED)
                         ;          BX = NUMBER OF BYTES READ
                         ; CALLS: SNDREG
                         ; DESTROYS: ALL
                         ; DESCRIPTION: ALL PARAMETERS ARE PASSED THROUGH COMMON(PUBLIC)
                         ;              VARIABLES( SEE MODULE HEADER).
                         ;
002E                     BMREAD:
002E 32C0                        XOR     AL,AL               ; A = 0
0030 A20000        E             MOV     STATUS,AL           ; CLEAR STATUS.
0033 8BD9                        MOV     BX,CX               ; SAVE BYTE COUNT FOR LOOP
0035 E8A200                      CALL    SNDREG              ; SEND REGISTERS TO BMC.
0038 8B3E0000      E             MOV     DI,BUFADR           ; SET UP DEST BFR PTR (IN EXTRA SEG)
003C 8CD8                        MOV     AX,DS
003E 8EC0                        MOV     ES,AX               ; SET EXTRA SEG FOR BYTE MOVE DEST
0040 A00000        E             MOV     AL,BMCMD            ; GET COMMAND
0043 E6E1                        OUT     BMSTAT,AL           ; ISSUE IT.
                         ;
0045 B9FFFF                      MOV     CX,0FFFFH
0048                     BMRD1:
0048 E4E1                        IN      AL,BMSTAT
004A A880                        TEST    AL,BUSYBT
004C E1FA                        LOOPZ   BMRD1               ; WAIT FOR BUSY, BUT NOT FOREVER
004E E3C4                        JCXZ    CTRL99              ; CX=0 PROBABLY AN ERROR
0050 8BCB                        MOV     CX,BX
                         ;
                         ;        READ LOOP
                         ;        ==== ====
                         ;
0052                     BMRD2:
0052 E4E1                        IN      AL,BMSTAT           ; GET STATUS
0054 A801                        TEST    AL,FIFOBT           ; FIFO EMPTY?
0056 7407                        JZ      SHORT BMRD3         ; YEP, GO CHECK FOR BUSY.
0058 E4E0                        IN      AL,BMDATA           ; NOPE, GET DATA
005A AA                          STOSB                       ; STORE IT
005B E2F5                        LOOP    BMRD2               ; AND GO FOR MORE.
005D EBBA                        JMP     BMWAIT              ; XFER DONE, WAIT FOR A GOOD STATUS
005F                     BMRD3:                              ; NOTHING IN FIFO, IS OP COMPLETE?
005F A880                        TEST    AL,BUSYBT           ; CHECK BUSY BIT
0061 75EF                        JNZ     BMRD2               ; STILL BUSY, WAIT.
0063 2BD9                        SUB     BX,CX               ; BX <- # OF BYTES XFERED
0065 EBAD                        JMP     CTRL99
                     +1  $EJECT
```

```
LOC  OBJ                    SOURCE

                            ;******************************
                            ;
                            ; FUNCTION: BMWRIT - WRITE BUBBLE MEMORY DATA.
                            ; INPUTS: CX = # OF BYTES TO WRITE.
                            ; OUTPUTS: A = STATUS; F/F(C=1:ERROR OCCURED), BX=# OF BYTES WRITTEN.
                            ; CALLS: SNDREG,BMWAIT.
                            : DESTROYS: ALL.
                            ; DESCRIPTION: THIS PROCEDURE PERFORMS A BUBBLE MEMORY WRITE OPERATION.
                            ;              AN ERROR WILL OCCUR IF THE NUMBER OF BYTES GIVEN FOR THE
                            ;              WRITE OPERATION EXCEED THE NUMBER THAT THE BMC EXPECTS
                            ;              (DERIVED FROM COMMAND, BLOCK LENGTH AND NUMBER OF FSA
                            ;               CHANNELS), OR IF THE NUMBER OF BYTES IS LESS THAN THAT
                            ;              WHICH THE BMC EXPECTS.
                            ;
0067                        BMWRIT:
0067 32C0                           XOR     AL,AL                   ; A = 0
0069 A20000          E              MOV     STATUS,AL               ; CLEAR STATUS
006C 8BD9                           MOV     BX,CX
006E B01D                           MOV     AL,CFR
0070 E6E1                           OUT     BMSTAT,AL               ; FIFO RESET
0072 E86500                         CALL    SNDREG                  ; SEND REGISTERS TO BMC.
0075 8B360000        E              MOV     SI,BUFADR               ; SET UP SRC BFR PTR (IN DATA SEG)
0079 A00000          E              MOV     AL,BMCMD                ; GET COMMAND
007C E6E1                           OUT     BMSTAT,AL               ; ISSUE IT.
007E                        WRIT01:
007E E4E1                           IN      AL,BMSTAT
0080 A880                           TEST    AL,BUSYBT               ; WAIT FOR BUSY...
0082 74FA                           JZ      WRIT01
0084 A801                           TEST    AL,FIFOBT               ;   AND FIFO READY
0086 74F6                           JZ      WRIT01
                            ;
                            ; KEEP STUFFING DATA INTO FIFO UNTIL DONE OR AN ERROR OCCURS.
                            ; (NOTE: BMC GOING NOT BUSY IS AN ERROR).
                            ;
0088                        WRIT03:
0088 E4E1                           IN      AL,BMSTAT               ; GET STATUS
008A A801                           TEST    AL,FIFOBT               ; FIFO READY?
008C 7407                           JZ      WRIT04                  ; NO. WAIT FOR IT
008E AC                             LODSB                           ; YES, GET DATA FOR IT
008F E6E0                           OUT     BMDATA,AL               ; GIVE IT TO BMC
0091 E2F5                           LOOP    WRIT03                  ; LOOP UNTIL DONE.
0093 EB84                           JMP     BMWAIT                  ; XFER DONE, WAIT FOR A GOOD STATUS
0095                        WRIT04:
0095 A880                           TEST    AL,BUSYBT
0097 75EF                           JNZ     WRIT03                  ; OK IF STILL BUSY
0099 2BD9                           SUB     BX,CX                   ; BX <- # OF BYTES XFERED
009B E976FF                         JMP     CTRL99                  ; ERROR IF NOT BUSY AND CX NOT ZERO
                            ;
                            ; SPECIAL WRITE FOR BOOTLOOP AND BOOTLOOP REG CMNDS
                            ;
009E                        BMWRTB:
009E 32C0                           XOR     AL,AL                   ; A = 0
00A0 A20000          E              MOV     STATUS,AL               ; CLEAR STATUS
00A3 8BD9                           MOV     BX,CX
00A5 B01D                           MOV     AL,CFR
```

```
LOC  OBJ                  SOURCE

00A7 E6E1                          OUT     BMSTAT,AL             ; FIFO RESET
00A9 E82E00                        CALL    SNDREG                ; SEND REGISTERS TO BMC.
00AC 8B360000       E              MOV     SI,BUFADR             ; SET UP SRC BFR PTR (IN DATA SEG)
                          ;
                          ; FILL FIFO WITH 20/40/41 BYTES
                          ;
00B0                      WRTB01:
00B0 AC                            LODSB
00B1 E6E0                          OUT     BMDATA,AL             ; STICK IN FIFO.
00B3 E2FB                          LOOP    WRTB01                ; LOOP UNTIL FILL COUNT=0.
00B5 A00000         E              MOV     AL,BMCMD
00B8 E6E1                          OUT     BMSTAT,AL             ; SEND CMND
00BA E95CFF                        JMP     BMWAIT
                     +1   $EJECT
```

```
LOC  OBJ                 SOURCE

                         ;******************************
                         ;
                         ; FUNCTION: ZAPREG - ZAP ALL INTERNAL REGISTERS.
                         ; INPUTS: NONE
                         ; OUTPUTS: NONE
                         ; CALLS: NOTHING
                         ; DESTROYS: NOTHING.
                         ; DESCRIPTION: SET ALL INTERNAL REGISTERS EXCEPT 'ENABLE' TO AN
                         ;              ACCEPTABLE VALUE.  NOTE:  AN ACCEPTABLE VALUE MAY
                         ;              OR MAY NOT BE THE ONE DESIRED AS A DEFAULT.
                         ;
00BD                     ZAPREG:
00BD 9C                          PUSHF                               ; SAVE FLAGS
00BE 50                          PUSH    AX                          ; SAVE REGISTERS
00BF 53                          PUSH    BX
00C0 BB0000                      MOV     BX,0
00C3 891E0000       E            MOV     PAGENO,BX                   ; STARTING PAGE NUMBER = 0
00C7 43                          INC     BX
00C8 891E0000       E            MOV     BLKLEN,BX                   ; BLOCK LENGTH = 1
00CC 32C0                        XOR     AL,AL
00CE A20000         E            MOV     BBLNUM,AL                   ; BUBBLE NUMBER = 0
00D1 FEC0                        INC     AL
00D3 A20000         E            MOV     NFC,AL                      ; # OF FSA CHANNELS = 1 (2 CHANNELS)
00D6 5B                          POP     BX                          ; RESTORE REGISTERS.
00D7 58                          POP     AX
00D8 9D                          POPF
00D9 C3                          RET
                     +1  $EJECT
```

```
LOC  OBJ                  SOURCE

                          ;******************************
                          ;
                          ; FUNCTION: SNDREG - FORMAT AND SEND INTERNAL REGISTERS TO BMC.
                          ; INPUTS: NONE
                          ; OUTPUTS: NONE
                          ; DESTROYS: NOTHING.
                          ; DESCRIPTION: FORMAT AND SEND ALL INTERNAL REGISTERS TO THE BMC.
                          ;
00DA                      SNDREG:
00DA 9C                           PUSHF
00DB 50                           PUSH    AX                  ; SAVE REGISTERS
00DC 53                           PUSH    BX
00DD 51                           PUSH    CX
00DE B00B                         MOV     AL,REG1             ; GET FIRST REGISTER ADDRESS.
00E0 E6E1                         OUT     BMSTAT,AL           ; SELECT IT.
                          ;
                          ; CONSTRUCT AND SEND BLOCK LENGTH.
                          ;
00E2 8B1E0000        E            MOV     BX,BLKLEN           ; HL = BLOCK LENGTH
00E6 8AC3                         MOV     AL,BL               ; A  = BLOCK LENGTH LSB
00E8 E6E0                         OUT     BMDATA,AL           ; GIVE IT TO BMC.
00EA A00000          E            MOV     AL,NFC              ; A  = NUMBER OF FSA CHANNELS.
00ED B104                         MOV     CL,4
00EF D2E0                         SHL     AL,CL
00F1 0AC7                         OR      AL,BH               ; MERGE INTO BLOCK MSB
00F3 E6E0                         OUT     BMDATA,AL           ; GIVE IT TO BMC.
                          ;
                          ; SEND ENABLE BYTE.
                          ;
00F5 A00000          E            MOV     AL,ENABLE           ; GET ENABLE BYTE
00F8 E6E0                         OUT     BMDATA,AL           ; GIVE IT TO BMC
                          ;
                          ; CONSTRUCT AND SEND ADDRESS REGISTER.
                          ;
00FA 8B1E0000        E            MOV     BX,PAGENO           ; HL = STARTING PAGE NUMBER
00FE 8AC3                         MOV     AL,BL               ; A  = ADDRESS REGISTER LSB
0100 E6E0                         OUT     BMDATA,AL           ; GIVE IT TO BMC.
0102 A00000          E            MOV     AL,BBLNUM           ; A  = BUBBLE NUMBER
0105 B103                         MOV     CL,3
0107 D2E0                         SHL     AL,CL
0109 0AC7                         OR      AL,BH               ; MERGE INTO PAGE NUMBER MSB.
010B E6E0                         OUT     BMDATA,AL           ; GIVE IT TO BMC.
                          ;
                          ; RESTORE REGISTERS AND RETURN.
                          ;
010D 59                           POP     CX
010E 5B                           POP     BX
010F 58                           POP     AX
0110 9D                           POPF
0111 C3                           RET
                     +1   $EJECT
```

```
LOC  OBJ                LINE     SOURCE

                         405     CODE     ENDS
                         406              END
```

```
XREF SYMBOL TABLE LISTING
---- ------ ----- -------

NAME      TYPE      VALUE  ATTRIBUTES, XREFS

??SEG . . SEGMENT          SIZE=0000H PARA PUBLIC
BBLNUM. . V BYTE    0000H  EXTRN  16# 346 391
BLKLEN. . V WORD    0000H  EXTRN  15# 344 372
BMCMD . . V BYTE    0000H  EXTRN  16# 167 229 279 322
BMCTRL. . L NEAR    0000H  CODE PUBLIC  131 165#
BMDATA. . NUMBER    00E0H    54# 247 296 320 374 379 384 390 395
BMRD1 . . L NEAR    0048H  CODE  233# 236
BMRD2 . . L NEAR    0052H  CODE  243# 249 253
BMRD3 . . L NEAR    005FH  CODE  246 251#
BMREAD. . L NEAR    002EH  CODE PUBLIC  131 221#
BMSTAK. . L NEAR    0000H  EXTRN  8#
BMSTAT. . NUMBER    00E1H    53# 168 197 202 230 234 244 276 280 282 292 312 323 368
BMWAIT. . L NEAR    0019H  CODE PUBLIC  131 169 193# 250 298 324
BMWRIT. . L NEAR    0067H  CODE PUBLIC  131 271#
BMWRTB. . L NEAR    009EH  CODE PUBLIC  131 307#
BPADR . . V WORD    0000H  EXTRN  23#
BUFADR. . V WORD    0000H  EXTRN  15# 226 278 314
BUSYBT. . NUMBER    0080H    65# 198 203 235 252 283 300
CAB . . . NUMBER    0019H    43#
CFR . . . NUMBER    001DH    47# 275 311
CIZ . . . NUMBER    0011H    35#
CODE. . . SEGMENT          SIZE=0112H PARA PUBLIC   78# 79 405
CORERR. . NUMBER    0008H    61#
CPURG . . NUMBER    001EH    48#
CRB . . . NUMBER    001BH    45#
CRBR. . . NUMBER    0015H    39#
CRCD. . . NUMBER    001CH    46#
CRD . . . NUMBER    0012H    36#
CRFS. . . NUMBER    0018H    42#
CRS . . . NUMBER    0014H    38#
CSR . . . NUMBER    001FH    49#
CTRL99. . L NEAR    0014H  CODE  172 179# 205 237 255 303
CWB . . . NUMBER    0017H    41#
CWBR. . . NUMBER    0016H    40#
CWBRM . . NUMBER    0010H    34#
CWD_. . . NUMBER    0013H    37#
CWRS. . . NUMBER    001AH    44#
DATA. . . SEGMENT          SIZE=0000H PARA PUBLIC   11# 28 79
DBGMOD. . NUMBER    0080H    153#
DEFADR. . V WORD    0000H  EXTRN  13#
DEFBLK. . V WORD    0000H  EXTRN  14#
DEFBUB. . V BYTE    0000H  EXTRN  13#
DEFENA. . V BYTE    0000H  EXTRN  13#
DEFMOD. . V BYTE    0000H  EXTRN  14#
DEFNFC. . V BYTE    0000H  EXTRN  13#
DEFPAG. . V WORD    0000H  EXTRN  14#
DMAENA. . NUMBER    0004H    71#
DMAMOD. . NUMBER    0002H    152#
ENABLE. . V BYTE    0000H  EXTRN  15# 383
FIFOBT. . NUMBER    0001H    58# 245 285 293
ICDENA. . NUMBER    0040H    75#
```

```
NAME       TYPE      VALUE  ATTRIBUTES, XREFS
IERENA. . NUMBER    0002H    70#
INBUF . . V BYTE    0000H  EXTRN  17#
INBUFA. . V WORD    0000H  EXTRN  18#
INBUFC. . V BYTE    0000H  EXTRN  17#
INBUFL. . V BYTE    0000H  EXTRN  18#
INBUFP. . V WORD    0000H  EXTRN  17#
INTENA. . NUMBER    0001H    69#
INTMOD. . NUMBER    0001H    151#
LEVMSK. . V BYTE    0000H  EXTRN  22#
MODE. . . V BYTE    0000H  EXTRN  16#
MYBUF . . V BYTE    0000H  EXTRN  12#
NFC . . . V BYTE    0000H  EXTRN  16# 348 375
OPDONE. . NUMBER    0040H    64#
OPFAIL. . NUMBER    0020H    63#
OUTBFA. . V WORD    0000H  EXTRN  20#
OUTBFC. . V BYTE    0000H  EXTRN  19#
OUTBFL. . V BYTE    0000H  EXTRN  20#
OUTBFP. . V WORD    0000H  EXTRN  19#
OUTBUF. . V BYTE    0000H  EXTRN  19#
PAGENO. . V WORD    0000H  EXTRN  15# 342 388
PARERR. . NUMBER    0002H    59#
POPREGS . V WORD    0000H  EXTRN  24#
PROMPT. . V BYTE    0000H  EXTRN  22#
PUSHREGS. V WORD    0000H  EXTRN  24#
RAM . . . V BYTE    0000H  EXTRN  12#
RCDENA. . NUMBER    0020H    74#
RDLEN . . V WORD    0000H  EXTRN  21#
REG1. . . NUMBER    000BH    137# 367
RSVD1 . . NUMBER    0008H    72#
RSVD2 . . NUMBER    0080H    76#
SCRBUF. . V BYTE    0000H  EXTRN  12#
SNDREG. . L NEAR    00DAH  CODE   166 225 277 313 362#
STACK . . SEGMENT          SIZE=0000H PARA STACK
STATER. . NUMBER    003CH    138# 170
STATUS. . V BYTE    0000H  EXTRN  16# 171 180 207 223 273 309
TIMERR. . NUMBER    0010H    62#
UNCERR. . NUMBER    0004H    60#
USERBP. . V WORD    0000H  EXTRN  25#
USERBX. . V WORD    0000H  EXTRN  25#
USERCS. . V WORD    0000H  EXTRN  26#
USERDS. . V WORD    0000H  EXTRN  25#
USERFL. . V WORD    0000H  EXTRN  26#
USERIP. . V WORD    0000H  EXTRN  26#
USERPC. . V WORD    0000H  EXTRN  27#
USERRG. . V WORD    0000H  EXTRN  23#
USERSP. . V WORD    0000H  EXTRN  26#
USERSS. . V WORD    0000H  EXTRN  25#
WAITEX. . L NEAR    002AH  CODE   199 206#
WAITPO. . L NEAR    0022H  CODE   201# 204
WBLENA. . NUMBER    0010H    73#
WRITO1. . L NEAR    007EH  CODE   281# 284 286
WRITO3. . L NEAR    0088H  CODE   291# 297 301
WRITO4. . L NEAR    0095H  CODE   294 299#
WRLEN . . V WORD    0000H  EXTRN  21#
WRTBO1. . L NEAR    00B0H  CODE   318# 321
```

```
NAME      TYPE     VALUE  ATTRIBUTES, XREFS

ZAPREG. . L NEAR   00BDH  CODE PUBLIC  131 337#

ASSEMBLY COMPLETE. NO ERRORS FOUND
```

intel®

APPLICATION
NOTE

AP-127

December 1982

Powerfail Considerations for Magnetic Bubble Memories

Dick Pierce
Applications Engineer
Intel Corporation

December 1982
ORDER NUMBER 210300-001

INTRODUCTION

Intel has developed a new, comprehensive power-fail circuit that is incorporated into all Intel Bubble Board Memory products: BPK 72 Bubble Memory Prototype Kit, iSBX™ 251 MULTIMODULE™ board, and the iSBC® 254 MULTIBUS® compatible board. The use of this circuit also is recommended for all customer-designed bubble memory boards. The overall performance enhancements offered by this circuit include improved noise immunity and a factor-of-four reduction in the time required to shut down the bubble system.

Scope and Organization

In an effort to focus on implementation details, this application note is organized so that a reader can obtain sufficient information to implement a bubble design without an intimate working knowledge of the powerfail circuitry. However, for those interested, a complete detailed explanation of the integrated powerfail circuitry and the additional external circuitry is included. Appendix A contains a technical discussion of the effects of power loss on a Magnetic Bubble Chip. In addition, the previous circuit versions (Revision 0 and Revision 1), along with the present circuit, are completely documented and compared in Appendix B.

Bubble Memory Operation and the Powerfail Function

The power-fail circuitry is partially integrated into two of the five MBM support components, and additional required circuitry is provided by external components. Historically, several evolutionary improvements have been made in the external circuitry (see Table 1) to further reduce the risk of data loss following an abrupt power failure.

An essential feature of the bubble memory (MBM) is non-volatile data storage. This non-volatility results from two permanent magnets within the bubble device that produce a magnetic field (bias field) that maintain the magnetic domains, or bubbles (representing data) in the chip even when power is removed. The bubbles remain stationary in fixed positions until the data is accessed. To move the bubbles, an in-plane rotating magnetic field is induced by pulsing two mutually-perpendicular coils surrounding the bubble chip. Special conductor lines on the bubble chips provide all the current related functions for reading and writing to the bubble device. A special support IC produces current pulses (swap, relicate, and generate) to perform these functions. A complete set of support circuits provides the necessary timing and waveforms to precisely maneuver the bubbles to their desired positions. To prevent bubbles from moving to undesired positions, certain precautions must be observed.

As power is applied or removed, the system must prevent any current transients in the coils or bubble function conductors. If power is removed with the coils operating, the system must ensure that the coil currents are shut down in an orderly fashion to guarantee that the magnetic bubbles come to rest in stable, known positions. The powerfail reset circuit ensures that the system is powered up in an orderly manner and serves to alert the system should power fail. Both the power-up and

Table 1. Powerfail Reset Circuit Product History

Product	Powerfail Circuit Revision Level		
	0	1	2
BPK 72	July 1979 thru August 1982 Rev. A thru Rev. G	N/A	September 1982
iSBX™-251 Board	N/A	September 1981 thru October 1982	November 1982
iSBX-251C Board	N/A	N/A	July 1982
iSBC®-254 Board	December 1980 thru July 1982	July 1982 thru November 1982	November 1982

power-down sequences require a finite period of time to complete their functions until the sequence is complete. To allow proper execution of a power down sequence, the system voltages (+5V DC, +12V DC) must not decay to a level that prevents operation of the powerfail circuitry and critical bubble memory functions. In most power supply designs, adequate energy storage is available to provide enough "hold time" to complete an orderly shutdown. However, if dc power decays too rapidly sufficient time may not exist for a proper shutdown and may cause data to be lost within the MBM.

System Description

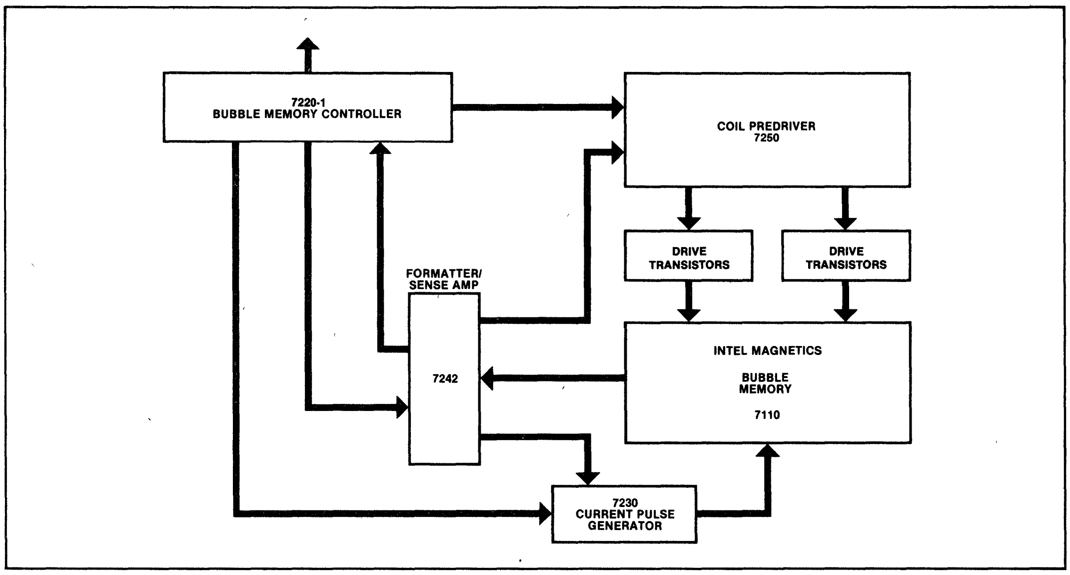
The basic Intel Bubble Memory system consists of one 7110 magnetic bubble memory and five integrated support components: a 7220-1 Bubble Memory Controller (BMC), a 7230 Current Pulse Generator (CPG), a 7242 Formatter-Sense Amplifier (FSA), a 7250 Coil Predriver (CPD), and two 7254 quad drive transistor packages. These support circuits are interfaced to the MBM as shown in Figure 1 to form the basic one megabit (128K byte) system. The support components provide all of the functions necessary for the storage and retrieval of data within the MBM. In addition, two of the support components, the 7220-1 BMC and the 7230 CPG, contain the integrated powerfail circuitry that facilitates proper power-up and power-down operations.

OVERVIEW — POWER UP/DOWN OPERATION

A block diagram of the power fail circuitry for the bubble memory system is shown in Figure 2. The following paragraphs provide an operational overview of the integrated powerfail circuit and the external circuit requirements.

During a power up sequence, the 7230 holds PWR.FAIL/* active (low) until both supplies are above the minimum required level. The 7230 contains power supply monitors (+5V and +12V) that determine when either supply falls below threshold level and activate PWR.FAIL/ signal accordingly. On power-up, the PWR.FAIL/ signal is delayed an additional 2 msec by an external RC network (time delay 1) to allow the 7220-1 substrate bias generator to fully charge. Following this delay, the positive-going transition on the 7220-1 PWR.FAIL/ input initiates a 7220-1 power-up sequence.

The RESET.OUT/ signal was designed to remain active during the power-up sequence and then to go inactive at the conclusion of the 50 μs power-up sequence. However, the RESET.OUT/ signal is indeterminate during execution of the 7220-1 power-up sequence. A second external RC network (time delay 2) derived from PWR.FAIL/ ensures that RESET.OUT/ is



*"/" denotes an inactive signal.

Figure 1. System Block Diagram

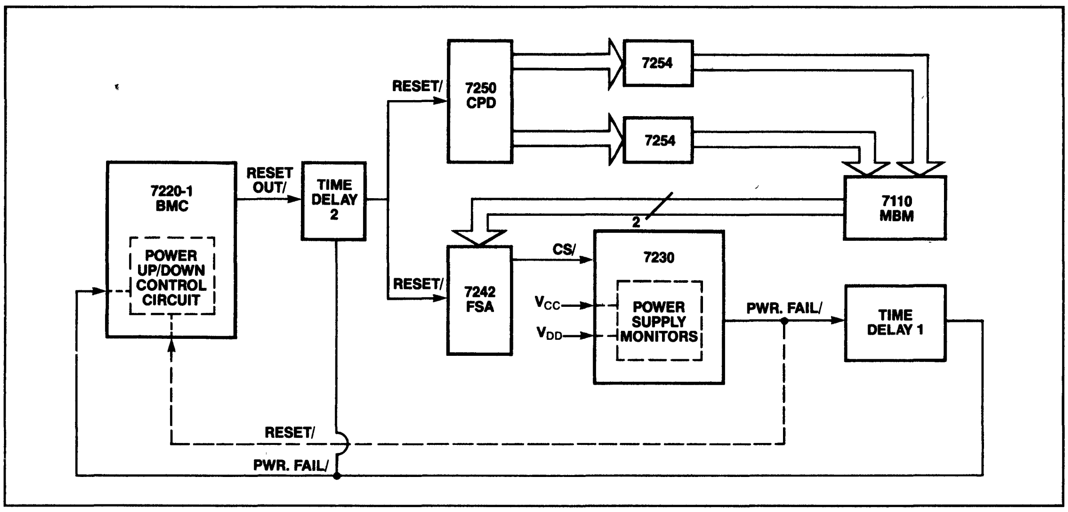


Figure 2. Block Diagram of Powerfail

held active (≤0.8V) during this time. The RESET.OUT/ signal occassionally will remain in its active state following a power-up sequence; accordingly the first command issued to the BMC during an initialization sequence must be an Abort command to ensure that RESET.OUT/ is deactivated.

The power-up sequence is designed to power the system up in an orderly fashion and to prevent any current transients from reaching the bubble device. The power-down sequence ensures that the coil drivers are shut down in the proper phase and that the support circuits are reset. When power fails, the 7230 notifies the 7220-1 by asserting the PWR.FAIL/ signal. The 7220-1 responds to a negative transition on either the PWR.FAIL/ input or the RESET/ input (external circuit revision level dependent) and initiates a power-down sequence. If the coils are active (i.e., bubbles propagating), the 7220-1 first terminates the coil drive control signals during the appropriate phase and then resets the support circuits by asserting the RESET.OUT/ signal. The two system supply voltages must not decay faster than the specified rates to ensure the RESET/ input to all the support circuits (excluding the 7220-1) reaches an active level (less than 0.8 volts).

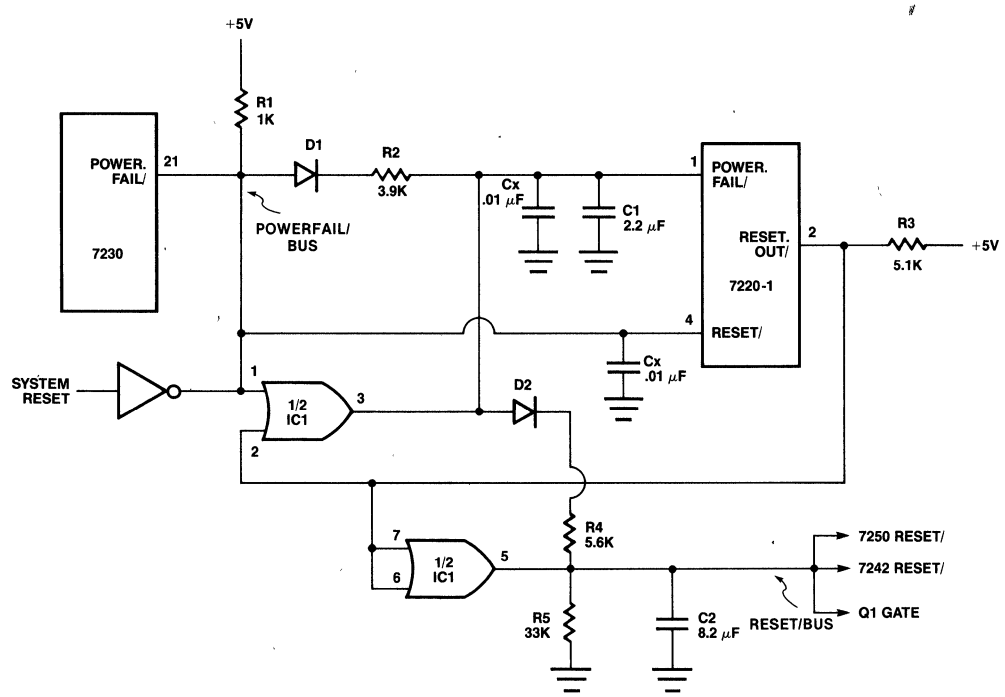
Powerfail Reset Circuit Solution

The external circuitry shown in Figure 3, in conjunction with the integrated circuitry contained in the 7230 and 7220-1, comprises the powerfail circuit (revision 2). This design contains six additional components compared to previous powerfail circuits and includes an 8-pin DIP IC (TI 75463).

This revised circuit has been fully developed and tested by Intel and currently is incorporated in many bubble products. Operational details are not required for the user to implement a custom design using the circuit in Figure 3. However, for any bubble memory designs that cannot conform to the recommended powerfail circuit, a reader must understand the system characteristics and requirements prior to choosing an alternative design.

The software implementation details to ensure correct powerfail circuit operation are shown in Figure 4. This routine should be implemented as a routine for cold start operation (application of power) and warm start operation (a RESET/ pulse applied to the 7220-1 BMC). The voltage decay rates shown in Table 2 also cannot be exceeded.

The power-up routine is based on the typical power-up timing shown in Figure 5. This timing does not assume that a system reset has been incorporated into the powerfail circuit. If the hardware reset line is used, the user must ensure that the 7220-1 RESET/ input is inactive before issuing the first Abort command. In addition, user software always must issue an Abort command every time the system is reset.



NOTES
1. ALL RESISTORS 1/4 WATT, 5% TOL
2 ALL CAPACITORS 10 VDC, 10% TOL
3 D1,D2 ARE IN914 OR EQUIVALENT
4 IC1 IN75463
5 Cx IS OPTIONAL—RECOMMENDED FOR EXTREMELY NOISY POWER SUPPLY SITUATIONS
6 SYSTEM RESET MAY BE USED AS SHOWN PROVIDED OUTPUT OF INVERTER IS OPEN COLLECTOR

Figure 3. External Powerfail Circuit Solution

Table 2. Power Supply Decay Rate Specifications During Power-down or Power Failure

Power Down/Powerfail Decay Rate			
V_{CC} (volts/msec)		V_{DD} (volts/msec)	
Min.	**Max.**	**Min.**	**Max.**
None	0.45	None	1.1

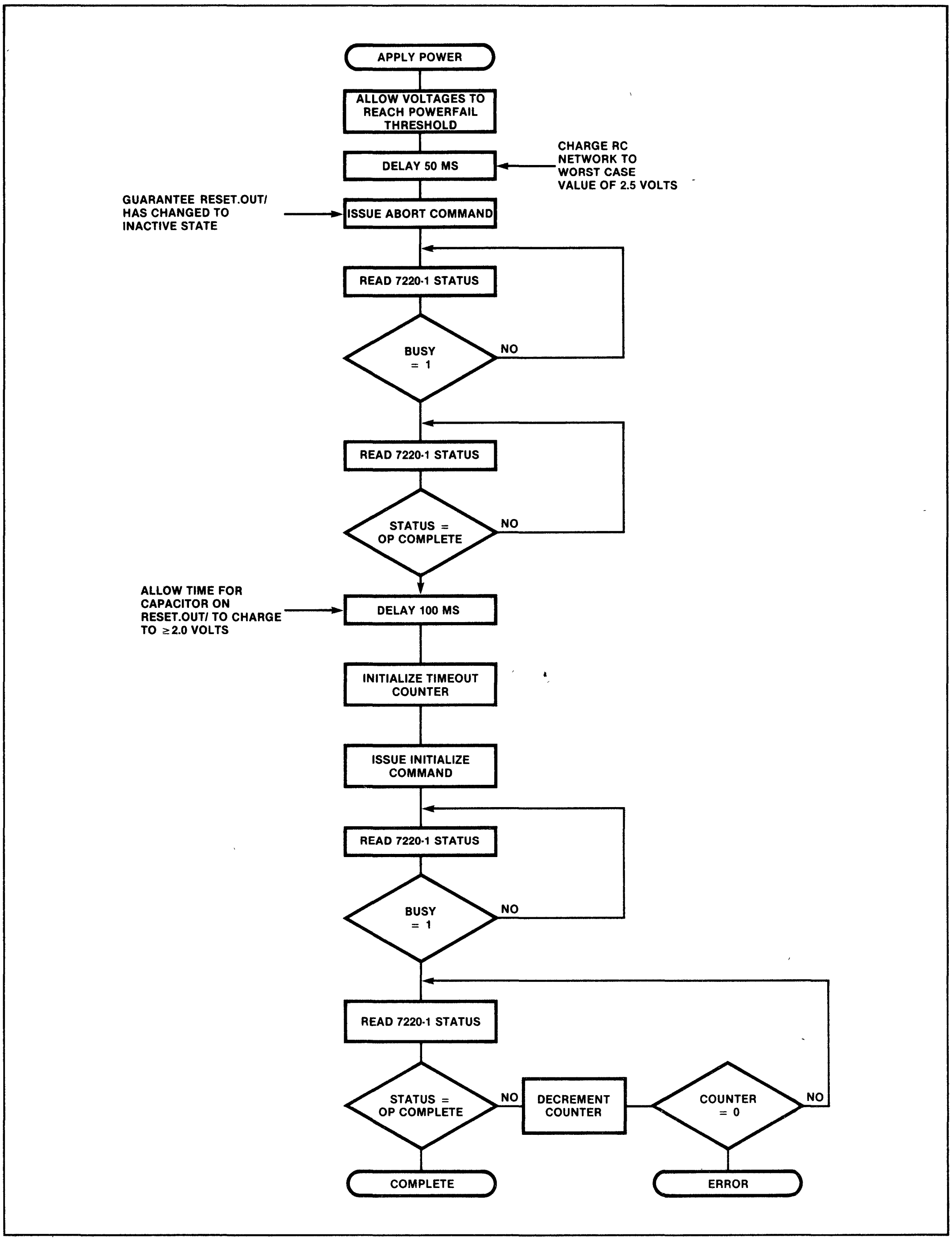


Figure 4. Power-up Flowchart

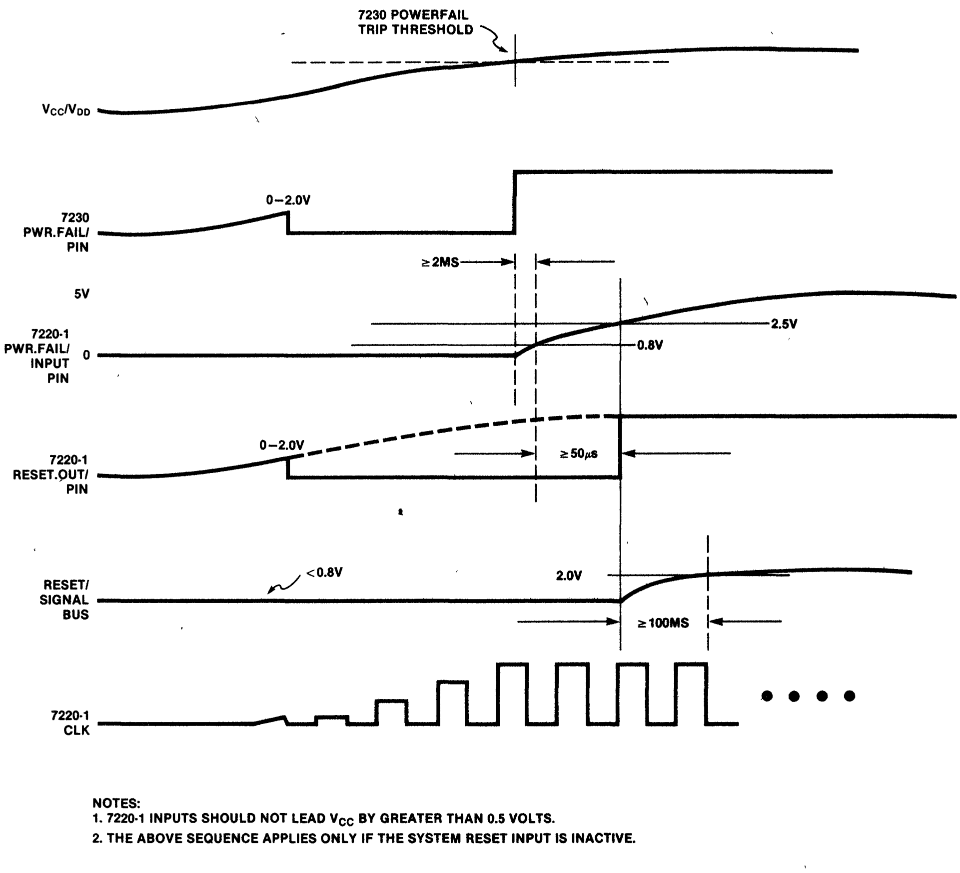


Figure 5. Power-up Timing for Powerfail Reset Circuit (Revision 2)

The worst case power-down timing sequence is also included in Figure 6. The total system power-down time varies according to whether the coils are active (i.e., rotating magnetic field is on) or inactive. The worst case power-down sequence is guaranteed to be completed provided that the above voltage decay rates are met.

INTEGRATED POWERFAIL CIRCUIT CHARACTERISTICS

Introduction

The following section provides an in-depth look at the input and output characteristics of the support circuits that contain the integrated powerfail circuitry. A complete understanding of these characteristics establishes the groundwork necessary for the detailed description of the overall powerfail circuit operation that follows.

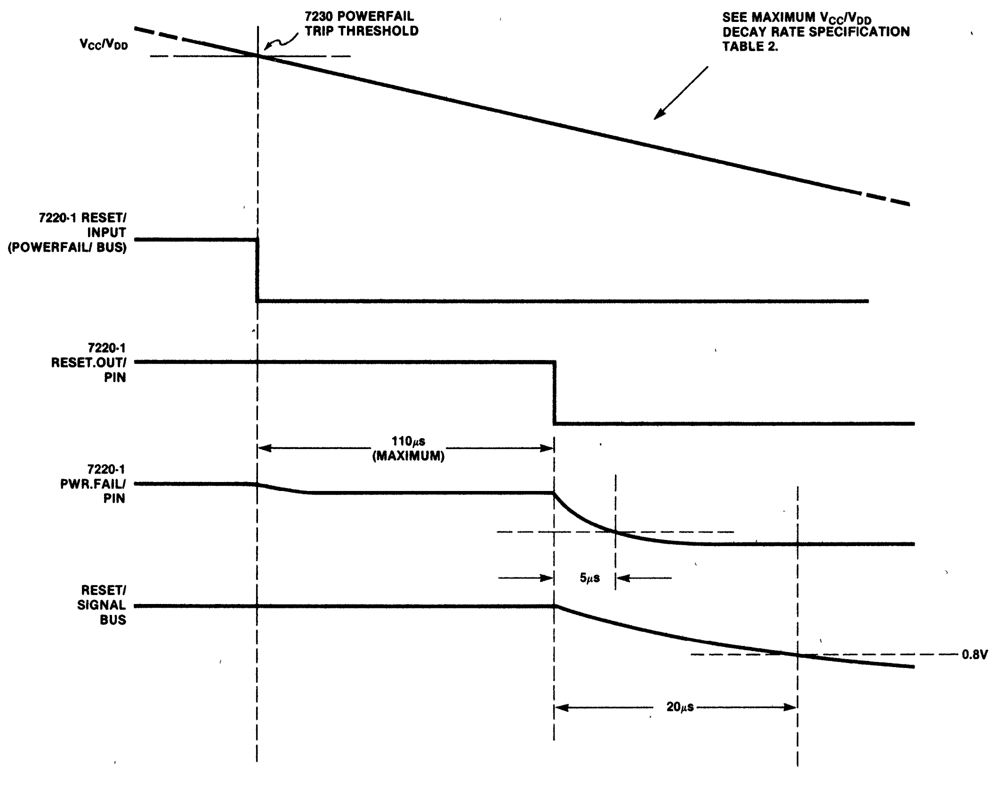


Figure 6. Power-down Timing for Powerfail Reset Circuit (Revision 2)

7230 PWR.FAIL/ OUTPUT

The 7230 Current Pulse Generator PWR.FAIL/ output is responsible for indicating when the system supply voltages (+5V, +12V) reach correct operating levels. During power up, normal operation, and power down, an internal zener reference comparator circuit within the 7230 senses both V_{CC} and V_{DD} and indicates when both levels are above approximately 92 percent of their nominal values. An active state on PWR.FAIL/ indicates one or both dc voltages are below this threshold. The PWR.FAIL/ output is an active-low, open-collector output requiring an external pullup resistor.

The PWR.FAIL/ output is asserted (active low) as power is applied until the +5V and +12V supplies both reach approximately their 92 percent levels at which point the 7230 output transistor switches off to allow the PWR.FAIL/ signal to rise to an inactive level governed by an external RC network. The RC networks on the PWR.FAIL/ line must hold the PWR.FAIL/ signal at an active level for at least 2.0 milliseconds to guarantee adequate time for the BMC to power up. The 7230 PWR.FAIL/ output then will remain inactive until one or both system voltages fall below the threshold.

The PWR.FAIL/ output is not an internally latched signal. In other words, the output responds immediately to any transition through the threshold (trip point). The disadvantage to this excellent response capability is that the output will toggle on transitions through the threshold. Systems should be designed to avoid an extremely noisy power supply or temporary power loss that could cause the PWR.FAIL/ signal to pulse for a very short duration.

During temporary power loss in Revision 0 and Revision 1 circuits, the PWR.FAIL/ input to the 7220-1 could pulse below V_{IH} (2.5 volts) and initiate a power down sequence. The 7220-1 PWR.FAIL/ input should remain active until the entire power down sequence is completed (maximum 110 μsec). As detailed later in the 7220-1 PWR.FAIL/ input description, if the 7220-1 PWR.FAIL/ input goes inactive during execution of a power down sequence, the sequence is immediately terminated. This type of termination can stop the drive field in the wrong phase and compromise bubble data. The solution is to use the 7220-1 RESET/ input to initiate a power down sequence rather than the 7220-1 PWR.FAIL/ input.

Two important considerations in properly designing a powerfail circuit are 1) the accuracy of threshold trip point of the 7230 PWR.FAIL/ output and 2) the behavior of this output at low voltages (below 2 volts).

The worst case threshold level that the 7230 PWR.FAIL/ output will trip must be above the worst case operating limits of the support circuits with an additional margin to allow for an adequate period of time to complete a power down sequence (worst case 110 microseconds for revision level 1 and 2 powerfail reset circuits). In the case of the 7230 CPG and the 7110 MBM, which both have a ±5% voltage specification for V_{CC} and/or V_{DD}, special powerfail characteristics are applicable. As shown below, (Table 3) only critical bubble memory functions are guaranteed at these supply values and not full memory operation.

Table 3. Powerfail Characteristics for 7230 Threshold Trip Point*

Symbol	Min.	Typ.	Max.
V_{CC} TH	4.43V	4.60V	4.70V
V_{DD} TH	10.75V	11.10V	11.28V

*Powerfail characteristics apply to 7110 bubble memory data integrity only and not to full memory operation

Second, the 7230 PWR.FAIL/ output cannot be guaranteed active (low) until V_{CC} reaches about 2.0 volts since the output transistor is not operational until that point. As V_{CC} is applied, the output is not active and will track (follow within a few tenths of a volt) V_{CC} until V_{CC} reaches approximately 2.0 volts. At this point, the output transistor turns on and the output goes active (low) and remains low until the system voltages both reach the threshold trip point as described earlier. A similar response occurs as power is removed. The output transistor turns on and pulls the output active (low) at the threshold point and remains turned on until V_{CC} reaches approximately 2.0 volts where the output goes inactive (transistor not operating). This operation must be controlled on power-up and depends on the rate of rise of system voltages. This is because the PWR.FAIL/ output is indirectly connected to the RESET/ input of the support circuits (7250 and 7242 and Q1 reference current switch) through two RC networks in Rev. 0 and Rev. 1 power-fail circuits. These inputs can rise to as much as 1.5V before the 7230 PWR.FAIL/ output turns on, which is above V_{IL} maximum (0.8V) thus potentially enabling these circuits. This could result in current transients reaching the drive coils or bubble function conductors and move bubbles from their rest position resulting in data loss. Observing the rate of rise specifications protects against this possibility . The revision 2 powerfail circuit eliminates this problem and has no rate of rise limitation.

7220-1 PWR.FAIL/ INPUT

The 7220-1 PWR.FAIL/ input serves a dual function; a positive transition initiates a power-up sequence while a negative transition initiates a power-down sequence of the bubble memory system. In order for the 7220-1 to become fully functional an on chip back-bias generator must fully charge the 7220-1 substrate. Therefore, before any sequence can be executed, including the power-up sequence a time delay is required. An external RC delay on the PWR.FAIL/ input ensures this input is held low (<0.8V) at least 2.0 milliseconds after V_{CC} has reached the 7220-1 voltage specification range.

The power-up sequence is initiated once the RC network charges to a point where the 7220-1 recognizes a positive transition on the PWR.FAIL/ input. From a cold start (application of power), a positive transition must occur or the controller will not power-up correctly. Once the power-up sequence is completed, the RESET.OUT/ is designed to be released, however, two possible exceptions exist. First, if the 7220-1 RESET/ is held low during power-up, the 7220-1 internal power-up sequence will be completed however RESET.OUT/ will not be released until RESET/ is inactive. Second, the 7220-1's internal RESET.OUT/ output transistor may remain turned on dependent upon the power-up status of certain internal 7220-1 flip-flops. Because of this an ABORT command is always necessary to internally reset these flip-flops, in turn ensuring release of the RESET.OUT/ output.

If the 7220-1 BMC does not receive a positive transition on PWR.FAIL/ during power-up, a power-up sequence is not initiated. This leaves the controller in an unknown state. In this unknown state the controller cannot communicate properly with the data and control inputs. This can only occur as a result of:

1. **"Brown out"** — short duration of power failure in which power drops below specified levels.
2. **Power-up circuit failure** — The PWR.FAIL/ pin never reaches V_{IH} (minimum) of 2.5 volts.

The above conditions are resolved by ensuring a positive transition occurs on the PWR.FAIL/ input during power-up and after brownout. It is necessary to execute a power-up sequence even though power to the system is only interrupted momentarily in order to restore the 7220-1 to the required internal state.

Once the PWR.FAIL/ positive transition has occured, this input should remain in the inactive state (V_{IH} > 2.5V) as long as power is applied to the system. If power is removed, it is the negative transition of this input which intitiates the second function, power down. The function can also be initiated with the RESET/ input of the 7220-1.

An important consideration is how the 7220-1 PWR.FAIL/ input distinguishes between positive and negative transitions. On power up (positive transition), crossing the input threshold (typically 1.6V to 1.9V) a pulse is generated internally which resets the 7220-1 to a known state and initiates a power-up sequence. On power down (negative transition), crossing the input threshold (typically 1.35V to 1.6V with the designed- in hysteresis) the signal initiates a power-down sequence. If a power-down sequence has been initiated, a positive transition must not inadvertently occur on the 7220-1 PWR.FAIL/ input prior to the power-down sequence completion. A positive transition internally generates a reset pulse (to halt any current BMC activity) and initiates a power-up sequence effectively terminating a power down sequence. The result is a possibility of shutting the coil drives down in the improper phase resulting in data loss in the MBM.

The PWR.FAIL/ input has built in hysteresis to reduce the susceptibility to multiple threshold crossings or glitching. However, the values of hysteresis range from 50 mV to 400 mV. To improve noise and power fluctuation immunity, the use of PWR.FAIL/ input for initiating a power down sequence was abandoned in Revision 1 and Revision 2 circuit designs. The 7220-1 RESET/ input is used instead to initiate power down (see next section.)

7220-1 RESET/ INPUT

The 7220-1 RESET/ input, when asserted, will terminate any current BMC activity and initiate a RESET sequence (identical to the sequence initiated by the PWR.FAIL/ input going active). After the sequence is concluded, the RESET.OUT/ is activated to reset the MBM support circuitry. RESET.OUT/ will remain active until RESET/ is inactive.

The RESET/ input is a level sensitive latched input. This is a distinct advantage over the PWR.FAIL/ input; where any fluctuations of the input once the signal was recognized could possibly terminate the power down sequence. The RESET/ input is latched on the negative edge of the BMC clock and must be active low (< .8V) for at least one clock period (250ns) to guarantee recognition.

7220-1 RESET.OUT/

The RESET.OUT/ output has two functions: 1) to guarantee the bubble memory system is disabled during power-up and after power down of the bubble memory system and 2) to provide a pulse (reset) to the support circuits during normal operation. Since the RESET.OUT/ output is an active low open drain, it requires an external pullup resistor to V_{CC}.

The support circuits controlled by RESET.OUT/ are the 7250 Coil Predriver, the 7242 Formatter Sense Amplifier, and a VMOS transistor switch which enables a reference current for the 7230. These circuits must be disabled during the entire power-up sequence and immediately following the conclusion of a power-down sequence to prevent any current transients or extraneous enable pulses. Data loss is a possible consequence should the support circuits not remain disabled during power cycling.

During power up the RESET.OUT/ signal can not be guaranteed active (low) until the 7220-1 power-up sequence has executed. Therefore, external circuitry must assure RESET.OUT/ does not rise above V_{IL} maximum (.8V) until 50 μs after initiation of the power-up sequence. By ensuring the RESET.OUT/ is active during power-up it guarantees the support circuits are reset to a known state. The 7220-1 BMC is designed with the capability to reset the support circuits during normal operations by pulsing the RESET.OUT/ 750 μs (3 clock periods). This pulse can occur as the result of two user issued commands to the BMC: an INITIALIZE command and an MBM PURGE command.

The external RC network on the RESET.OUT/ signal prevents the RESET.OUT/ pulse from going active during its 750 μs duration. In spite of an inability to reset the support circuits by issuing the proper command, correct operation is guaranteed since the support circuits only require a one time reset signal at power-on.

Additional Bubble Memory Controller Inputs

The 7220-1 has several additional inputs that could indirectly affect power up operation. It is important that the user exercise caution and adhere to all requirements to ensure proper power-up operations. The following outlines those requirements.

CLK (CLOCK)

The CLK input of the 7220-1 must be present when the positive power up transition occurs at the 7220-1 PWR.FAIL/ input. This requirement allows the BMC to properly execute a power-up sequence. The input requirements are a precise 4MHz (±.1%) with a 50 percent duty cycle (±5%).

DACK/ (DATA ACKNOWLEDGE)

The DACK/ input is normally used in conjunction with an INTEL DMA controller chip (8257 or 8237) which automatically provides drive for this input. However, if DMA is not used a 5.1K pullup resistor to V_{CC} is required. This requirement prevents erratic BMC operation.

WAIT/

The WAIT/ input must also be guaranteed inactive through an external 5.1K pullup resistor. It is designed to be used in parallel controller applications to maintain synchronization between controllers should an error be detected in one during a data transfer.

CS/, RD/, WR/, A0, D0-D8

These inputs require no special considerations other than to observe the V_{IH} minimum specification. This specification prevents an incorrect power-up sequence execution.

ENERGY STORAGE REQUIREMENTS

The data integrity and non-volatility of the MBM during power down operations is guaranteed by design provided the voltage decay rates specifications for both V_{CC} and V_{DD} are observed. Most commercially available power supplies provide enough energy storage to fulfill these requirements. However, some applications may exist where the bubble memory could suddenly become disconnected from the dc supply; a case where the power supply energy storage is not of value. In these special applications, the local onboard capacitance must meet the hold up time requirement.

The worst case onboard capacitance values can be determined according to the following equation:

$$C = \frac{Q\ max}{V\ min} = \frac{I\ max\ \Delta T\ max}{\Delta V\ min}$$

A worst case calculation must include the following considerations: 1) If any additional circuitry exists on the pc board that uses the same power supplies, the additional current drain must be accounted for and 2) the worst case (minimum) threshold trip point of the 7230 is used.

The capacitance required on a pc board containing one / megabit bubble memory system is calculated as follows:

$$C_{5V} = \frac{366 \times 10^{-3}\ \text{amp} \times (110 \times 10^{-6}\ \text{sec})}{0.01 \quad \times 5\ \text{volts}} = 805\ \mu F$$

$$C_{12V} = \frac{381 \times 10^{-3}\ \text{amp} \times (110 \times 10^{-6}\ \text{sec})}{0.01 \quad \times 12\ \text{volts}} = 350\ \mu F$$

Supplemental Powerfail Sensing

In many systems, additional signals are available that provide advanced warning of an imminent power failure or the existence of an abnormal condition prior to actual loss of dc power (e.g., AC powerfail sensing, AC or DC overvoltage, ambient over/under temperature). These signals are easily incorporated into the powerfail circuit design via an open-collector gate or inverter connected to the PWR.FAIL/ signal bus.

The advantage of utilizing these signals is the bubble memory system can complete a power down sequence prior to losing dc power. However, local dc powerfail sensing is always required due to the possibility of local dc power loss without the loss of AC power.

Noise Effects of Powerfail Circuit Operation

The 7230's powerfail voltage monitoring function is implemented internally with two independent, logically-OR'ed voltage comparators. The comparators respond quickly to a sudden loss of V_{CC} or V_{DD} and therefore can respond to noise transients on the power supply lines that cross the comparator switching threshold. As much as 100 mV of noise from coil drive switching is not uncommon. Note that the operating power supply tolerance for all INTEL Bubble Memory products is ±5% including up to 50 mV of noise on the power supply lines. This tolerance should not be confused with the operation of the powerfail circuit beyond the normal operating range during power-down operation.

To minimize "nuisance" activation of the PWR.FAIL/ signal bus, ample high frequency decoupling on the 7230's V_{CC} and V_{DD} pins should be provided. Typically, 0.01 μF to 0.1 μF ceramic disk or mica capacitors are sufficient. Another source of unwanted powerfail circuit activation is noise that is coupled directly onto the PWR.FAIL/ signal bus. This noise is minimized through good printed circuit layout practices and, if required, by the inclusion of a small capacitor directly on the PWR.FAIL/ bus. This capacitor slightly increases the power-down time and should be kept as small as possible (0.01 μF maximum).

APPENDIX A

TECHNICAL DISCUSSION OF POWER LOSS EFFECT ON 7110

The effects of power loss on an MBM are best understood by describing the way in which the device functions and the way in which it can lose data.

A magnetic bubble memory device (See Figure 7) consists of a bubble memory chip, two mutually-perpendicular coils, two permanent magnets, and a shield to provide protection from interference by external magnetic fields. The two permanent magnets produce an external magnetic field (bias field) that maintains the magnetic domains, or bubbles, in the chip even when power is removed. To move the bubbles, an in-plane rotating magnetic field is induced by pulsing the two mutually-perpendicular coils.

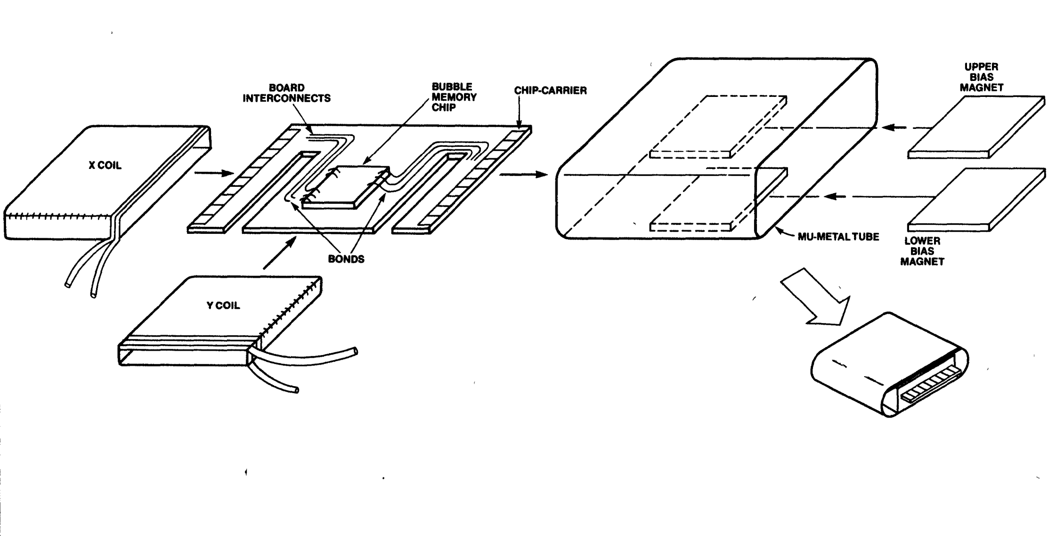


Figure 7. Device Break-down

The bubble memory chip itself consists of a thin magnetic garnet crystal film grown on a non-magnetic gadolinium-gallium-garnet substrate. This thin film possesses a property that magnetic moments associated with each atom in the single crystal structure have only two possible directions: an upward or downward direction perpendicular to the plane of the film. This constraint in direction results in only two conditions of magnetization (see Figure 8). These magnetic moments tend to group themselves together into magnetic domains. The size and shape of the domains are determined primarily by a balancing of several forces that minimize the sum of magnetic energy.

Without an external field, the film surface area of upward domains is equal to that of downward domains and there is no net magnetic field within the plane of film. Application of an external magnetic field perpendicular to the film causes domains to line up in the direction of the field. As the external field is increased, the downward domains enlarge while the opposing (upward) domains shrink until they finally are reduced to a cylindrical shape. This microscopic magnetized cylinder opposing the externally applied field is a magnetic bubble. Within the magnetic film, the presence of a magnetic bubble represents a binary one and the absence of a magnetic bubble represents a binary zero.

The memory function is provided by the bubble. However, an organized means is needed to propagate the bubbles along certain paths and to provide storage sites. A soft ferromagnetic material (permalloy) is deposited on the thin garnet film in C-shaped patterns. These patterns are arranged to form shift-register like loops that provide the means to store and move bubbles. Each pattern is magnetized according to the rotating magnetic field, and the polarity of each pattern changes instantaneously as the rotating magnetic field vector changes. The rotating field is generated by driving the X and Y coils with triangular-waveform currents, one lagging the other by 90° in phase. A magnetic bubble propagates from one storage site (permalloy pattern) to the next for every 360° of rotation of the rotating field. Each storage site has a preferred position (home) for the bubble to reside corresponding to zero degrees of the rotating magnetic field. All bubbles start, stop and are stored in this position.

In the event of power failure, it is important that the rotating magnetic field is shut down in the proper phase (i.e., 0°). If an orderly shut down is not complete, the rotating field may be shut down in an improper phase that causes bubbles to stop in an unstable position within the storage loops. When this type of stoppage occurs, the bubbles either will come to rest in another, but incorrect, stable position or will leave their original storage loop (possibly contaminating valid data in another storage loop).

As power is applied, it also is important that the rotating magnetic field does not move (i.e., current transients must be prevented from reaching the coils). This function also is provided via the powerfail circuitry. Thus, the purpose of the powerfail circuitry is twofold 1) to prevent any current transients from reaching the X-Y coils or bubble function generators and 2) to halt the coils in proper phase should power fail.

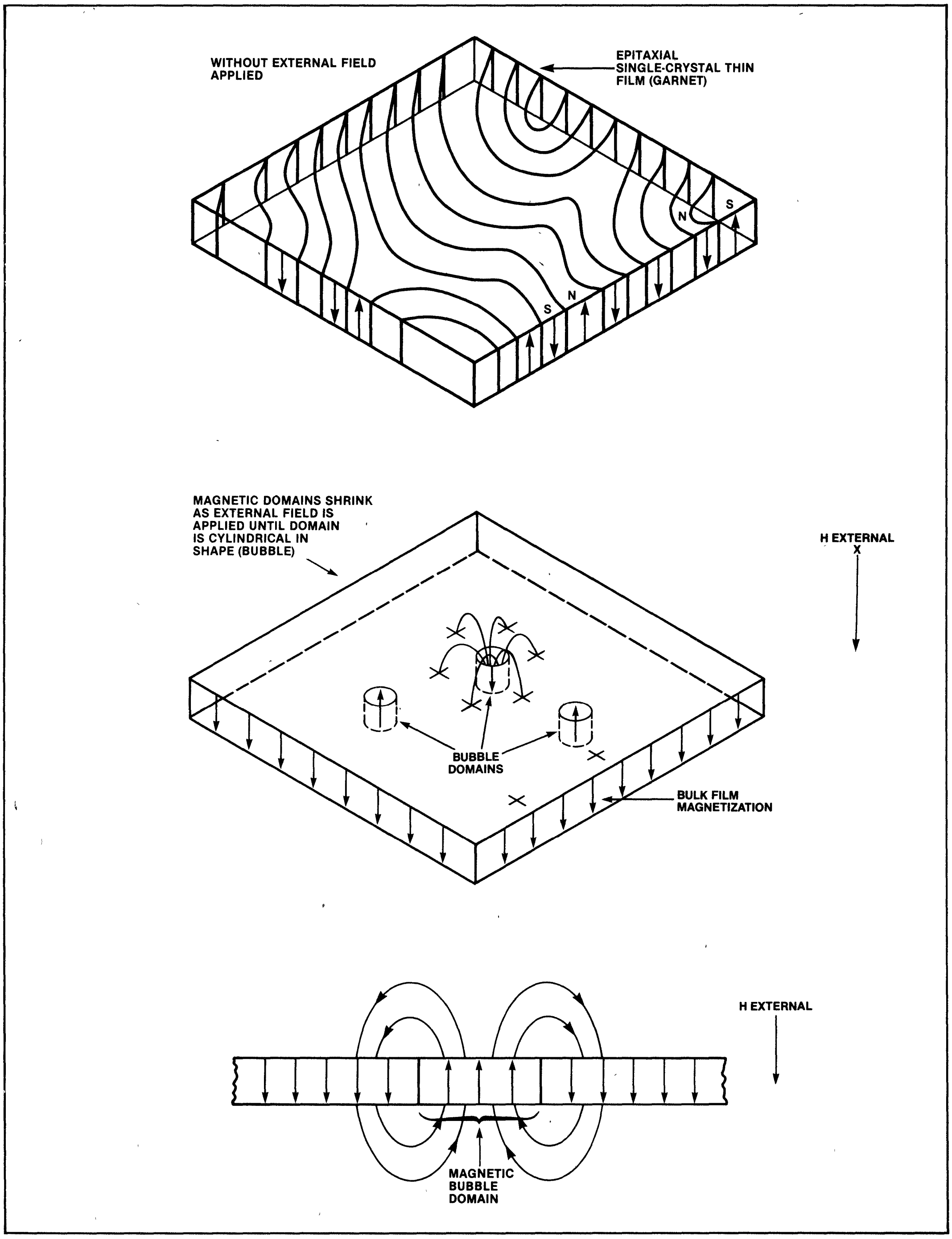


Figure 8. Device Magnetization

APPENDIX B

DETAIL POWER CIRCUIT DESCRIPTION

As discussed in the Introduction, the powerfail reset circuit actually consists of two portions — an integrated section and several additional external components. The degree to which external disturbances (noise, power fluctuations) influence system performance depends heavily on the system environment and configuration. Consequently, the reliable analysis of their effect on system performance is difficult and generally is best accomplished by measurement. In this Appendix, each revision level of the powerfail reset circuit is detailed. Several timing diagrams based on measurement and computer simulation also are included.

Powerfail Reset Circuit — Revision 0

Summary

The overall performance of the powerfail reset circuit (revision 0) is adequate provided that a specific set of conditions is observed. The requirements are summarized below (Table 4). Noise is also a concern. System generated noise is typically low level and can usually be neglected in portions of the circuit where the signal levels are high. Often, however, bubble systems generate significant levels of noise in a system where signal levels are low. Even low-level noise can degrade overall bubble memory system performance.

Table 4. Power Supply Requirements for Powerfail Reset Circuit (Revision 0)

	V_{CC} (volts/msec)		V_{DD} (volts/msec)	
	Min.	Max.	Min.	Max.
Power-Up Voltage Rate of Rise	0.11	None	None	None
Power-Down/Power Failure Decay Rate	None	0.70	None	.15

Noise, power fluctuations, and a rapid decay of voltage are the primary contributors to the incorrect operation of the first powerfail reset circuit (revision level 0). Since noise and power fluctuations are unavoidable in most practical systems, techniques for minimizing these effects were developed for subsequent circuits. Note that no bubble memory is immune to extremely abrupt removal of dc power. All bubble memory systems require a minimal amount of time to effect an orderly shutdown in order to maintain data integrity.

Subsequent circuit designs have been implemented to minimize system requirements by reducing the overhead required to power-down the bubble system.

The most serious fault of any powerfail reset circuit is where bubble memory data integrity is jeopardized. The first powerfail reset circuit design (revision 0) could not prevent data loss when:

1) Power was removed too rapidly for the system to ensure proper power-down.

2) Power was applied too slowly.

3) Multiple threshold crossings or "glitches" occured on the 7220-1 PWR.FAIL/ input while the coils were active.

The first two conditions can be easily prevented by following the requirements shown in Table 4. The third condition was difficult to reliably prevent and was the motivation for the revision of the circuit.

Power-up

When power initially is applied to the system (Figure 9), the PWR.FAIL/ signal is designed to be asserted by the 7230 CPG until both V_{CC} and V_{DD} reach approximately 92 percent of their nominal values. Referring to Figures 9 and 10, the 7230 internal PWR.FAIL/ output transistor cannot be guaranteed operational until V_{CC} reaches approximately 2.0 volts. During this indeterminate state of the output transistor, the floating output lags V_{CC} by approximately 0.7 volts. Therefore, the RC networks on the PWR.FAIL/ signal line (R1/C1 and R2/C2) begin charging immediately after power is applied. They continue to charge until the 7230 PWR.FAIL/ output transistor turns on. The 7230 PWR.FAIL/ output goes inactive (transistor off) when both supplies have reached the power-fail trip point. Since the RESET/ input of the 7242 FSA and the 7250 CPD are tied via the R1C1/R2C2 network to 7230 PWR.FAIL/ output, these support circuits potentially could be enabled if the 7230 PWR.FAIL/ output were allowed to rise above V_{IL} (0.8 volts). A current transient then could activate the MBM coils or bubble function conductors and cause bubbles to move to an unstable position. Note that a slow power-on ramp would be the only condition that could prematurely enable the support circuits.

Once V_{CC} reaches approximately 2.0 volts, the PWR.FAIL/ output transistor turns on to pull the PWR.FAIL/ signal low until both V_{CC} and V_{DD} reach the powerfail trip point. When the trip point is reached, the output transistor is turned-off and the PWR.FAIL/ signal is allowed to rise to the inactive level. The RC networks continue to hold the PWR.FAIL/ signal at an active level for at least 2.0 milliseconds after V_{CC} and V_{DD} have reached the trip point level. The RC delay ensures adequate time for the 7220-1 BMC's substrate bias generator to become fully operational and fully charge the 7220-1 substrate to its operational bias voltage. At some time before the PWR.FAIL/ signal reaches the 7220-1 V_{IH} (maximum) of 2.5 volts, the 7220-1 power-on initialization sequence starts. Up to this point, the 7220-1 is in an indeterminate state and the RESET.OUT/ signal, which is derived from the PWR.FAIL/ signal should be active. The behavior of the RESET.OUT/ signal, however, is similar to the 7230 PWR.FAIL/ output at low V_{CC} (below approximately 2.0 volts). As V_{CC} is slowly applied to the system, the RESET.OUT/ output transistor initially is inactive and the pullup resistor forces this output to follow 7220-1 PWR.FAIL input. Once V_{CC} reaches approximately 1.8 volts, the output transistor should turn on (RESET.OUT/ active) and remain active until completion of the power up sequence. During the inactive period, the RESET.OUT/ signal is capable of reaching the inactive level and potentially enabling the support circuits prematurely.

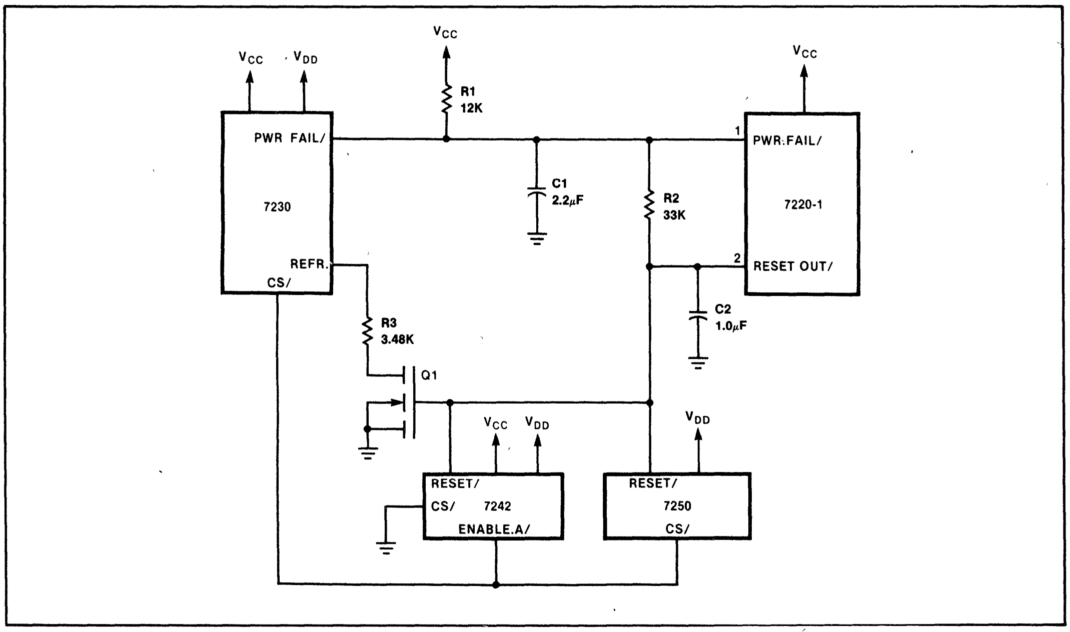
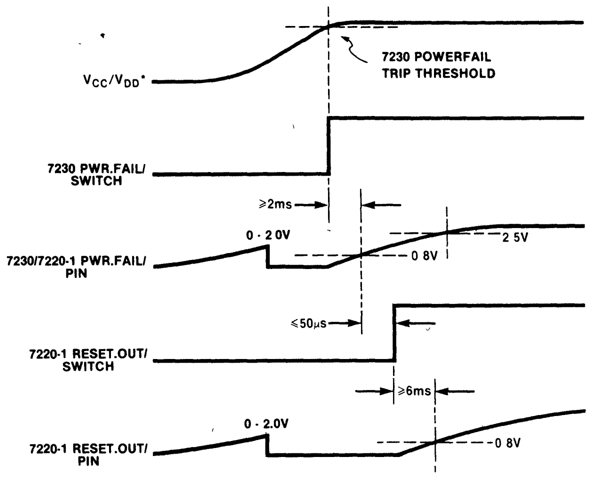
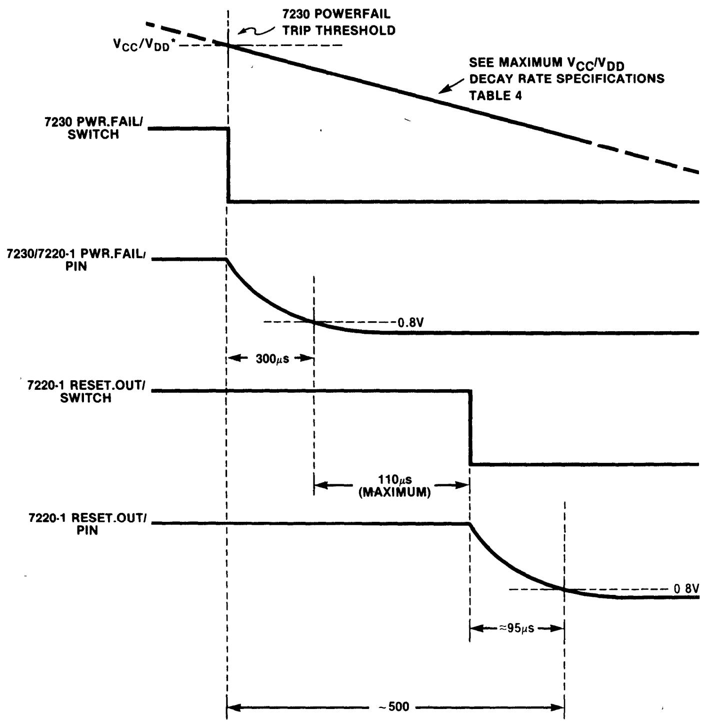


Figure 9. Revision 0 Circuit



Power-up Timing

*V_{CC} OR V_{DD}, WHICHEVER IS LAST TO CROSS POWERFAIL TRIP THRESHOLD.



Power-down Timing

*V_{CC} OR V_{DD}, WHICHEVER IS FIRST TO CROSS POWERFAIL TRIP THRESHOLD.

Figure 10. Power-up/Power-down Timing (Revision 0)

At the completion of the power-on initialization sequence, the 7220-1's internal RESET.OUT/ output transistor should be allowed to turn off. However, depending on the power-up state of certain internal 7220-1 flip-flops, this output may remain active. An Abort command is capable of internally resetting these flip-flops and releasing the RESET.OUT/ output to allow it to rise to the inactive level as determined by the R2/C2 delay network. When RESET.OUT/ reaches its inactive level, the 7242 FSA and 7250 CPD RESET/ lines are deactivated and 7230 current reference switch Q1 is turned on. The 7242 ENABLE.A/ line, which is controlled by the 7220-1, may now be activated; when active, this line enables the 7230 CS/ and 7250 CS/ (chip select) lines. The system now is fully operational and ready to execute an Initialize command (provided the Abort command had been issued).

Power-down Operation

If either V_{CC} or V_{DD} falls below the 7230 powerfail trip level, the internal PWR.FAIL/ signal in the 7230 is asserted immediately. However, due to the charge on capacitor C1 in the power-up delay network, the PWR.FAIL/ signal is prevented from reaching the active low level until C1 discharges to V_{IL} (maximum 0.8V).

When the PWR.FAIL/ signal level reaches the logic low-level threshold of the 7220-1's PWR.FAIL/ input, an internal power-down sequence is initiated within the 7220-1. As discussed earlier in the 7220-1 PWR.FAIL/ input description, the 7220-1 PWR.FAIL/ input cannot tolerate any positive threshold crossings during the power-down sequence. If a positive transition should occur; a power-up sequence will be initiated taking precedence over the power-down sequence currently in progress, and this unorderly shutdown could result in the loss of data.

The execution time of 7220-1 power-down sequence varies according to whether the coils are active (i.e., rotating magnetic field is on) or inactive. If the rotating field is off, the power down sequence is completed in approximately 10 microseconds. If the rotating field is on and a swap operation has not been initiated, the worst-case power-down time is increased to 26 microseconds; if a swap operation has been initiated, the power-down time sequence requires a maximum of 110 microseconds. The power-down time is shown in Figure 10. Note that the total system power-down time, since the operation is not complete until the RESET.OUT/ signal line is asserted, is the sum of the 7220-1's internal power-down sequence time and the discharge times for capacitors C1 and C2. To ensure proper operation of the bubble system for data integrity during power-down operations, the power supply maximum decay rates must be observed.

Powerfail Reset Circuit — Revision 1

Summary

The powerfail reset circuit (revision 1) was designed to reduce the requirements placed on the revision 0 powerfail reset circuit and to further reduce the risk of data loss during power-up/down operation. Specifically, the improvements realized were:

1. The possibility of data loss was eliminated provided that the circuit was operated within voltage decay rates specifications.
2. Power-down time was shortened to reduce the energy storage requirements.

The power supply requirements (shown in Table 5) were relaxed with this implementation, which reduces the system requirements and the possibility of data loss.

Power-up

The power-up operation of the circuit shown in Figure 11 is unchanged from the power-up operation of the revision 0 circuit. The characteristics associated with the operation of the powerfail reset circuit below approximately 2.0 volts were not resolved with this circuit solution. If the voltage rise time specifications were not observed, the support circuits could have been enabled prematurely and would allow current transients to reach the drive coils or bubble function conductors (resulting in data loss).

Table 5. Power Supply Requirements for Powerfail Reset Circuit (Revision 1)

	V_{CC} (volts/msec)		V_{DD} (volts/msec)	
	Min.	Max.	Min.	Max.
Power-Up Voltage Rate of Rise	0.12	None	None	None
Power-Down/Power Failure Decay Rate	None	0.45	None	1.1

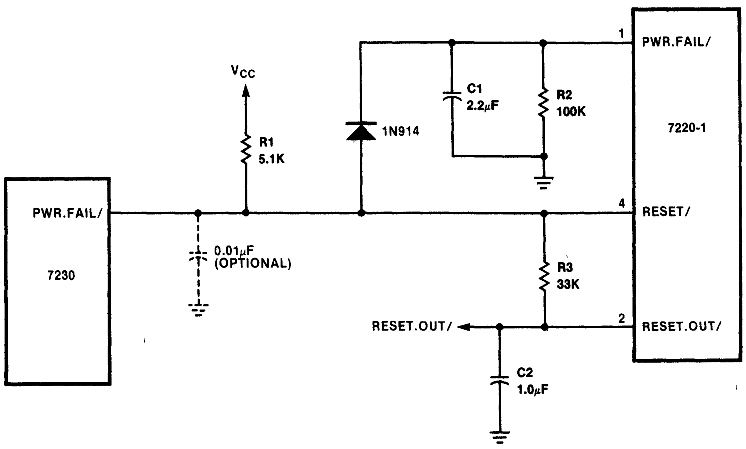


Figure 11. Revision 1 Circuit

Power-down

The simple modifications implemented in the external powerfail circuit (revision 1) greatly reduced the overall power-down operation timing (See Figure 12). This modification made use of the 7220-1 RESET/ input to initiate a power-down sequence instead of the 7220-1 PWR.FAIL/ input by effectively isolating the 7230 PWR.FAIL/ signal from delay capacitor C1 during power-down operations (eliminating an initial capacitor discharge delay). The 7220-1 BMC initiates an internal power-down sequence whenever its RESET/ input goes active, identical to the negative transition of the 7220-1 PWR.FAIL/ input. The difference between these two 7220-1 input signals is that the RESET/ input is latched and does not recognize a low-to-high transition and power-up therefore must be initiated by the positive transition of the 7220-1 PWR.FAIL/ input. With this circuit, the power-up operation timing was unaltered, and the power-down operation timing was reduced from approximately 500 microseconds in the revision 0 powerfail circuit to approximately 200 microseconds in the revision 1 powerfail circuit.

The primary reason for further refining this approach was the increased possibility for a "communication lockout" by the 7220-1. "Communication lockout" resulted when power was temporarily lost from the system. Specifically, the following two conditions were responsible for the "communication lockout":

1) The 7220-1 RESET/ input was activated low due to power loss (minimum pulse width must be 250 nanoseconds to ensure that it is latched) and initiated a power-down sequence.

2) The 7220-1 PWR.FAIL/ discharged but not below the inactive state (0.8 to 2.5 volts, typically 1.5 volts), before power was restored. A power-up sequence could not be initiated to reset the BMC to a known state and communication is "locked out."

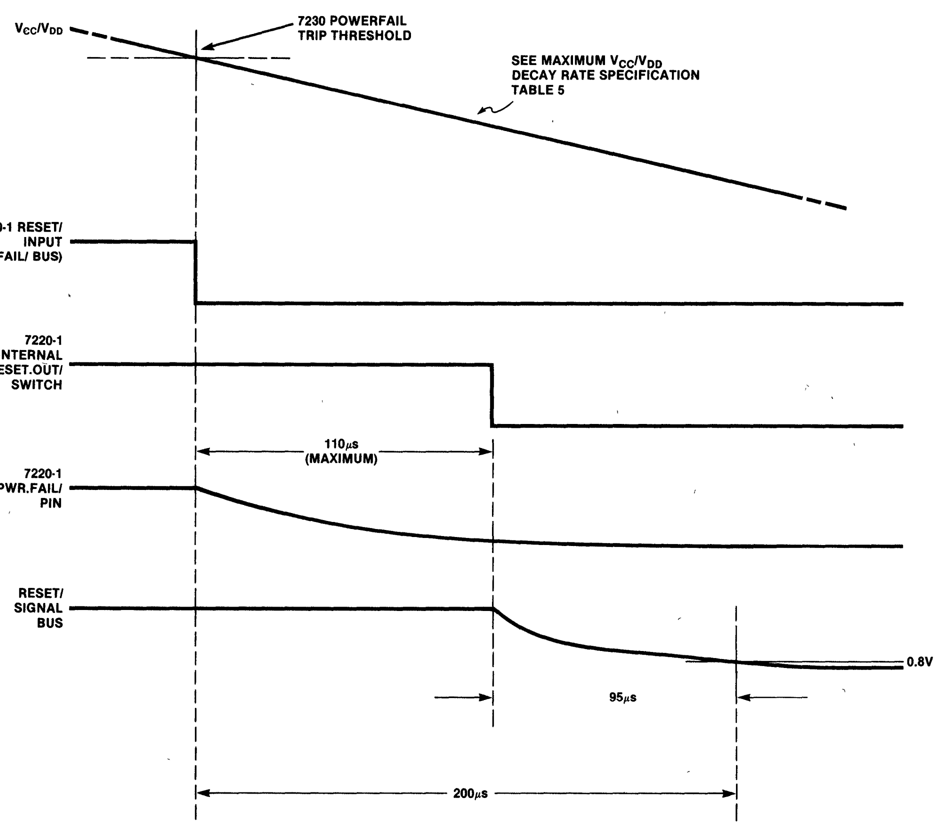


Figure 12. Power-down Timing (Revision 1)

Even if the circuit is operated within the voltage decay rate specifications, this inconvenience is still possible; the only solution is to pulse the 7220-1 PWR.FAIL/ input long enough to discharge C1 to a worst case value of 0.8 volt either by power cycling or external control. This user inconvenience and special system requirement led to the development of the next powerfail reset circuit.

Powerfail Reset Circuit — Revision 2

The powerfail reset circuit (revision 2) was developed to eliminate the possibility of data loss during power-up and power-down operation provided the power supply requirements are observed. The following paragraphs describe the principals of operation of the powerfail reset circuit. As power is applied or removed, several different signal value combinations are possible which complicate the analysis of this circuit. For the sake of simplicity, a general overview of a typical case is included rather than a detailed representation of each case. Throughout this discussion it is helpful for the reader to refer to the schematic diagram (Figure 3) and the timing diagrams (Figure 5 and 6).

Power-up

The overall circuit operation is complicated by the additional component, IC1. The power-up operation of the revision 2 circuit is very similar to previous circuits, however, the possibility of prematurely enabling the support components is eliminated. Diodes D1, D2 and resistor R5 serve to prevent capacitor C2 from charging beyond a level (0.8V) that could potentially deactivate the RESET/ signal bus to the 7242 FSA, the 7250 CPD and the VMOS transistor switch. Resistor R5 is chosen so that as V_{CC} is applied, diodes D1 and D2 will be forward biased and provide sufficient voltage drop to prevent capacitor C2 from charging above 0.8V.

Once the 7220-1 power-up sequence is complete or the first Abort command is received, the 7220-1 RESET.OUT/ is deactivated and capacitor C2 is allowed to fully charge. When the RESET/ signal bus reaches an inactive state the power-up sequence is complete and the system is prepared to accept an Initialize command (provided the Abort command has been issued).

Power-down

The power-down operation of the external powerfail reset circuit (revision 2) is very similar to revision 1. The fundamental difference is the ability to maintain a charge on capacitor C1 throughout the 7220-1 power-down sequence. This eliminates any glitch sensitivity or incorrect circuit operation during momentary power loss. The 7220-1 BMC initiates an internal power-down sequence whenever its RESET/ input goes active. The 7220-1 RESET.OUT/ signal is gated through IC1 and remains inactive during this time period preventing capacitor C1 from discharging. At the completion of the 7220-1 power-down sequence RESET.OUT/ signal is pulled low which causes both of the IC1 OR gate outputs to go low. The current sinking capability of these outputs act to quickly discharge capacitors C1 and C2 and complete the power-down sequence.

intel®

APPLICATION NOTE

AP-150

December 1982

8085 To BPK 72 Interface

Ulmont Smith Jr.
Applications Engineer

DECEMBER 1982
ORDER NUMBER: 210849-001

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. After the kit is assembled, the designer is left 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 one megabit Bubble Memory Module, a 10cm x 10cm printed circuit board (IMB-72), 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.

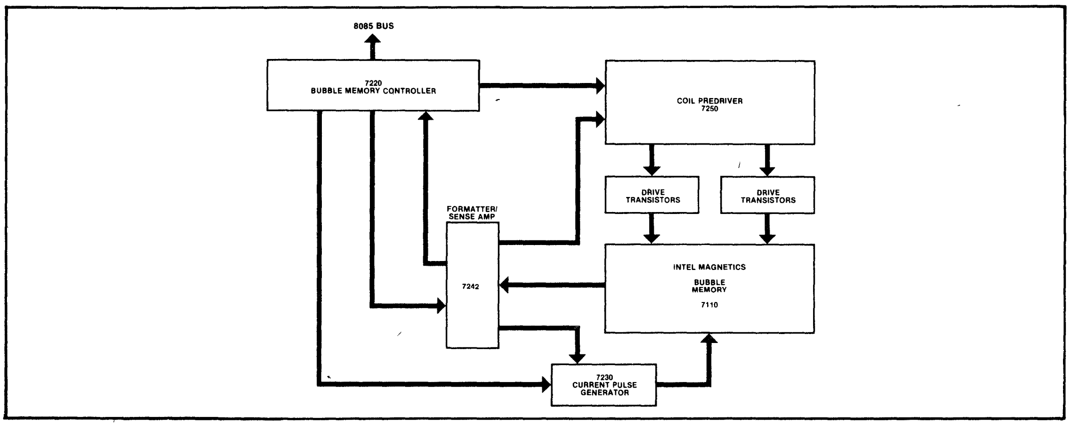


Figure 1. Block Diagram of the BPK 72

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

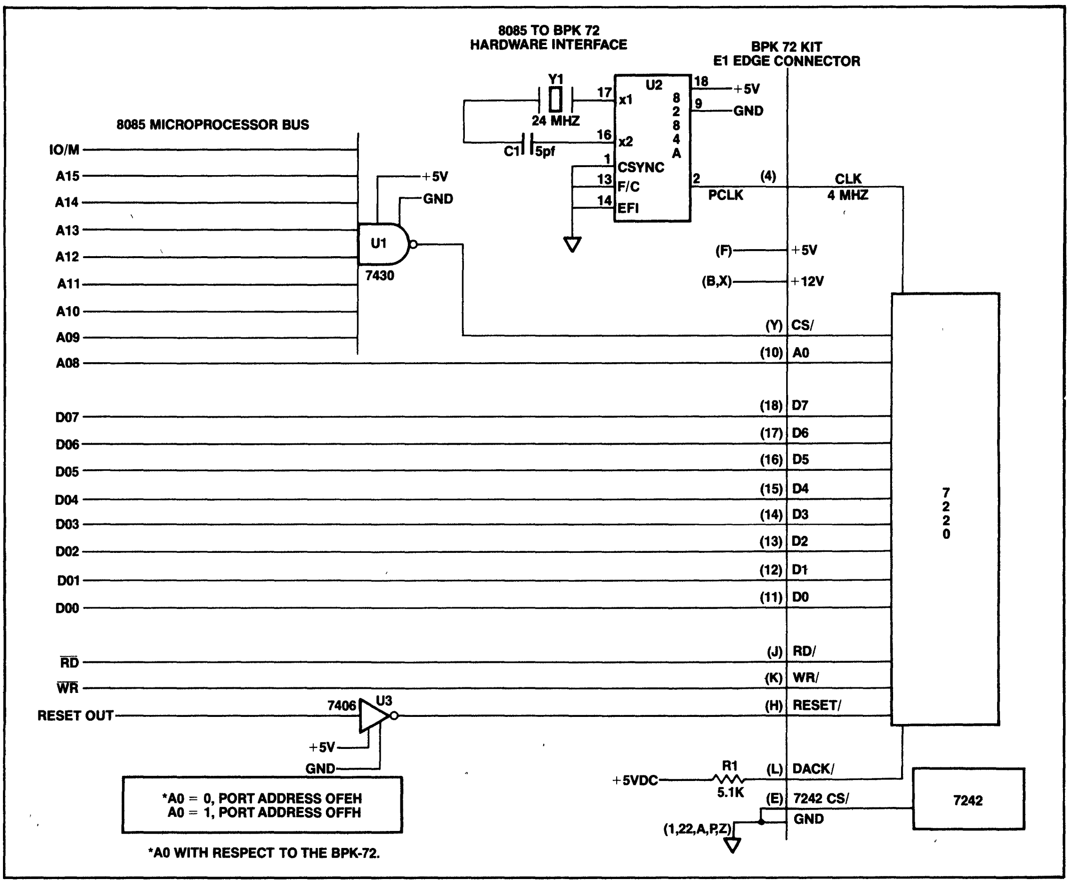


Figure 2. Hardware Interface

The interface designer should verify that the system power supply decay rates meet 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. Do not use the IMB-72 board with the 7110 dummy module during the power supply decay measurements. The dummy module is not capable of fully loading the power supplies. No attempt should be made to use the IMB-72 board with the 7110 Bubble Memory until the power supply decay rates have been verified.

Note: The procedure outlined in Appendix B, "Powering-Up for the First Time," should be performed prior to installing the 7110 Bubble Memory Module in a newly assembled IMB-72 board.

All BPK 72 kits Rev G or earlier contain an older version of the power fail circuit. The revision letter can be found on the solder side of the IMB-72 printed circuit board. The old version performs the power fail protection function, but has limited immunity to power supply noise. An IMB-72 printed circuit board containing the new power fail circuit is available from Intel Magnetics. A new board, additional hardware, and documentation are available at no cost to customers with older BPK 72 kits containing the original power fail circuit. Customers are urged to utilize the improved power fail circuit in all future designs.

Intel Magnetics Applications
3065 Bowers Avenue SC2-962
Santa Clara, CA 95051
(408) 987-7602

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:

8085 Microprocessor		**8085 Addressable Memory**
B Reg = 10H	C Reg = 00H →	1000H = 01H Block Length Reg LSB
		1001H = 10H Block Length Reg MSB
D Reg = XXH	E Reg = XXH	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.		XX—Don't care No effect on the operation of the BPK 72.
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 C 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

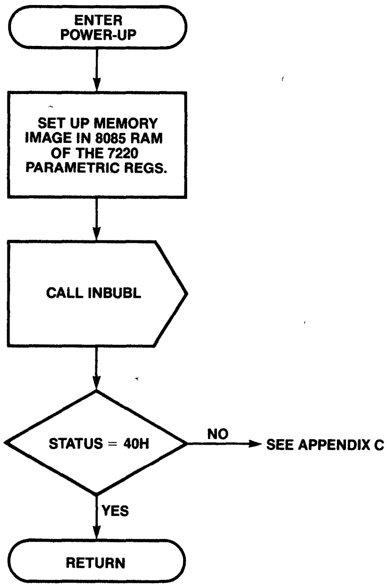


Figure 3. Initializing the BPK 72

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 D Reg/E Reg = 00H line points to 2000H.)

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.

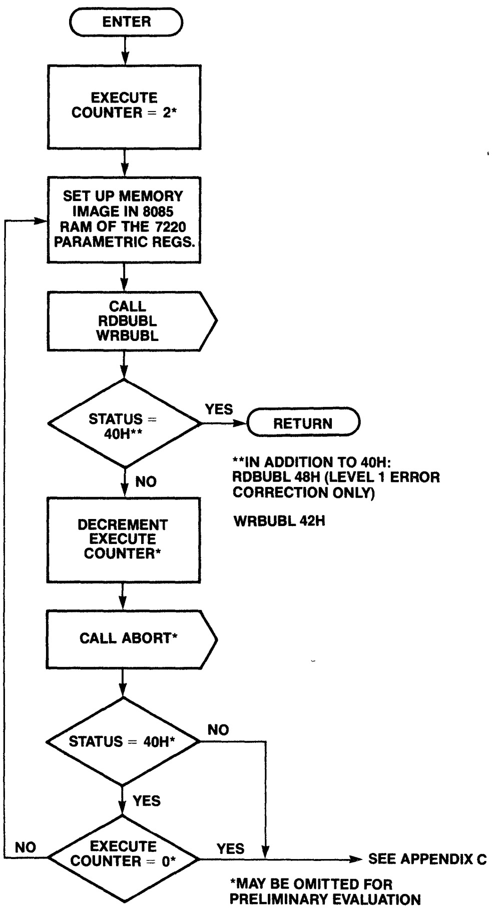


Figure 4. Reading and Writing to the BPK 72

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 42H	OP-complete OP-complete, parity error
RDBUBL	40H 48H	OP-complete 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 C 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

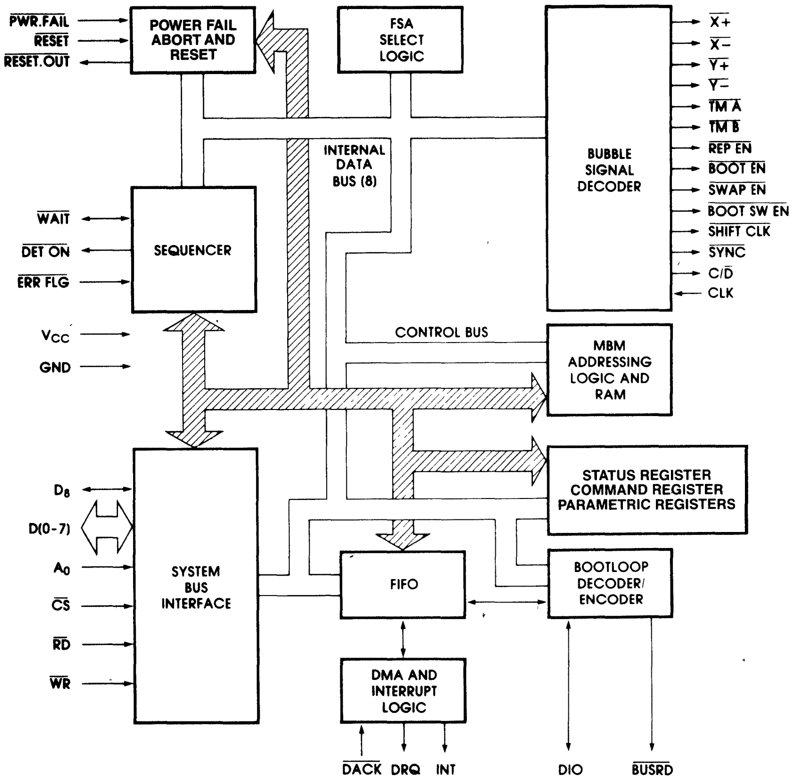


Figure 5. Block Diagram of the 7220 Bubble Memory Controller

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
** Necessary to verify successful assembly of the BPK 72 (see Appendix B)
*** Diagnostic routines (see Appendix C)

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				Symbol	Name of Register	Read/Write
A0	B3	B2	B1	B0			
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.

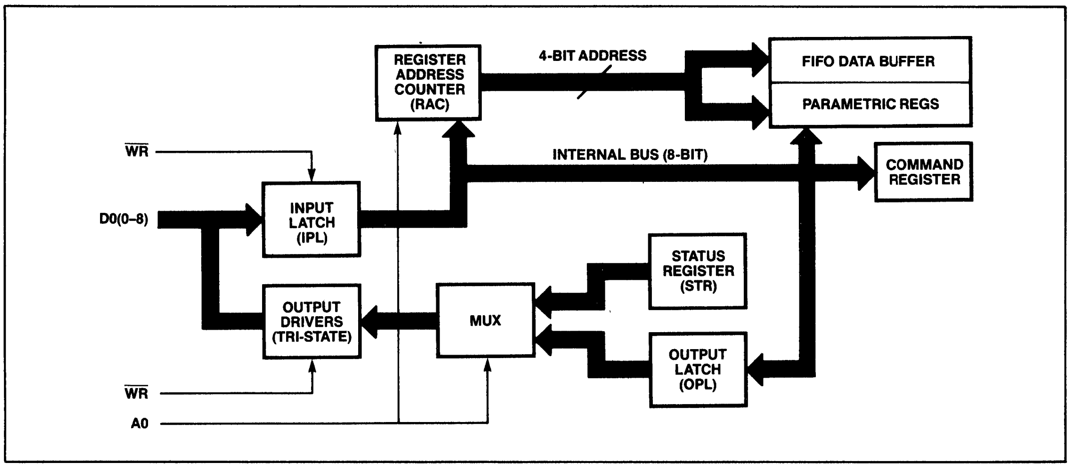


Figure 6. 7220 User Accessible Registers

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:

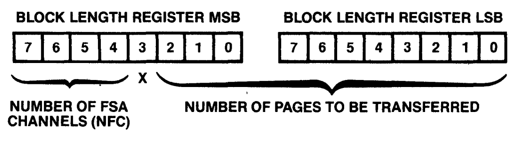


Figure 7. Block Length Registers

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:

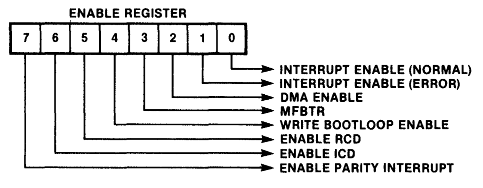


Figure 8. Enable Register

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:

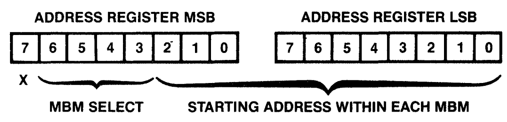


Figure 9. Address Registers

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:

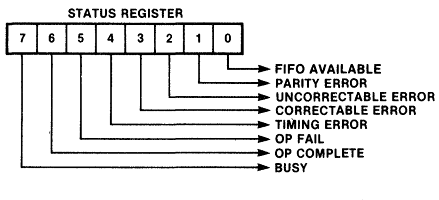


Figure 10. Status Register

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.

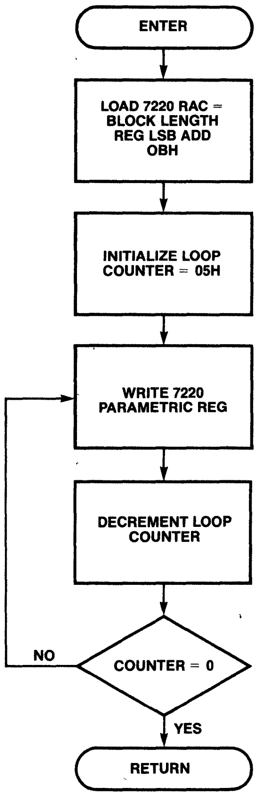
APPENDIX A
8085 TO BPK-72 INTERFACE
SOFTWARE DRIVER LISTING
AND
FLOWCHARTS

```
ASM80 :F1:BPKHDR


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

 LOC  OBJ         LINE        SOURCE STATEMENT

;**************************************************************************************
;
;
;                      PROGRAM: 8085 TO BPK72 SOFTWARE DRIVER V1.0
;                      ULMONT S. SMITH JR.
;                      INTEL CORPORATION
;                      3065 BOWERS AVENUE
;                      SANTA CLARA, CALIFORNIA 95051
;
;
;**************************************************************************************
;
;
;
; ABSTRACT
;
;       THIS PROGRAM  CONSISTS OF A SET OF  BUBBLE MEMORY SOFTWARE DRIVERS
;       THAT SUPPORT A POLLED MODE INTERFACE BETWEEN A BPK72, 1MBIT BUBBLE
;       MEMORY PROTOTYPE KIT,  AND  A STANDARD  8085  MICROPROCESSOR.  THE
;       PROGRAM  UTILIZES A SET  OF PUBLIC  DIRECTIVES THAT  CAN BE CALLED
;       TO PERFORM A BUBBLE MEMORY INITIALIZATION, READ, WRITE,  AND OTHER
;       COMMONLY USED COMMANDS.  IN THE UNLIKELY EVENT THAT THE 7110 BUBBLE
;       MEMORY BOOT LOOP IS LOST, TWO ROUTINES ARE PROVIDED TO EXAMINE AND
;       REWRITE THE BOOT LOOP CODE.
;
;
;
; PROGRAM ORGANIZATION.
;
;       FUNCTIONS:
;                      INTPAR
;                      FIFOPS
;                      BYTCNT
;                      WRITE
;                      READ
;                      ABORT
;                      WRBUBL
;                      RDBUBL
;                      INBUBL
;                      BOOTUP
;                      RDBOOT
;                      WRFIFO
;                      RDFIFO
;                      WPBLRS
;                      RDBLRS
;                      MBMPRG
;
;
; EXTERNAL DECLARATIONS: NONE
;
;
$EJECT
```

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

 LOC  OBJ        LINE       SOURCE STATEMENT

                      ;
                      ; PUBLIC SYMBOLS
                      ;
                      ;                FIFORS - RESET 7220 FIFO DATA BUFFER
                      ;                ABORT  - ABORT PRESENT COMMAND, RESET BPK72
                      ;                WRBUBL - WRITE BUBBLE MEMORY DATA
                      ;                RDBUBL - READ BUBBLE MEMORY DATA
                      ;                INBUBL - INITIALIZE THE BPK72
                      ;                BOOTUP - WRITE BUBBLE MEMORY BOOT LOOP
                      ;                RDBOOT - READ BUBBLE MEMORY BOOT LOOP
                      ;                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
                      ;
                      ;
                      ;*********************************************************************************
                      ;
                                               NAME BPK72
                      ;
                      ;*********************************************************************************
                      ;
                      ;
                      ;*********************************************************************************
                      ;
 0800                                          ORG     0800H
                      ;
                      ;*********************************************************************************
                      ;
                      ;
                      ;*********************************************************************************
                      ;
                      ;                        PROGRAM EQUATES
                      ;
                      ;*********************************************************************************
                      ;
                      ;
 00FE                 PRTA00  EQU     0FEH    ; A POLLED  MODE  INTERFACE  REQUIRES  ONLY  TWO  I/O
 00FF                 PRTA01  EQU     0FFH    ; PORTS DESIGNATED BY THE A0 LINE ON THE BPK72 BOARD.
                                              ; THIS APPLICATION USES:
                                              ;
                                              ;       0FEH - A0=0 FOR PRTA00 (PORT A0= 0)
                                              ;              RD/WR BUBBLE MEMORY DATA AND REGS
                                              ;
                                              ;       0FFH - A0=1 FOR PRTA01 (PORT A0= 1)
                                              ;              RD STATUS REG
                                              ;              WR BUBBLE MEMORY COMMANDS
                      ;
                      $EJECT
```

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

```
 LOC  OBJ         LINE        SOURCE STATEMENT

                        ;*******************************************************************************
                        ;
                        ; FUNCTION: INTPAR
                        ; INPUTS:   B-C REGS, STARTING ADDRESS OF PARAMETRIC REGS IN RAM
                        ; OUTPUTS:  7220 PARAMETRIC REGS
                        ; CALLS:    NONE
                        ; DESTROYS: A, F/FS
                        ;
                        ; DESCRIPTION: LOAD THE 7220 PARAMETRIC REGS
                        ;        THE B-C REGS  CONTAIN THE ADDRESS TO THE FIRST OF FIVE CONTIGUOUS
                        ;        MEMORY LOCATIONS IN RAM.  THE DATA  ADDRESSED BY THE  B-C REGS IS
                        ;        USED TO LOAD  THE PARAMETRIC  REGISTERS IN THE 7220 BUBBLE MEMORY
                        ;        CONTROLLER. INTPAR COPIES THE DATA IN RAM TO THE PARAMETRIC REGS.
                        ;
 0800 C5                INTPAR: PUSH    B        ; SAVE B-C REGS
 0801 D5                        PUSH    D        ; SAVE D-E REGS
 0802 3E0B                      MVI     A,0BH    ; LOAD A REG WITH BLR LSB ADDRESS
 0804 D3FF                      OUT     PRTA01   ; LOAD 7220 RAC WITH BLR LSB ADDRESS
 0806 1E05                      MVI     E,05H    ; INITIALIZE LOOP COUNTER
 0808 0A                LOAD:   LDAX    B        ; LOAD A REG FROM B-C REG ADDRESS
 0809 D3FE                      OUT     PRTA00   ; WRITE PARAMETRIC REG
 080B 03                        INX     B        ; INCREMENT B-C REGS TO THE NEXT ADDRESS IN RAM
 080C 1D                        DCR     E        ; DECREMENT LOOP COUNTER
 080D C20808                    JNZ     LOAD     ; IF NOT ZERO, JMP LOAD
 0810 D1                        POP     D        ; RESTORE D-E REGS
 0811 C1                        POP     B        ; RESTORE B-C REGS
 0812 C9                        RET              ; RETURN TO CALL
                        ;
                        ;
                        ;
                        $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

                    ;*****************************************************************************
                    ;
                    ; FUNCTION: FIFORS
                    ; INPUTS.   BPK72 STATUS REG
                    ; OUTPUTS:  ISSUE FIFO RESET COMMAND TO BPK72
                    ;           A REG= BPK72 STATUS REG
                    ; CALLS.    NONE
                    ; DESTROYS. A, F/FS
                    ;
                    ; DESCRIPTION. RESET 7220 FIFO DATA BUFFER
                    ;       A FIFO RESET COMMAND IS ISSUED TO THE BPK72. AFTER ISSUING THE
                    ;       COMMAND, THE BPK72 STATUS REG IS POLLED UNTIL AN OP-COMPLETE,
                    ;       40H, HAS BEEN READ OR THE TIME OUT LOOP COUNTER DECREMENTS TO
                    ;       ZERO. FIFORS RETURNS THE VALUE OF THE BPK72 STATUS REG TO  THE
                    ;       CALLING ROUTINE  VIA THE 8085 S A REG.  ONLY  A  STATUS OF 40H
                    ;       INDICATES A SUCCESSFUL EXECUTION OF THE FUNCTION FIFORS.
                    ;
                            PUBLIC  FIFORS  ; DECLARE PUBLIC FUNCTION
 0813 D5            FIFORS: PUSH    D       ; SAVE D-E REGS
 0814 C5                    PUSH    B       ; SAVE B-C REGS
 0815 0640                  MVI     B,40H   ; LOAD B REG= 40H, OP-COMPLETE
 0817 11FFFF                LXI     D,0FFFFH; INTIALIZE TIME OUT LOOP COUNTER
 081A 3E1D                  MVI     A,1DH   ; LOAD A REG= FIFO RESET COMMAND
 081C D3FF                  OUT     PRTA01  ; WRITE FIFO RESET COMMAND
 081E DBFF          BUSYFR  IN      PRTA01  ; READ STATUS REG
 0820 07                    RLC             ; TEST BUSY BIT= 1
 0821 DA2E08                JC      POLLFR  ; IF BUSY= 1, POLL STATUS REG FOR 40H
 0824 1B                    DCX     D       ; DECREMENT TIME OUT LOOP COUNTER
 0825 AF                    XRA     A       ; CLEAR A REG
 0826 B2                    ORA     D       ; TEST D REG= 00H
 0827 B3                    ORA     E       ; TEST E REG= 00H
 0828 C21E08                JNZ     BUSYFR  ; IF NOT ZERO, CONTINUE POLLING FIFO RESET COMMAND
 082B C33B08                JMP     RETFR   ; TIME OUT ERROR, RETURN
 082E DBFF          POLLFR  IN      PRTA01  ; READ STATUS REG
 0830 A8                    XRA     B       ; TEST STATUS= 40H, OP-COMPLETE
 0831 CA3B08                JZ      RETFR   ; IF OP-COMPLETE, JMP RETFR
 0834 1B                    DCX     D       ; DECREMENT TIME OUT LOOP COUNTER
 0835 AF                    XRA     A       ; CLEAR A REG
 0836 B2                    ORA     D       ; TEST D REG= 00H
 0837 B3                    ORA     E       ; TEST E REG= 00H
 0838 C22E08                JNZ     POLLFR  ; IF NOT ZERO, CONTINUE POLLING FIFO RESET COMMAND
 083B C1            RETFR   POP     B       ; RESTORE B-C REGS
 083C D1                    POP     D       ; RESTORE D-E REGS
 083D DBFF                  IN      PRTA01  ; READ STATUS REG
 083F C9                    RET             ; RETURN TO CALL
                    ;
                    ;
                    ;
                    $EJECT
```

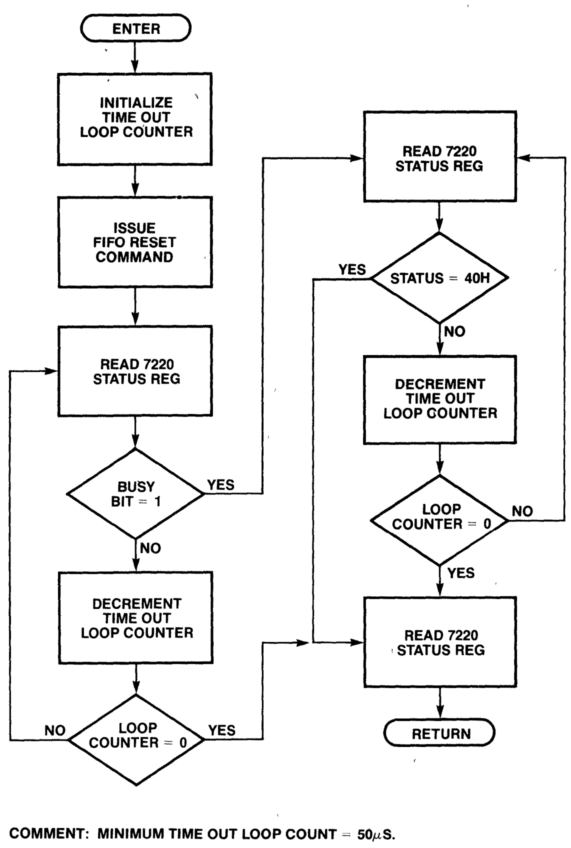
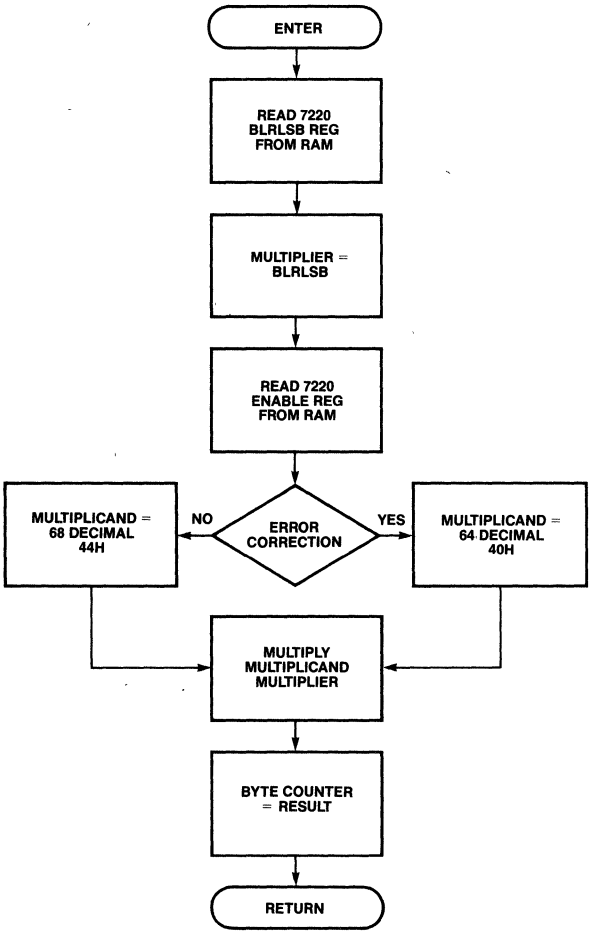


Figure 12. FIFORS

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


 LOC  OBJ              SOURCE STATEMENT

                ;*****************************************************************************
                ; FUNCTION: BYTCNT
                ; INPUTS:   B-C REGS, STARTING ADDRESS OF PARAMETRIC REGS IN RAM
                ; OUTPUTS:  H-L REGS= BYTE COUNTER
                ; CALLS     NONE
                ; DESTROYS: A, H, L, F/FS
                ;
                ; DESCRIPTION: BYTE COUNTER
                ;         THE B-C REGS CONTAIN THE ADDRESS TO THE FIRST OF FIVE CONTIGUOUS MEMORY
                ;         LOCATIONS IN RAM.  THE DATA  ADDRESSED BY  THE B-C REGS IS USED TO LOAD
                ;         THE PARAMETRIC REGS IN  THE 7220  BUBBLE MEMORY CONTROLLER.  THE ENABLE
                ;         REG IS READ FROM RAM TO DETERMINE IF ERROR CORRECTION HAS BEEN ENABLED.
                ;         THE USE OF ERROR CORRECTION REQUIRES A 64 BYTE TRANSFER/PAGE  - 68 BYTE
                ;         TRANSFER/PAGE WITHOUT ERROR CORRECTION.  THE  BLOCK  LENGTH  REG LSB IS
                ;         ALSO READ FROM RAM TO  DETERMINE THE NUMBER  OF PAGES TO BE TRANSFERRED
                ;         DURING THE NEXT READ OR WRITE COMMAND.  THE  NUMBER  OF  BYTES PER PAGE
                ;         MULTIPLIED BY THE NUMBER OF PAGES IS COMPUTED AND PASSED TO THE CALLING
                ;         ROUTINE VIA THE 8085'S H-L REGS.  DATA TRANSFERS ARE LIMITED TO  16,320
                ;         BYTES WITH ERROR CORRECTION AND  17,340 BYTES WITHOUT   ONLY THE BLRLSB
                ;         IS USED TO GENERATE THE BYTE COUNTER
                ;
0840 C5         BYTCNT: PUSH  B       ; SAVE B-C REGS
0841 D5                 PUSH  D       ; SAVE D-E REGS
0842 0A                 LDAX  B       ; LOAD A REG WITH BLRLSB
0843 6F                 MOV   L,A     ; MOVE BLRLSB TO L REG
0844 03                 INX   B       ;
0845 03                 INX   B       ; INCREMENT B-C REGS TO ADDRESS THE ENABLE REG IN RAM
0846 0A                 LDAX  B       ; LOAD A REG WITH ENABLE REG
0847 67                 MOV   H,A     ; MOVE ENABLE REG TO H REG
0848 1640               MVI   D,40H   ; INITIALIZE D REG 64 BYTES/PAGE XFER, 40H
084A 3E60               MVI   A,60H   ; ERROR CORRECTION DETECTION MASK
084C A4                 ANA   H       ; LOGICAL AND MASK WITH H REG, TEST FOR ERROR CORRECTION
084D C25208             JNZ   MULT    ; IF ZERO, ERROR CORRECTION IS NOT ENABLED
0850 1644               MVI   D,44H   ; NO ERROR CORRECTION, 68 BYTES/PAGE XFER, 44H
                                      ; MULTIPLY (D REG) X (L REG)
                                      ; 64 OR 68 BYTES X NO  OF PAGES IN BLRLSB
                                      ; RESULT WILL BE PLACED IN THE H-L REGS
                                      ; BEGIN MULTIPLY ROUTINE
0852 2600       MULT:   MVI   H,0H    ; INITIALIZE MOST SIGNIFICANT BYTE OF RESULT
0854 1E09               MVI   E,09H   ; INITIALIZE BIT COUNTER
0856 7D         MULT0:  MOV   A,L     ; MOVE LOW ORDER BYTE INTO A REG
0857 1F                 RAR           ; ROTATE LEAST SIGNIFICANT BIT OF MULTIPLIER
0858 6F                 MOV   L,A     ; MOVE LOW ORDER BYTE OF RESULT INTO L REG
0859 1D                 DCR   E       ; DECREMENT BIT COUNTER
085A CA6708             JZ    DONE    ; EXIT IF COMPLETE
085D 7C                 MOV   A,H     ; MOVE HIGH ORDER BYTE INTO A REG
085E D26208             JNC   MULT1   ; IF CARRY= 0, JMP MULT1
0861 82                 ADD   D       ; ADD D REG TO A REG
0862 1F         MULT1:  RAR           ; CARRY= 0, SHIFT HIGH ORDER BYTE OF RESULT
0863 67                 MOV   H,A     ; MOVE HIGH ORDER RESULT INTO H REG
0864 C35608             JMP   MULT0   ; CONTINUE LOOPING
0867 D1         DONE    POP   D       ; RESTORE D-E REGS
0868 C1                 POP   B       ; RESTORE B-C REGS
0869 C9                 RET           ; RETURN TO CALL
                ;
```



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

                       ;*********************************************************************************
                       ;
                       ; FUNCTION: WRITE
                       ; INPUTS:    D-E REGS, STARTING ADDRESS OF DATA IN RAM
                       ,            H-L REGS, BYTE COUNTER
                       ;            BPK72 STATUS REG
                       ; OUTPUTS:   WRITE DATA TO BUBBLE MEMORY
                       ; CALLS:     NONE
                       , DESTROYS: A, H, L, F/FS
                       ,
                       ; DESCRIPTION: TRANSFER DATA FROM RAM TO BUBBLE MEMORY
                       ;         THE D-E REGS CONTAIN THE STARTING ADDRESS IN RAM OF DATA
                       ;         TO BE WRITTEN INTO THE BUBBLE MEMORY. THE H-L REGS MUST
                       ;         CONTAIN A BYTE COUNTER INDICATING THE NUMBER OF DATA BYTES
                       ,         TO BE TRANSFERRED. THIS FUNCTION BEGINS BY ISSUING THE WRITE
                       ;         BUBBLE MEMORY DATA COMMAND FOLLOWED BY POLLING THE STATUS REG
                       .         TO DETERMINE IF THE 7220 FIFO DATA BUFFER IS READY TO RECEIVE
                       ,         DATA. DATA IS TRANSFERRED UNTIL THE BYTE COUNTER OR TIME
                       ;         OUT LOOP COUNTER DECREMENTS TO ZERO. THE PARAMETRIC REGISTERS
                       ;         MUST BE LOADED WITH THE DESIRED VALUES PRIOR TO CALLING THIS
                       ;         FUNCTION.
                       ;
086A D5                WRITE:  PUSH   D        , SAVE D-E REGS
086B C5                        PUSH   B        ; SAVE B-C REGS
086C 01FFFF                    LXI    B,0FFFFH; INITIALIZE TIME OUT LOOP COUNTER
086F 3E13                      MVI    A,13H    ; LOAD A REG= WRITE BUBBLE MEMORY DATA COMMAND
0871 D3FF                      OUT    PRTA01   ; WRITE, WRITE BUBBLE MEMORY DATA COMMAND
0873 0B                BUSYWR. DCX    B        ; DECREMENT TIME OUT LOOP COUNTER
0874 AF                        XRA    A        , CLEAR A REG
0875 B0                        ORA    B        , TEST B REG= 00H
0876 B1                        ORA    C        ; TEST C REG= 00H
0877 CAA108                    JZ     FINSHW   . IF ZERO, TIME OUT ERROR, JMP FINSHW
087A DBFF                      IN     PRTA01   , READ STATUS REG
087C 07                        RLC             , TEST BUSY BIT= 1
087D D27308                    JNC    BUSYWR   ; IF ZERO, CONTINUE POLLING BUSY BIT
                                               , CONTINUED ON NEXT PAGE
                       $EJECT
```

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

 LOC  OBJ         LINE        SOURCE STATEMENT

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

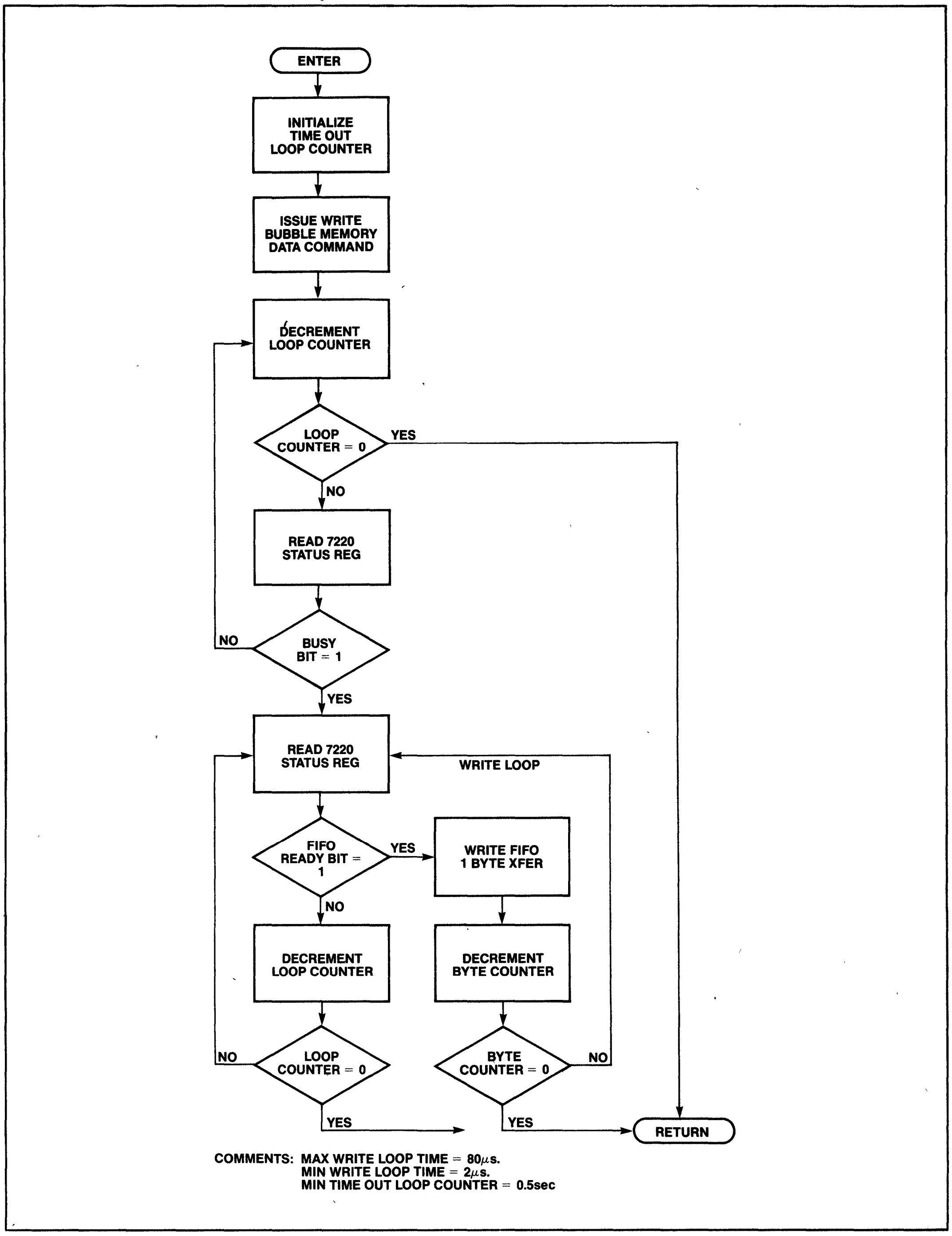
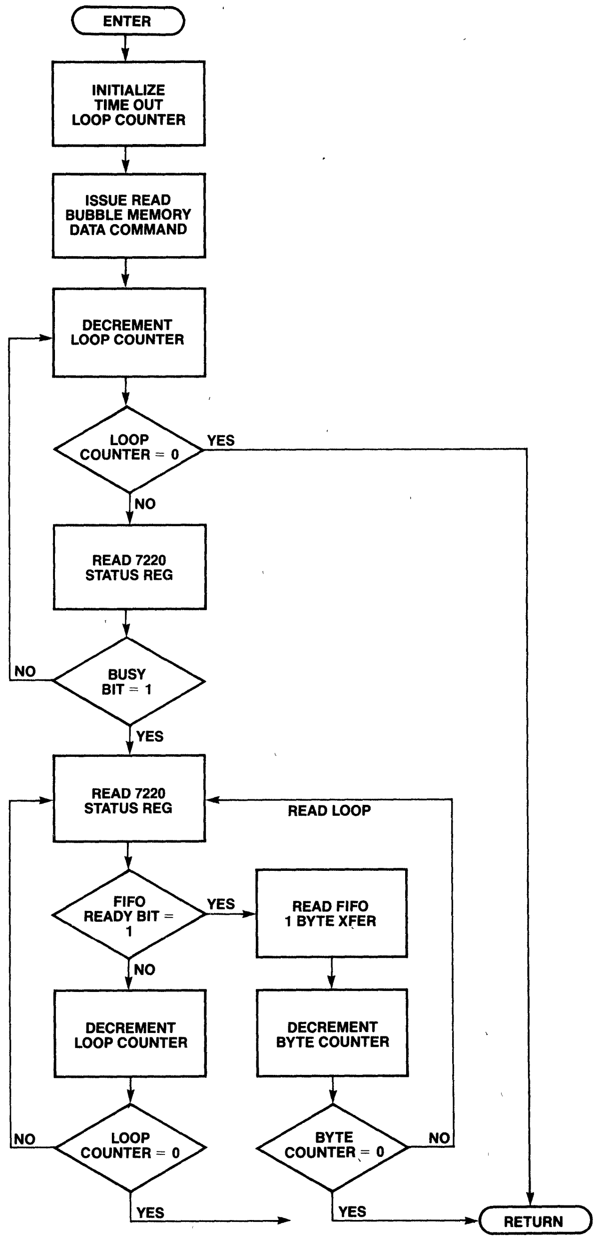


Figure 14. WRITE

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

 LOC  OBJ         LINE        SOURCE STATEMENT

                      ;*****************************************************************************
                      ;
                      ; FUNCTION: READ
                      ; INPUTS    D-E REGS, STARTING ADDRESS IN RAM
                      ;           H-L REGS, BYTE COUNTER
                      ;           BPK72 STATUS REG
                      ;           READ DATA FROM BUBBLE MEMORY
                      ; OUTPUTS   WRITE DATA TO RAM
                      ; CALLS.    NONE
                      ; DESTROYS. A, H, L, F/FS
                      ;
                      ; DESCRIPTION. TRANSFER DATA FROM BUBBLE MEMORY TO RAM
                      ;       THE D-E REGS CONTAIN THE STARTING ADDRESS IN RAM USED TO STORE
                      ;       DATA READ FROM THE BUBBLE MEMORY.  THE H-L REGS MUST CONTAIN
                      ;       A BYTE COUNTER INDICATING THE NUMBER OF DATA BYTES TO BE
                      ;       TRANSFERRED  THIS FUNCTION BEGINS BY ISSUING THE READ BUBBLE
                      ;       MEMORY DATA COMMAND FOLLOWED BY POLLING THE STATUS REG
                      ;       TO DETERMINE IF THE 7220 FIFO DATA BUFFER CONTAINS DATA
                      ;       AVAILABLE FOR READING  DATA IS TRANSFERRED UNTIL THE BYTE
                      ;       COUNTER OR TIME OUT LOOP COUNTER DECREMENTS TO ZERO.  THE
                      ;       PARAMETRIC REGS MUST BE LOADED WITH THE DESIRED VALUES PRIOR
                      ;       TO CALLING THIS FUNCTION
                      ;
08A4 D5               READ.   PUSH  D        ; SAVE D-E REGS
08A5 C5                       PUSH  B        ; SAVE B-C REGS
08A6 01FFFF                   LXI   B,0FFFFH; INITIALIZE TIME OUT LOOP COUNTER
08A9 3E12                     MVI   A,12H    ; LOAD A REG= READ BUBBLE MEMORY DATA COMMAND
08AB D3FF                     OUT   PRTA01   ; WRITE, READ BUBBLE MEMORY DATA COMMAND
08AD 0B               BUSYRD: DCX   B        ; DECREMENT TIME OUT LOOP COUNTER
08AE AF                       XRA   A        ; CLEAR A REG
08AF B0                       ORA   B        ; TEST B REG= 00H
08B0 B1                       ORA   C        ; TEST C REG= 00H
08B1 CADB08                   JZ    FINSHR   ; IF ZERO, TIME OUT ERROR, JMP FINSHR
08B4 DBFF                     IN    PRTA01   ; READ STATUS REG
08B6 07                       RLC            ; TEST BUSY BIT= 1
08B7 D2AD08                   JNC   BUSYRD   ; IF ZERO, CONTINUE POLLING BUSY BIT
                                             ; CONTINUED ON NEXT PAGE
                      $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 PEG= 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
```



COMMENTS: MIN READ LOOP TIME = 2μs.
MFBTR = 0:
MAX READ LOOP TIME = 20μs.*
MFBTR = 1:
MAX READ LOOP TIME = 80μs.
MINIMUM TIME OUT LOOP COUNTER = 0.5sec
*ONLY FOR SINGLE PAGE AND LAST PAGE OF MULTIPAGE TRANSFERS.

Figure 15. READ

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

 LOC  OBJ         LINE        SOURCE STATEMENT

                  ;**********************************************************************************
                  ;
                  ; FUNCTION  ABORT
                  ; INPUTS:   BPK72 STATUS REG
                  ; OUTPUTS:  ISSUE ABORT COMMAND TO BPK72
                  ;           A REG= BPK72 STATUS REG
                  ; CALLS:    NONE
                  ; DESTROYS: A, F/FS
                  ;
                  ; DESCRIPTION: ABORT PRESENT COMMAND, RESET BPK72
                  ;       AN ABORT  COMMAND IS  ISSUED  TO THE BPK72.  AFTER  ISSUING THE
                  ;       COMMAND,  THE  BPK72 STATUS REG IS POLLED UNTIL AN OP-COMPLETE,
                  ;       40H, HAS BEEN  READ  OR THE  TIME OUT  LOOP COUNTER DECREMENTS
                  ;       TO ZERO.  THE ABORT FUNCTION  RETURNS  THE  VALUE  OF THE BPK72
                  ;       STATUS REG TO THE CALLING ROUTINE VIA THE 8085'S A REG.  ONLY A
                  ;       STATUS  OF 40H INDICATES A  SUCCESSFUL  EXECUTION  OF THE ABORT
                  ;       FUNCTION
                  ;
                          PUBLIC  ABORT   ; DECLARE PUBLIC FUNCTION
08DE D5               ABORT   PUSH    D       ; SAVE D-E REGS
08DF C5                       PUSH    B       ; SAVE B-C REGS
08E0 11FFFF                   LXI     D,0FFFFH; INITIALIZE TIME OUT LOOP COUNTER
08E3 0640                     MVI     B,40H   ; LOAD B REG= 40H, OP-COMPLETE
08E5 3E19                     MVI     A,19H   ; LOAD A REG= ABORT COMMAND
08E7 D3FF                     OUT     PRTA01  ; WRITE ABORT COMMAND
08E9 DBFF             BUSYA:  IN      PRTA01  ; READ STATUS REG
08EB 07                       RLC             ; TEST BUSY BIT= 1
08EC DAF908                   JC      POLLA   ; IF BUSY= 1, POLL STATUS REG FOR 40H
08EF 1B                       DCX     D       ; DECREMENT TIME OUT LOOP COUNTER
08F0 AF                       XRA     A       ; CLEAR A REG
08F1 B2                       ORA     D       ; TEST D REG= 00H
08F2 B3                       ORA     E       ; TEST E REG= 00H
08F3 C2E908                   JNZ     BUSYA   ; IF NOT ZERO, CONTINUE POLLING ABORT COMMAND
08F6 C30609                   JMP     RETA    ; TIME OUT ERROR, RETURN
08F9 DBFF             POLLA   IN      PRTA01  ; READ STATUS REG
08FB A8                       XRA     B       ; TEST STATUS= 40H, OP-COMPLETE
08FC CA0609                   JZ      RETA      IF OP-COMPLETE, JMP RETA
08FF 1B                       DCX     D       ; DECREMENT TIME OUT LOOP COUNTER
0900 AF                       XRA     A       ; CLEAR A REG
0901 B2                       ORA     D       ; TEST D REG= 00H
0902 B3                       ORA     E       ; TEST E REG= 00H
0903 C2F908                   JNZ     POLLA   ; IF NOT ZERO, CONTINUE POLLING ABORT COMMAND
0906 C1               RETA    POP     B       ; RESTORE B-C REGS
0907 D1                       POP     D       ; RESTORE D-E REGS
0908 DBFF                     IN      PPTA01  ; READ STATUS REG
090A C9                       RET             ; RETURN TO CALL
                  ;
                  ;
                  ;
                  $EJECT
```

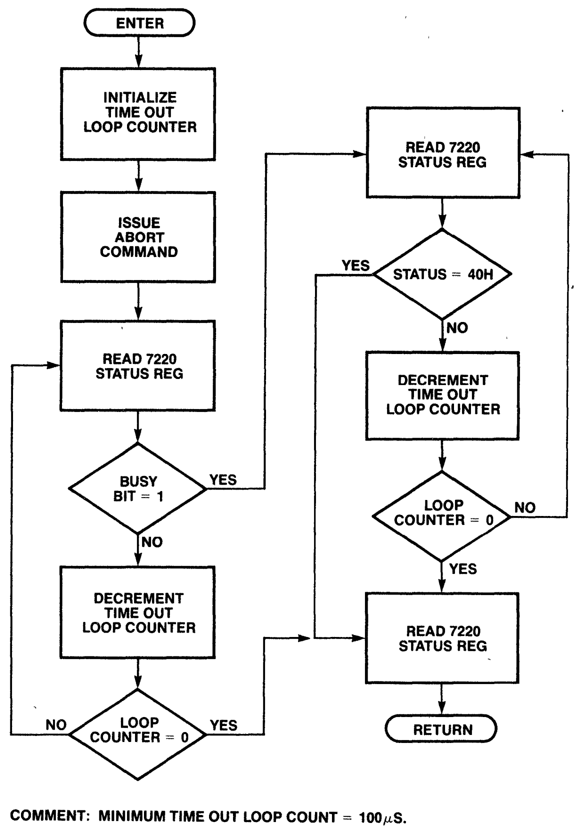


Figure 16. ABORT

```
ISIS-II 8060/8085 MACRO ASSEMBLER, V3 0        BPK72    PAGE   11

 LOC  OBJ         LINE        SOURCE STATEMENT

                       ;**************************************************************************
                       ;
                       ; FUNCTION: WRBUBL
                       ; INPUTS:    B-C REGS, STARTING ADDRESS OF PARAMETRIC REGS IN RAM
                       ;            D-E REGS, STARTING ADDRESS OF DATA IN RAM
                       ;            BPK72 STATUS REG
                       ; OUTPUTS:   WRITE DATA TO BUBBLE MEMORY
                       ;            A REG= BPK72 STATUS REG
                       ; CALLS      FIFORS
                       ;            INTPAR
                       ;            BYTCNT
                       ;            WRITE
                       ; DESTROYS  A, F/FS
                       ;
                       ; DESCRIPTION: WRITE BUBBLE MEMORY DATA
                       ;        THE B-C  REGS CONTAIN THE ADDRESS  TO THE  FIRST  OF FIVE
                       ;        CONTIGUOUS  MEMORY  LOCATIONS IN RAM.  THE DATA ADDRESSED
                       ;        BY THE  B-C  REGS IS USED  TO LOAD THE  PARAMETRIC  REGS.
                       ;        THE D-E  REGS  CONTAIN  THE STARTING ADDRESS  IN  RAM  OF
                       ;        DATA TO BE WRITTEN INTO THE BUBBLE MEMORY. GIVEN THE DATA
                       ;        IN RAM  USED TO  LOAD THE PARAMETRIC REGS,  THIS FUNCTION
                       ;        WILL  RESET  THE 7220 FIFO,  LOAD  THE  PARAMETRIC  REGS,
                       ;        COMPUTE THE BYTE COUNTER, AND COPY THE DATA FROM RAM INTO
                       ;        THE BUBBLE MEMORY.  WRBUBL RETURNS THE VALUE OF THE BPK72
                       ;        STATUS  REG TO THE CALLING ROUTINE VIA  THE 8085'S A REG.
                       ;        ONLY  A  STATUS  OF  40H OR 42H  INDICATES  A  SUCCESSFUL
                       ;        EXECUTION OF WRBUBL
                       ;
                                PUBLIC  WPBUBL  ; DECLARE PUBLIC FUNCTION
090B E5                WPBUBL:  PUSH    H       ; SAVE H-L REGS
090C C5                         PUSH    B       ; SAVE B-C REGS
090D 0640                       MVI     B,40H   ; LOAD B REG= 40H, OP-COMPLETE
090F CD1308                     CALL    FIFORS  ; CALL FIFORS, WRITE FIFO RESET COMMAND
0912 A8                         XRA     B       ; TEST FOR STATUS= 40H, OP-COMPLETE
0913 C23109                     JNZ     RETWR   ; IF NOT ZERO, FIFO ERROR, JMP RETWR
0916 C1                         POP     B       ; RESTORE B-C REGS
0917 CD0008                     CALL    INTPAR  ; CALL INTPAR, LOAD PARAMETRIC REGS
091A CD4008                     CALL    BYTCNT  ; CALL BYTCNT, COMPUTE BYTE COUNTER
091D CD6A08                     CALL    WRITE   ; CALL WRITE,  WRITE BUBBLE DATA
0920 C5                         PUSH    B       ; SAVE B-C REGS
                                                ; CONTINUED ON NEXT PAGE
                       $EJECT
```

```
ISIS-II 8080/8085 MACRO ASSEMBLER, V2.0        BPK72    PAGE   12

 LOC  OBJ         LINE        SOURCE STATEMENT

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

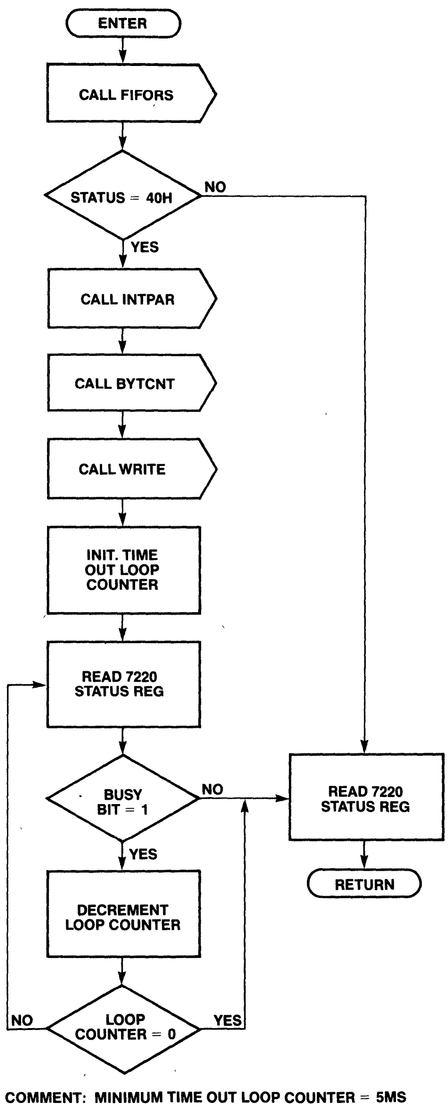


Figure 17. WRBUBL

```
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
```

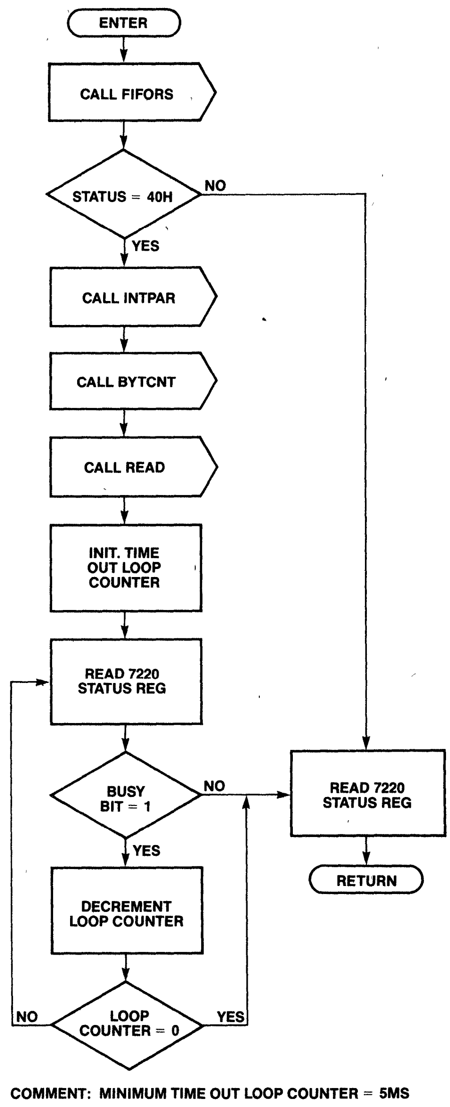


Figure 18. RDBUBL

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

 LOC  OBJ         LINE       SOURCE STATEMENT

                        ;*******************************************************************************
                        ;
                        ; FUNCTION: INBUBL
                        ; INPUTS.   B-C REGS, STARTING ADDRESS OF PARAMETRIC REGS IN RAM
                        ;           BPK72 STATUS REG
                        ; OUTPUTS:  A REG= BPK72 STATUS REG
                        ; CALLS.    ABORT
                        ;           INTPAR
                        ; DESTROYS: A, F/FS
                        ;
                        ; DESCRIPTION: INITIALIZE THE BPK72
                        ;       THE B-C REGS CONTAIN THE ADDRESS TO THE FIRST OF  FIVE CONTIGUOUS
                        ;       MEMORY LOCATIONS IN RAM.  THE  DATA ADDRESSED  BY  THE  B-C  REGS
                        ;       IS  USED TO  LOAD THE  PARAMETRIC REGS.  THIS FUNCTION WILL WRITE
                        ;       THE  PARAMETRIC   REGS  FOLLOWED  BY  ISSUING  A  BUBBLE   MEMORY
                        ;       INITIALIZATION  COMMAND.  AFTER  ISSUING  THE COMMAND,  THE BPK72
                        ;       STATUS REG IS POLLED UNTIL AN OP-COMPLETE,  40H,  IS  READ OR THE
                        ;       TIME  OUT  LOOP  COUNTER  DECREMENTS TO ZERO.  THIS  COMMAND MUST
                        ;       PRECEED ALL  OTHER  COMMANDS AFTER POWERING UP THE BPK72.  INBUBL
                        ;       RETURNS THE VALUE OF THE BPK72 STATUS  REG TO THE CALLING ROUTINE
                        ;       VIA THE 8085'S A REG. ONLY A STATUS OF 40H INDICATES A SUCCESSFUL
                        ;       EXECUTION OF INBUBL.
                        ;
                                PUBLIC  INBUBL  ; DECLARE PUBLIC FUNCTION
0961 D5                 INBUBL: PUSH    D       ; SAVE D-E REGS
0962 C5                         PUSH    B       ; SAVE B-C REGS
0963 0640                       MVI     B,40H   ; LOAD B REG= 40H, OP-COMPLETE
0965 CDDE08                     CALL    ABORT   ; CALL ABORT COMMAND
0968 A8                         XRA     B       ; TEST STATUS= 40H, OP-COMPLETE
0969 C29709                     JNZ     RETIN   ; IF ZERO, OP-COMPLETE, CONTINUE
096C C1                         POP     B       ; PARAMETRIC REGS STARTING ADDRESS IN REG B
096D CD0008                     CALL    INTPAR  ; CALL INTPAR, LOAD PARAMETRIC REGS
0970 C5                         PUSH    B       ; SAVE B-C REGS
0971 0640                       MVI     B,40H   ; LOAD B REG= 40H, OP-COMPLETE
0973 11FFFF                     LXI     D,0FFFFH; INITIALIZE TIME OUT LOOP COUNTER
0976 3E11                       MVI     A,11H   ; LOAD A REG= INITIALIZE COMMAND
0978 D3FF                       OUT     PRTA01  ; WRITE INITIALIZE COMMAND
                                                ; CONTINUED ON NEXT PAGE
                        $EJECT
```

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

```
 LOC  OBJ         SOURCE STATEMENT

097A DBFF        BUSYIN: IN      PRTA01  ; READ STATUS REG
097C 07                  RLC             ; TEST BUSY BIT= 1
097D DA8A09              JC      POLLIN  ; IF BUSY= 1, POLL STATUS REG FOR 40H
0980 1B                  DCX     D       ; DECREMENT TIME OUT LOOP COUNTER
0981 AF                  XRA     A       ; CLEAR A REG
0982 B2                  ORA     D       ; TEST D REG= 00H
0983 B3                  ORA     E       ; TEST E REG= 00H
0984 C27A09              JNZ     BUSYIN  ; IF NOT ZERO, CONTINUE POLLING THE INITIALIZE COMMAND
0987 C39709              JMP     RETIN   ; TIME OUT ERROR, RETURN
098A DBFF        POLLIN: IN      PRTA01  ; READ STATUS REG
098C A8                  XRA     B       ; TEST STATUS= 40H, OP-COMPLETE
098D CA9709              JZ      RETIN   ; IF OP-COMPLETE, JMP RETIN
0990 1B                  DCX     D       ; DECREMENT TIME OUT LOOP COUNTER
0991 AF                  XRA     A       ; CLEAR A REG
0992 B2                  ORA     D       ; TEST D REG= 00H
0993 B3                  ORA     E       ; TEST E REG= 00H
0994 C28A09              JNZ     POLLIN  ; IF NOT ZERO, CONTINUE POLLING INITIALIZE COMMAND
0997 C1          RETIN:  POP     B       ; RESTORE B-C REGS
0998 D1                  POP     D       ; RESTORE D-E REGS
0999 DBFF                IN      PRTA01  ; READ STATUS REG
099B C9                  RET             ; RETURN TO CALL
                 ;
                 ;
                 $EJECT
```

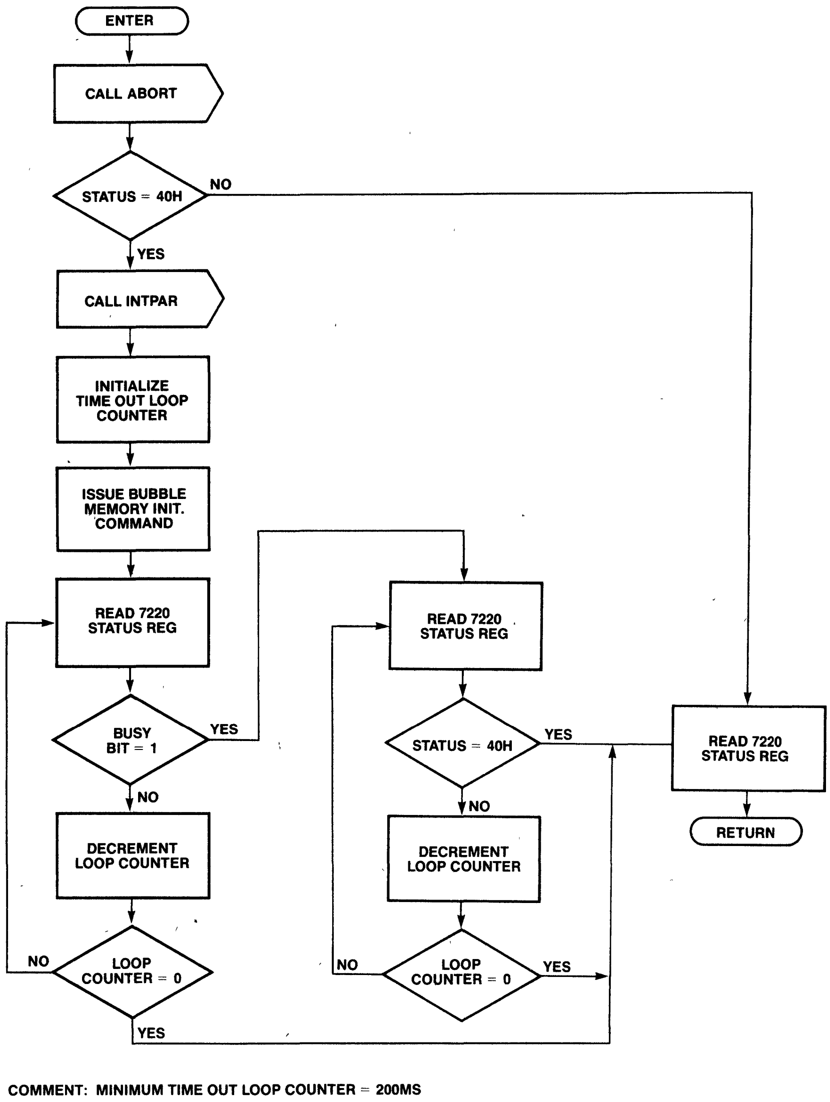


Figure 19. INBUBL

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

 LOC  OBJ         LINE        SOURCE STATEMENT

                  ;******************************************************************************
                  ;
                  ; FUNCTION: BOOTUP
                  ; INPUTS:   B-C REGS, STARTING ADDRESS OF PARAMETRIC REGS IN RAM
                  ;           D-E REGS, STARTING ADDRESS OF BOOT LOOP CODE IN RAM
                  ;           BPK72 STATUS REG
                  ; OUTPUTS:  WRITE BUBBLE MEMORY BOOT LOOP
                  ;           A REG= BPK72 STATUS REG
                  ; CALLS:    FIFORS
                  ;           INTPAR
                  ; DESTROYS: A, F/FS
                  ;
                  ; DESCRIPTION: WRITE BUBBLE MEMORY BOOT LOOP
                  ;       THIS  FUNCTION WILL  WRITE THE  BOOT LOOP CODE  INTO THE 7110
                  ;       BUBBLE MEMORY. THE D-E  REGS PROVIDE THE ADDRESS TO THE FIRST
                  ;       OF FORTY  CONTIGUOUS BYTES  IN RAM THAT CONTAIN THE BOOT LOOP
                  ;       CODE. THE B-C REGS  CONTAIN  THE ADDRESS TO THE FIRST OF FIVE
                  ;       CONTIGUOUS MEMORY  LOCATIONS ALSO IN RAM.  THE DATA ADDRESSED
                  ;       BY THE  B-C  REGS  IS  USED  TO  LOAD  THE  PARAMETRIC  REGS.
                  ;       NOTE   THAT  THE  PARAMETRIC  ENABLE  REG  WRITE   BOOT  LOOP
                  ;       BIT  IS  AUTOMATICALLY  SET  AND A  FORTY-FIRST BYTE  OF ZERO
                  ;       IS WRITTEN  TO THE  FIFO DATA BUFFER TO AVOID A TIMING ERROR.
                  ;       BEFORE  A RETURN IS EXECUTED,  THE PARAMETRIC  ENABLE  REG IS
                  ;       RESTORED TO ITS VALUE PRIOR TO CALLING BOOTUP.  BOOTUP RETURNS
                  ;       THE VALUE OF THE BPK72  STATUS REG TO THE CALLING ROUTINE VIA
                  ;       THE 8085'S  A REG.  ONLY  A STATUS  OF  40H OR 42H  INDICATES
                  ;       A SUCCESSFUL EXECUTION OF BOOTUP.
                  ;
                          PUBLIC  BOOTUP  ; DECLARE PUBLIC FUNCTION
099C C5           BOOTUP: PUSH    B       ; SAVE B-C REGS
099D D5                   PUSH    D       ; SAVE D-E REGS
099E E5                   PUSH    H       ; SAVE H-L REGS
099F 3E0D                 MVI     A,0DH   ; LOAD A REG= 0DH, 7220 RAC ENABLE REG ADDRESS
09A1 D3FF                 OUT     PRTA01  ; WRITE 7220 RAC WITH ENABLE REG ADDRESS
09A3 03                   INX     B       ;
09A4 03                   INX     B       ; INCREMENT B-C REGS TO ENABLE REG RAM ADDRESS
09A5 0A                   LDAX    B       ; LOAD A REG= ENABLE REG FROM RAM
09A6 0610                 MVI     B,10H   ; LOAD B REG= BOOT LOOP ENABLE MASK
09A8 B0                   ORA     B       ; SET BOOT LOOP ENABLE BIT
09A9 D3FE                 OUT     PRTA00  ; WRITE ENABLE REG
09AB AF                   XRA     A       ; CLEAR A REG
09AC D3FF                 OUT     PRTA01  ; LOAD 7220 RAC WITH FIFO DATA BUFFER ADDRESS
09AE 0640                 MVI     B,40H   ; LOAD B REG= 40H, OP-COMPLETE
09B0 CD1308               CALL    FIFORS  ; CALL FIFO RESET
09B3 A8                   XRA     B       ; TEST STATUS= 40H, OP-COMPLETE
09B4 C2230A               JNZ     RETBT   ; IF NOT ZERO, ERROR, JMP RETBT
                                          ; CONTINUED ON NEXT PAGE
                  $EJECT
```

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

 LOC  OBJ           SOURCE STATEMENT

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

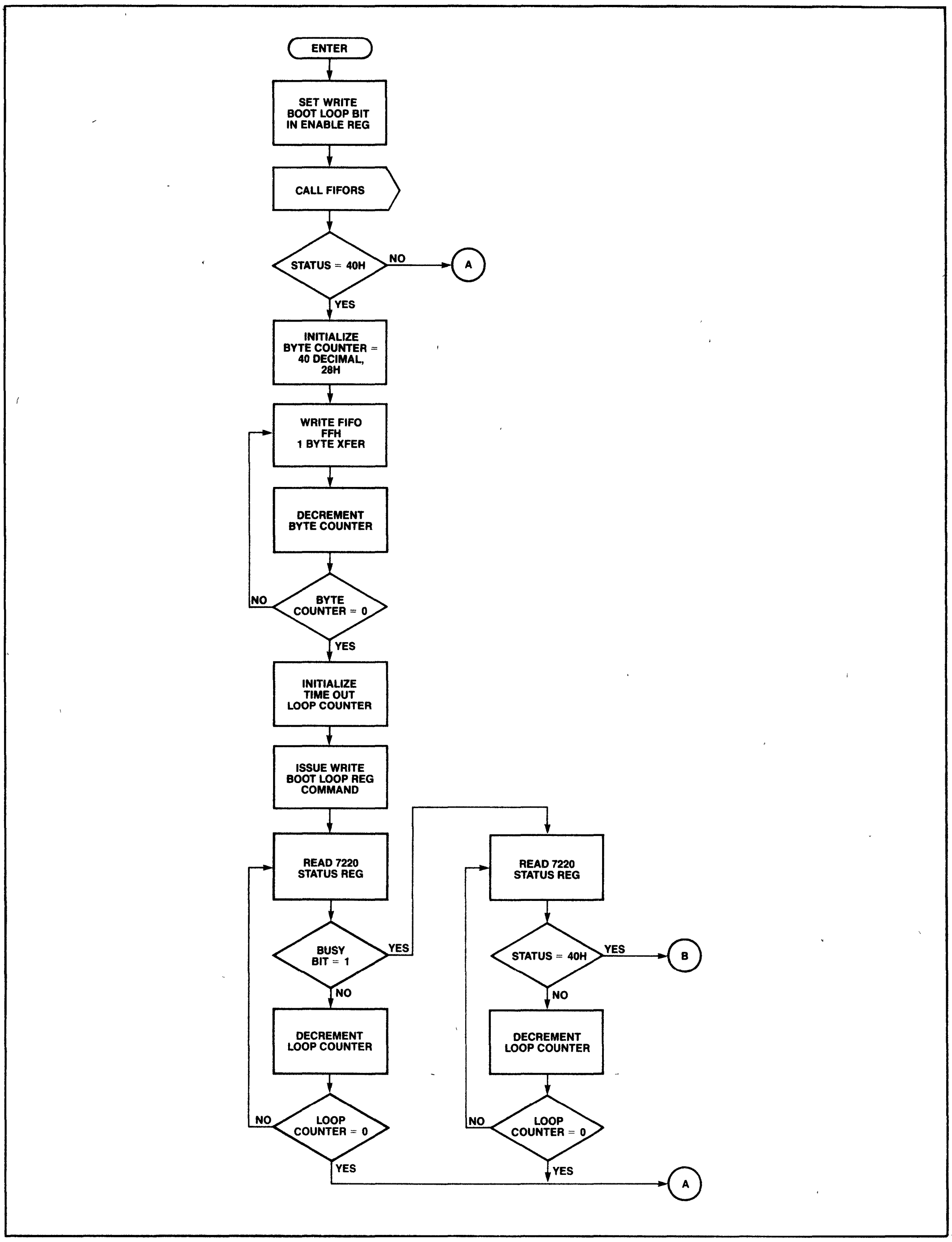


Figure 20. BOOTUP

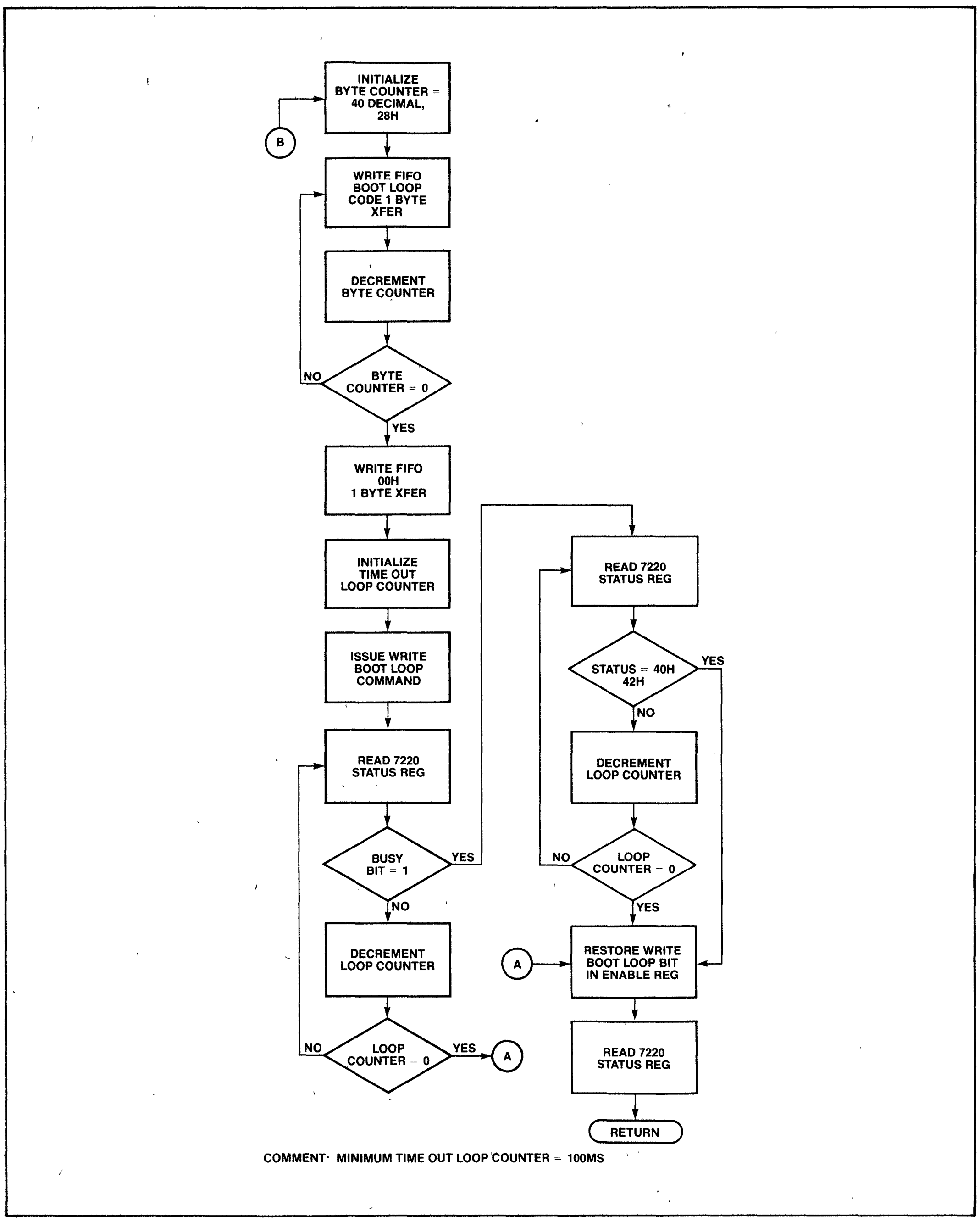


Figure 20 (Con't). BOOTUP

```
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
```

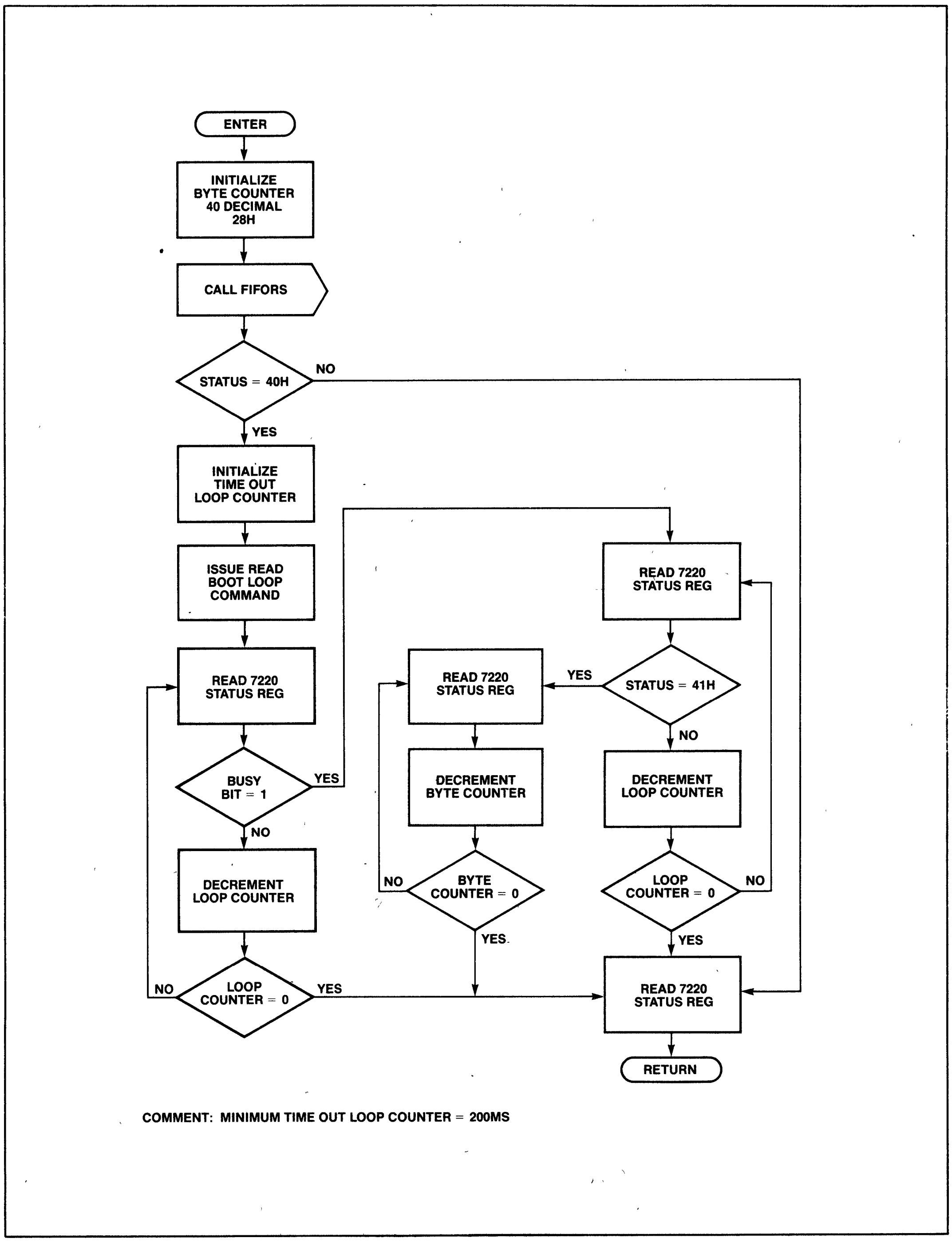


Figure 21. RDBOOT

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

 LOC  OBJ         LINE        SOURCE STATEMENT

                        ;**************************************************************************
                        ;
                        ; FUNCTION: WRFIFO
                        , INPUTS.    D-E REGS, STARTING ADDRESS OF DATA IN RAM
                        ,            BPK72 STATUS REG
                        ; OUTPUTS:   WRITE 40 BYTES IN THE BPK72 FIFO DATA BUFFER
                        ;            A REG= BPK72 STATUS REG
                        ; CALLS:     FIFORS
                        ; DESTROYS: A, F/FS
                        ;
                        ; DESCRIPTION: WRITE 7220 FIFO DATA BUFFER
                        ;       THE  D-E REGS PROVIDE THE ADDRESS TO THE FIRST OF 40 CONTIGUOUS
                        ;       BYTES IN RAM THAT CONTAIN DATA TO BE LOADED INTO THE BPK72 FIFO
                        ;       DATA BUFFER. WRFIFO WILL TRANSFER THE DATA FROM RAM TO THE FIFO
                        ;       DATA BUFFER.  WRFIFO RETURNS  THE VALUE OF THE BPK72 STATUS REG
                        ;       TO THE CALLING ROUTINE VIA THE 8085'S A REG.  ONLY  A STATUS OF
                        ;       41H OR 43H INDICATES A SUCCESSFUL EXECUTION OF WRFIFO.
                        ;
                                PUBLIC  WRFIFO  ; DECLARE PUBLIC FUNCTION
 0A70 C5                WRFIFO: PUSH    B       ; SAVE B-C REGS
 0A71 D5                        PUSH    D       ; SAVE D-E REGS
 0A72 0640                      MVI     B,40H   ; LOAD B REG= 40H, OP-COMPLETE
 0A74 0E28                      MVI     C,28H   ; LOAD C REG= 28H, INITIALIZE LOOP COUNTER
 0A76 CD1308                    CALL    FIFORS  , CALL FIFORS, WRITE FIFO RESET COMMAND
 0A79 A8                        XRA     B       ; TEST FOR STATUS REG= 40H, OP-COMPLETE
 0A7A C2850A                    JNZ     RETWF   ; IF NOT ZERO, FIFO ERROR, JMP RETWF
 0A7D 1A                INFIFO. LDAX    D       , LOAD A REG FROM D-E REG ADDRESS
 0A7E D3FE                      OUT     PRTA00  ; WRITE A REG TO 7220 FIFO DATA BUFFER
 0A80 13                        INX     D       ; INCREMENT D-E REGS TO NEXT ADDRESS IN RAM
 0A81 0D                        DCR     C       ; DECREMENT LOOP COUNTER
 0A82 C27D0A                    JNZ     INFIFO  ; IF LOOP COUNTER NOT ZERO, JMP INFIFO
 0A85 D1                RETWF:  POP     D       ; RESTORE D-E REGS
 0A86 C1                        POP     B       ; RESTORE B-C REGS
 0A87 DBFF                      IN      PRTA01  ; READ STATUS REG
 0A89 C9                        RET             ; RETURN TO CALL
                        ;
                        ;
                        ;
                        $EJECT
```

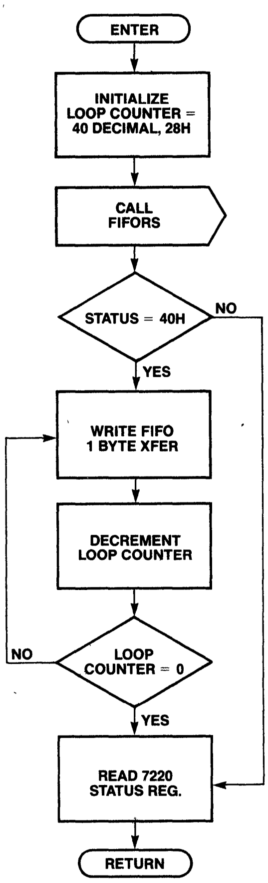


Figure 22. WRFIFO

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

 LOC  OBJ         LINE        SOURCE STATEMENT

                        ;***********************************************************************
                        ;
                        ; FUNCTION: RDFIFO
                        ; INPUTS:   D-E REGS STARTING ADDRESS IN RAM
                        ;           BPK72 STATUS REG
                        ;           READ 40 BYTES OF DATA FROM BPK72 FIFO DATA BUFFER
                        ; OUTPUTS:  TRANSFER FIFO DATA BUFFER TO RAM
                        ;           A REG= BPK72 STATUS REG
                        ; CALLS:    NONE
                        ; DESTROYS. A, F/FS
                        ;
                        ; DESCRIPTION: READ 7220 FIFO DATA BUFFER
                        ;       THE  D-E  REGS  CONTAIN THE ADDRESS TO THE  FIRST OF 40 CONTIGUOUS
                        ;       BYTES IN RAM THAT  WILL BE LOADED  WITH THE CONTENTS OF THE  BPK72
                        ;       FIFO DATA BUFFER. RDFIFO WILL TRANSFER THE DATA FROM THE FIFO DATA
                        ;       BUFFER TO RAM.  RDFIFO RETURNS THE VALUE OF  THE BPK72 STATUS  REG
                        ;       TO THE CALLING ROUTINE VIA THE 8085'S A REG.  ONLY A STATUS OF 40H
                        ;       OR 42H INDICATES A SUCCESSFUL EXECUTION OF RDFIFO.
                        ;
                                PUBLIC  RDFIFO  ; DECLARE PUBLIC FUNCTION
0A8A C5                 RDFIFO: PUSH    B       ; SAVE B-C REGS
0A8B D5                         PUSH    D       ; SAVE D-E REGS
0A8C 0E28                       MVI     C,28H   ; LOAD C REG= 28H, INITIALIZE LOOP COUNTER
0A8E DBFE               OUTFIF: IN      PRTA00  ; LOAD A REG WITH ONE BYTE FROM FIFO DATA BUFFER
0A90 12                         STAX    D       ; LOAD A REG IN D-E REG ADDRESS
0A91 13                         INX     D       ; INCREMENT D-E REGS TO NEXT ADDRESS
0A92 0D                         DCR     C       ; DECREMENT LOOP COUNTER
0A93 C28E0A                     JNZ     OUTFIF  ; IF LOOP COUNTER NOT ZERO, JMP OUTFIF
0A96 D1                         POP     D       ; RESTORE D-E REGS
0A97 C1                         POP     B       ; RESTORE B-C REGS
0A98 DBFF                       IN      PRTA01  ; READ STATUS REG
0A9A C9                         RET             ; RETURN TO CALL
                        ;
                        ;
                        $EJECT
```

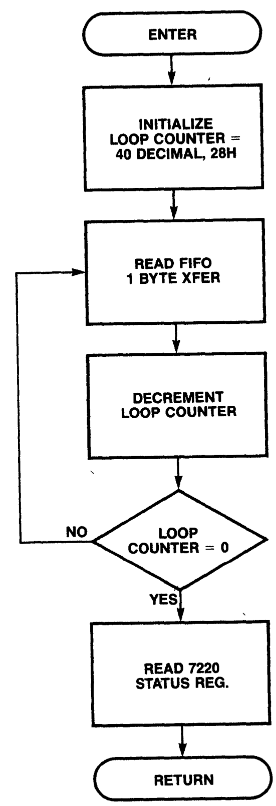


Figure 23. RDFIFO

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

```
 LOC  OBJ         LINE        SOURCE STATEMENT

                        ;***********************************************************************
                        ;
                        ; FUNCTION: WRBLRS
                        ; INPUTS:   D-E REGS, STARTING ADDRESS OF DATA IN RAM
                        ;           BPK72 STATUS REG
                        ; OUTPUTS:  WRITE BUBBLE MEMORY BOOT LOOP REGISTERS COMMAND
                        ;           A REG= BPK72 STATUS REG
                        ; CALLS:    WRFIFO
                        , DESTROYS: A, F/FS
                        ;
                        ; DESCRIPTION: WRITE 7242 BOOT LOOP REGISTERS
                        ;       THE D-E REGS PROVIDE THE ADDRESS TO THE FIRST OF 40 CONTIGUOUS
                        ;       MEMORY  LOCATIONS  IN RAM THAT  CONTAIN DATA TO BE LOADED INTO
                        ;       THE 7242,  FORMATTER  SENSE  AMPLIFIER,  BOOT  LOOP REGISTERS.
                        ;       WRBLRS WILL  TRANSFER  THE DATA  FROM  RAM  TO  THE  BOOT LOOP
                        ;       REGISTERS.  WRBLRS  RETURNS THE VALUE OF THE BPK72  STATUS REG
                        ;       TO THE CALLING  ROUTINE VIA THE 8085'S A REG. ONLY A STATUS OF
                        ;       40H INDICATES A SUCCESSFUL EXECUTION OF WRBLRS.
                        ;
                                PUBLIC  WRBLRS  ; DECLARE PUBLIC FUNCTION
0A9B C5                 WRBLRS: PUSH    B       ; SAVE B-C REGS
0A9C E5                         PUSH    H       ; SAVE H-L REGS
0A9D 0641                       MVI     B,41H   ; LOAD B REG= 41H, OP-COMPLETE, FIFO FULL
0A9F 0EFD                       MVI     C,0FDH  ; MASK, MASK OUT PARITY BIT
0AA1 21FFFF                     LXI     H,0FFFFH; INITIALIZE TIME OUT LOOP COUNTER
0AA4 CD700A                     CALL    WRFIFO  ; CALL WRITE FIFO DATA BUFFER
0AA7 A1                         ANA     C       ; RESET BIT 1, PARITY BIT
0AA8 A8                         XRA     B       ; TEST STATUS= 41H OR 43H, OP-COMPLETE, FIFO FULL
0AA9 C2CE0A                     JNZ     RETWBL  ; IF NOT ZERO, ERROR, JMP RETWBL
0AAC 05                         DCR     B       ; B REG= 40H, OP-COMPLETE
0AAD 3E16                       MVI     A,16H   ; LOAD A REG= WRITE BOOT LOOP REG COMMAND
0AAF D3FF                       OUT     PRTA01  ; WRITE, WRITE BOOT LOOP REG COMMAND
0AB1 DBFF               BSYWBL: IN      PRTA01  ; READ STATUS REG
0AB3 07                         RLC             ; TEST BUSY BIT= 1
0AB4 DAC10A                     JC      POLWBL  ; IF BUSY= 1, POLL STATUS REG FOR 40H
0AB7 2B                         DCX     H       ; DECREMENT TIME OUT LOOP COUNTER
0AB8 AF                         XRA     A       ; CLEAR A REG
0AB9 B4                         ORA     H       ; TEST H REG= 00H
0ABA B5                         ORA     L       ; TEST L REG= 00H
0ABB C2B10A                     JNZ     BSYWBL  ; IF NOT ZERO, CONTINUE POLLING WRBLR COMMAND
0ABE C3CE0A                     JMP     RETWBL  ; TIME OUT ERROR, RETURN
0AC1 DBFF               POLWBL: IN      PRTA01  ; READ STATUS REG
0AC3 A8                         XRA     B       ; TEST STATUS REG= 40H, OP-COMPLETE
0AC4 CACE0A                     JZ      RETWBL  ; IF ZERO, OP-COMPLETE, JMP RETWBL
0AC7 2B                         DCX     H       ; DECREMENT TIME OUT LOOP COUNTER
0AC8 AF                         XRA     A       ; CLEAR A REG
0AC9 B4                         ORA     H       ; TEST H REG= 00H
0ACA B5                         ORA     L       ; TEST L REG= 00H
0ACB C2C10A                     JNZ     POLWBL  ; IF NOT ZERO, CONTINUE POLLING WRBLR COMMAND
0ACE E1                 RETWBL: POP     H       ; RESTORE H-L REGS
0ACF C1                         POP     B       ; RESTORE B-C REGS
0AD0 DBFF                       IN      PRTA01  ; READ STATUS REG
0AD2 C9                         RET             ; RETURN TO CALL
                        $EJECT
```

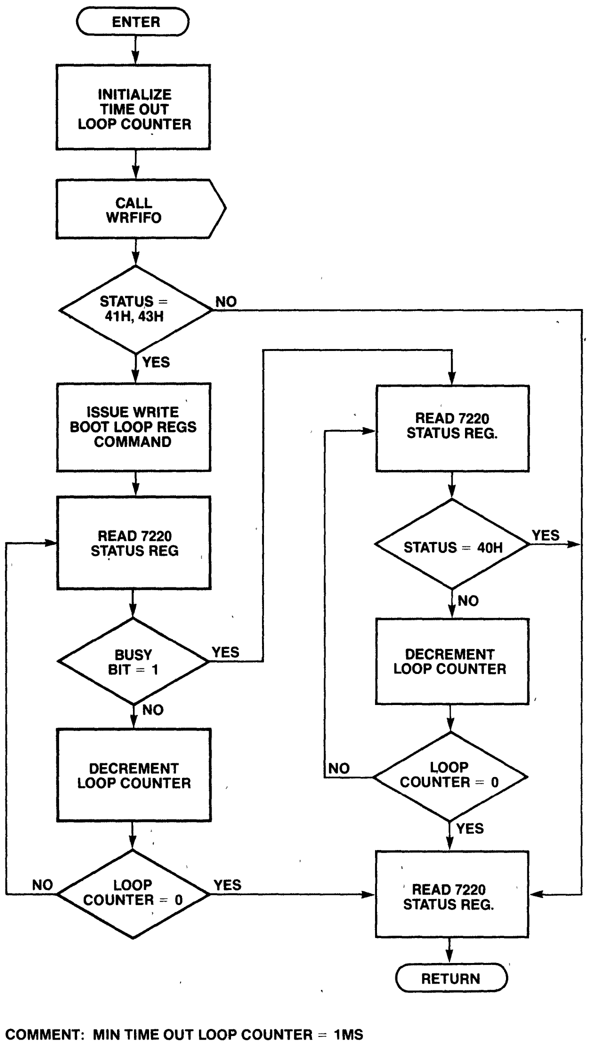


Figure 24. WRBLRS

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

 LOC  OBJ          LINE        SOURCE STATEMENT

                       ;***********************************************************************
                       ;
                       ; FUNCTION: RDBLRS
                       ; INPUTS:    D-E REGS, STARTING ADDRESS IN RAM
                       ;            BPK72 STATUS REG
                       ;            READ DATA FROM 7242 BOOT LOOP REGISTERS
                       ; OUTPUTS:   TRANSFER BOOT LOOP REGISTER DATA TO RAM
                       ;            A REG= BPK72 STATUS REG
                       ; CALLS:     RDFIFO
                       ; DESTROYS: A, F/FS
                       ;
                       ; DESCRIPTION: READ 7242 BOOT LOOP REGISTERS
                       ;        THE D-E REGS CONTAIN THE ADDRESS TO THE FIRST OF 40 CONTIGUOUS
                       ;        MEMORY LOCATIONS IN RAM TO BE  LOADED WITH THE CONTENTS OF THE
                       ;        7242, FORMATTER  SENSE AMPLIFIER, BOOT LOOP REGISTERS.  RDBLRS
                       ;        WILL COPY  THE CONTENTS  OF THE  BOOT  LOOP  REGISTERS TO RAM.
                       ;        RDBLRS  RETURNS THE  VALUE OF  THE  BPK72  STATUS  REG  TO THE
                       ;        CALLING ROUTINE  VIA THE 8085'S  A REG.  ONLY A  STATUS OF 40H
                       ;        INDICATES A SUCCESSFUL EXECUTION OF RDBLRS.
                       ;
                                PUBLIC  RDBLRS  ; DECLARE PUBLIC FUNCTION
 0AD3 C5                RDBLRS: PUSH    B       ; SAVE B-C REGS
 0AD4 E5                        PUSH    H       ; SAVE H-L REGS
 0AD5 06C1                      MVI     B,0C1H  ; LOAD B REG= C1H, OP-COMPLETE, FIFO FULL >22 BYTES (BUSY BIT=1)
 0AD7 21FFFF                    LXI     H,0FFFFH; INTIALIZE TIME OUT LOOP COUNTER
 0ADA 3E15                      MVI     A,15H   ; LOAD A REG= READ BOOT LOOP REGS COMMAND
 0ADC D3FF                      OUT     PRTA01  ; WRITE THE READ BOOT LOOP REGS COMMAND
 0ADE DBFF              BSYRBL: IN      PRTA01  ; READ STATUS REG
 0AE0 07                        RLC             ; TEST BUSY BIT= 1
 0AE1 DAEE0A                    JC      POLRBL  ; IF BUSY= 1, POLL STATUS REG FOR C1H
 0AE4 2B                        DCX     H       ; DECREMENT TIME OUT LOOP COUNTER
 0AE5 AF                        XRA     A       ; CLEAR A REG
 0AE6 B4                        ORA     H       ; TEST H REG= 00H
 0AE7 B5                        ORA     L       ; TEST L REG= 00H
 0AE8 C2DE0A                    JNZ     BSYRBL  ; IF NOT ZERO, CONTINUE POLLING READ BOOT LOOP REG COMMAND
 0AEB C3010B                    JMP     RETRBL  ; TIME OUT ERROR, RETURN
 0AEE DBFF              POLRBL: IN      PRTA01  ; READ STATUS REG
 0AF0 A8                        XRA     B       ; TEST STATUS= C1H, OP-COMPLETE, FIFO FULL
 0AF1 CAFE0A                    JZ      CALLRD  ; IF ZERO, OP-COMPLETE, JMP CALLRD
 0AF4 2B                        DCX     H       ; DECREMENT TIME OUT LOOP COUNTER
 0AF5 AF                        XRA     A       ; CLEAR A REG
 0AF6 B4                        ORA     H       ; TEST H REG= 00H
 0AF7 B5                        ORA     L       ; TEST L REG= 00H
 0AF8 CA010B                    JZ      RETRBL  ; IF ZERO, ERROR, JMP RETRBL
 0AFB C3EE0A                    JMP     POLRBL  ; CONTINUE POLLING READ BOOT LOOP REG COMMAND
 0AFE CD8A0A            CALLRD: CALL    RDFIFO  ; CALL READ FIFO
 0B01 E1                RETRBL: POP     H       ; RESTORE H-L REGS
 0B02 C1                        POP     B       ; RESTORE B-C REGS
 0B03 DBFF                      IN      PRTA01  ; READ STATUS REG
 0B05 C9                        RET             ; RETURN TO CALL
                        $EJECT
```

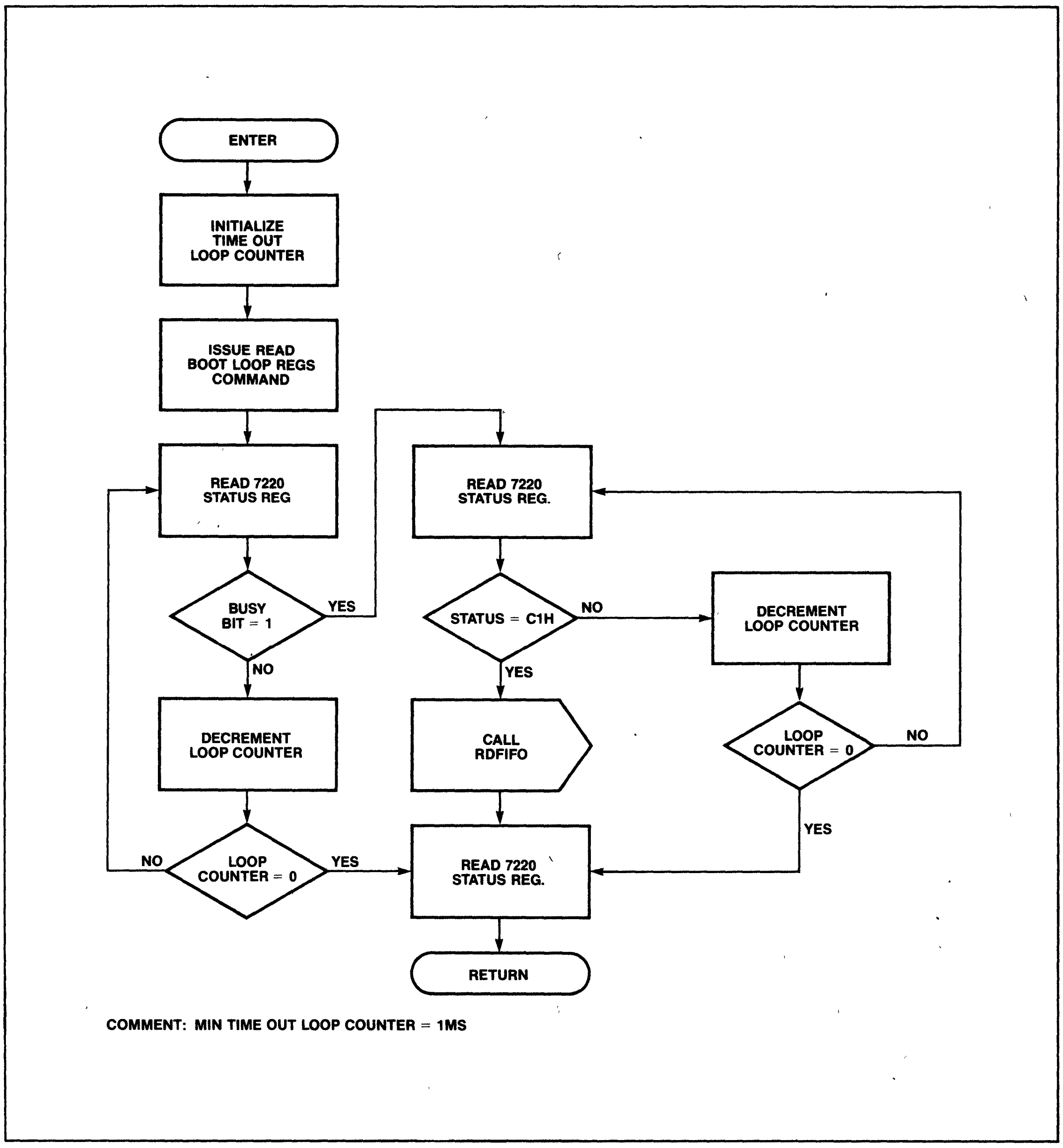


Figure 25. RDBLRS

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

 LOC  OBJ        SOURCE STATEMENT

                 ;*************************************************************************
                 ;
                 ; FUNCTION: MBMPRG
                 ; INPUTS:   BPK72 STATUS REG
                 ; OUTPUTS:  ISSUE MBM PURGE COMMAND
                 ;           A REG= BPK72 STATUS REG
                 ; CALLS:    NONE
                 ; DESTROYS: A, F/FS
                 ;
                 ; DESCRIPTION: MBM PURGE COMMAND
                 ;       AN MBM PURGE COMMAND IS ISSUED TO THE BPK72. AFTER ISSUING THE
                 ;       COMMAND, THE BPK72 STATUS REG IS POLLED UNTIL AN OP-COMPLETE
                 ;       40H, HAS BEEN READ OR THE TIME OUT LOOP COUNTER DECREMENTS
                 ;       TO ZERO. MBMPRG RETURNS THE VALUE OF THE BPK72 STATUS REG TO
                 ;       THE CALLING ROUTINE VIA THE 8085'S A REG. ONLY A STATUS OF 40H
                 ;       INDICATES A SUCCESSFUL EXECUTION OF MBMPRG.
                 ;
                         PUBLIC  MBMPRG  ; DECLARE PUBLIC FUNCTION
 0B06 D5         MBMPRG: PUSH    D       ; SAVE D-E REGS
 0B07 C5                 PUSH    B       ; SAVE B-C REGS
 0B08 0640               MVI     B,40H   ; LOAD B REG= 40H, OP-COMPLETE
 0B0A 11FFFF             LXI     D,0FFFFH; INITIALIZE TIME OUT LOOP COUNTER
 0B0D 3E1E               MVI     A,1EH   ; LOAD A REG= MBM PURGE COMMAND
 0B0F D3FF               OUT     PRTA01  ; WRITE MBM PURGE COMMAND
 0B11 DBFF       BSYMBM: IN      PRTA01  ; READ STATUS REG
 0B13 07                 RLC             ; TEST BUSY BIT= 1
 0B14 DA210B             JC      POLMBM  ; IF BUSY= 1, POLL STATUS REG FOR 40H
 0B17 1B                 DCX     D       ; DECREMENT TIME OUT LOOP COUNTER
 0B18 AF                 XRA     A       ; CLEAR A REG
 0B19 B2                 ORA     D       ; TEST D REG= 00H
 0B1A B3                 ORA     E       ; TEST E REG= 00H
 0B1B C2110B             JNZ     BSYMBM  ; IF NOT ZERO, CONTINUE POLLING THE MBMPRG COMMAND
 0B1E C32E0B             JMP     RETMBM  ; TIME OUT ERROR, RETURN
 0B21 DBFF       POLMBM: IN      PRTA01  ; READ STATUS REG
 0B23 A8                 XRA     B       ; TEST STATUS= 40H, OP-COMPLETE
 0B24 CA2E0B             JZ      RETMBM  ; IF OP-COMPLETE, JMP RETMBM
 0B27 1B                 DCX     D       ; DECREMENT TIME OUT LOOP COUNTER
 0B28 AF                 XRA     A       ; CLEAR A REG
 0B29 B2                 ORA     D       ; TEST D REG= 00H
 0B2A B3                 ORA     E       ; TEST E REG= 00H
 0B2B C2210B             JNZ     POLMBM  ; IF NOT ZERO, CONTINUE POLLING MBM PURGE COMMAND
 0B2E C1         RETMBM: POP     B       ; RESTORE B-C REGS
 0B2F D1                 POP     D       ; RESTORE D-E REGS
 0B30 DBFF               IN      PRTA01  ; READ STATUS REG
 0B32 C9                 RET             ; RETURN TO CALL
                 $EJECT
```

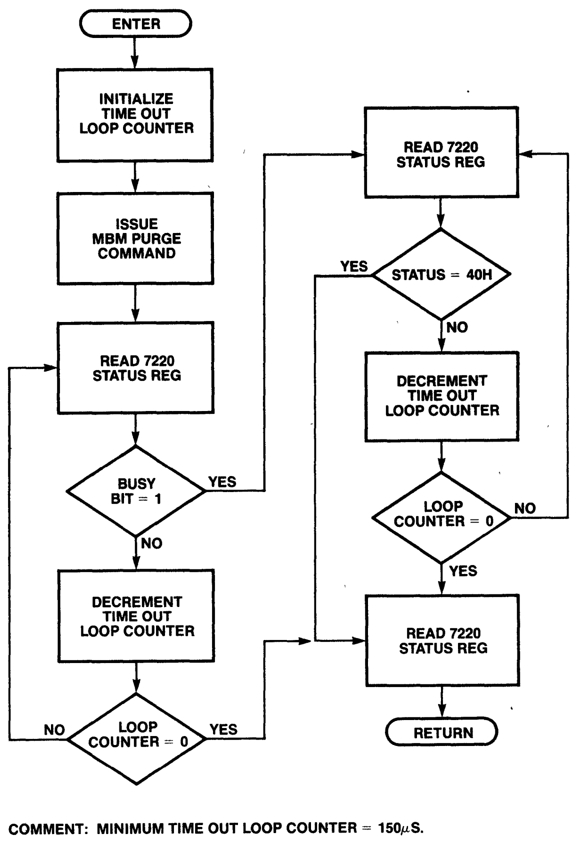


Figure 26. MBMPRG

```
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
POWERING-UP FOR THE FIRST TIME

POWERING-UP FOR THE FIRST TIME

The following procedures used to verify the operation of a BPK 72 should be performed with the dummy module in place of the 7110 Bubble Memory. No attempt should be made to use the 7110 Bubble Memory in the IMB-72 board until successfully completing tests 1, 2, 3A, 3B, 3C, and 3D.

The software driver in Appendix A, "BPK 72," contains several subroutines that can be used to systematically check-out a newly assembled BPK 72.

Test 1 ( )—After powering-up, the first step in checking out a new interface and BPK 72 is to verify the operation of the 7220's FIFO data buffer. Two subroutines, RDFIFO and WRFIFO, may be used to read and write 40 bytes to the FIFO data buffer. Additional detail concerning the operation of the subroutines, RDFIFO and WRFIFO, is available in the program listing. The read FIFO subroutine should be used to verify that the data in the FIFO data buffer is identical to the data written by the subroutine WRFIFO. An incrementing or decrementing data pattern is the most effective for testing the operation of the FIFO data buffer.

Incorrect data indicates a fundamental timing error and/or interface problem. In almost all cases, read and write FIFO data errors result from an interface or IMB-72 board wiring mistake.

Test 2 ( )—After successfully completing Test 1, communication between the 7220 controller and the 8085 microprocessor has been verified. The next step consists of verifying the communication path between the 7220 Bubble Memory Controller and the 7242 Formatter Sense Amplifier (FSA). Verification consists of comparing the data read from that written to the FSA's boot loop registers. Before attempting to read or write the boot loop registers, two subroutines must be called to clear the 7220. A call to the subroutine ABORT followed by a call to MBMPRG (Bubble Memory Purge Command) are necessary before any other commands may be issued to the BPK 72. The details concerning the use of the subroutines, ABORT and MBMPRG are presented in the program listing. After successfully executing an ABORT and MBMPRG command, communication between the 7220 and FSA can be verified using the subroutines, RDBLRS and WRBLRS (see program listing, Appendix A). RDBLRS and WRBLRS should be called to read and write the FSA's boot loop registers. The subroutine, RDBLRS, should be used to verify that the data in the boot loop registers is identical to that written by the subroutine WRBLRS. An incrementing or decrementing data pattern is also the most effective for testing the communication path between the 7220 controller and the FSA.

Test 3: Reading and writing to the 7110 Bubble Memory requires the application of specific control signals at the appropriate times within the read and write cycles. Test 3 consists of verifying the control signal waveforms.

A. ( ) The first control signal waveform to check is the coil drive on pins 9, 10, 11, and 12 of the 7110 Bubble Memory socket. The drive current can be verified by ensuring that the voltage waveform on these pins conforms to figure 29A when the field is rotated. To rotate the drive field, the following program sequence can be used:

1. Write 40 bytes of FFH into the boot loop registers via the subroutine WRBLRS.
2. Call RDBUBL (Read Bubble Memory)
   See the section titled, "Implementing the 8085/BPK72 Software Driver—Reading and Writing" for a detailed explanation of the subroutine RDBUBL. The following values should be used to load the parametric registers: FFH (BLR LSB), 10H (BLR MSB), 00H (ENABLE), 00H (add LSB), and 00H (add MSB).
3. Loop on RDBUBL.

In order to make a measurement of the coil drive waveforms, a multipage transfer is required. As shown above, the parametric block length register LSB is loaded with an FFH indicating the transfer of 255 contiguous pages, 68 bytes per page (17,340 total bytes). Since a 255 page transfer will take approximately two seconds, looping on the read Bubble Memory subroutine allows for a continuous measurement using a standard oscilloscope.

B. ( ) Next, the "cut and transfer" pulses generated during a read operation should be checked. The waveforms on pins 2 and 3 of the 7110 socket (replicate A and replicate B) should appear as shown in Figure 27B. The program sequence necessary to view the generate A and generate B waveforms is identical to the sequence used to verify the coil drive pulses with one exception; the write Bubble Memory subroutine, WRBUBL, must be used in place of the call to RDBUBL. The same values used to load the parametric register for RDBUBL should also be used for WRBUBL.

C. ( ) The "cut and transfer" pulses that occur during a Write Operation should now be verified. The waveforms on Pins 7 and 8 of the 7110 socket (generate A and generate B) should appear as shown in Figure 7C the program sequence necessary to view the generate A and generate B waveforms is identical to the sequence used to verify the coil drive pulses with one exception, the Write Bubble Memory subroutine, WRBUBL, must be used in place of the call to RDBUBL. The same values used to load the parametric registers for RDBUBL should also be used for WRBUBL.

D. ( ) Finally, the swap pulse must be tested for proper operation during a write operation. The waveforms on pins 13 and 14 of the 7110 socket (swap A and swap B) should appear as shown in Figure 27D. The program sequence used to measure the swap pulses is the same as that used to verify the write "cut and transfer" pulses.

After completing all the previous tests successfully, the 7110 Bubble Memory device may be inserted. Before attempting to insert the 7110 Bubble Memory, remove power from the system! Installing the 7110 is no different from installing any other device. Remove the dummy module in the 7110 socket and insert the 7110 Bubble Memory. Note that the 7110 is keyed to prevent the device from being inserted incorrectly. The user is now ready to put the BPK72 into actual system use.

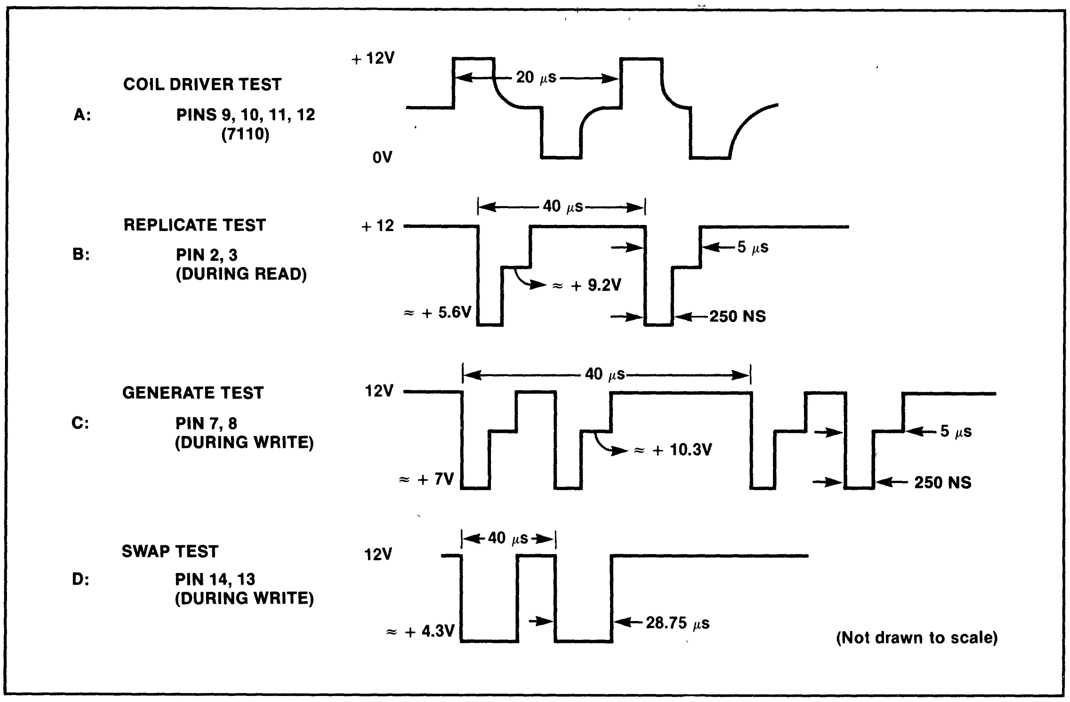


Figure 27. Control Signal Waveforms

APPENDIX C
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 C 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:

8085 Microprocessor		**8085 Addressable Memory**
B REG = XXH	C REG = XXH	
D REG = 30H	E REG = 00H →	3000H = First Byte*
		3027H = Last Byte
H REG = XXH	L REG = XXH	
H REG = XXH		*Boot Loop code read the 7110 Bubble Memory.
A REG = Will return the value of the status register (acceptable status = 40H)		

Call 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:

8085 Microprocessor			**8085 Addressable Memory**
B REG = 10H	C REG = 00H	→	1000H = 01H BLR LSB
D REG = 30H	E REG = 00H		1001H = 10H BLR MSB
H REG = XXH	L REG = XXH		1002H = 00H Enable REG
			1003H = 00H Add LSB
A REG = Will return the value of the status register (acceptable status = 40H, 42H)			1004H = 00H Add MSB
		→	3000H = First Byte*
			3027H = Last Byte
Call BOOTUP.			*Boot loop code found on the label attached to case of 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.

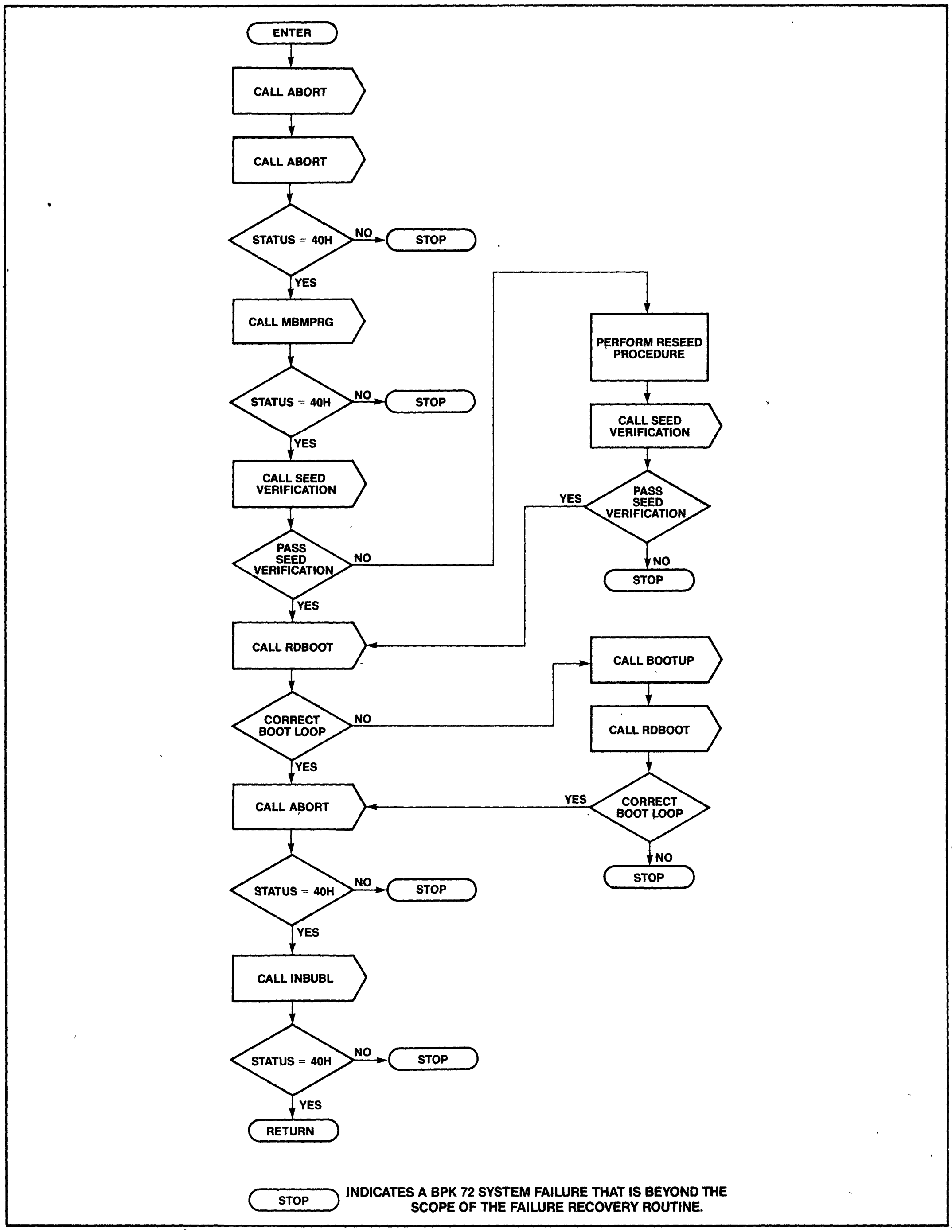
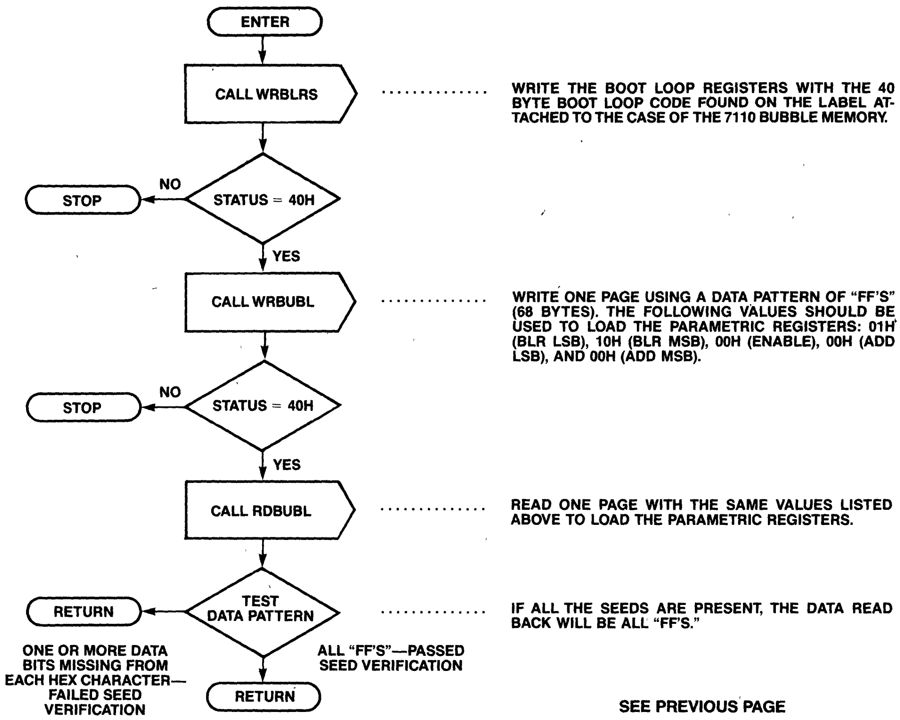


Figure 28. BPK 72 Failure Recovery



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.

ARTICLE REPRINT

AR-243

Thin-film detectors, X-ray lithography deliver 4-Mbit bubble chip

Next-generation bubble memory chip is even smaller than the compatible, 1-Mbit device; set of support circuits takes care of memory system requirements.

Propelled by X-ray lithography and thin-film permalloy detectors, bubble memory chips have climbed to the 4-Mbit level.

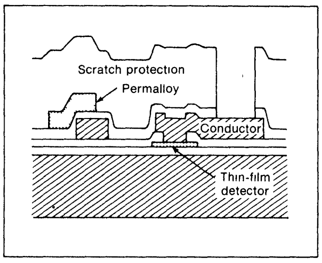
Using X-ray lithography, Intel Corp. (Santa Clara, Calif.) has managed to reduce the periodicity between bubbles from 11.2 (for its 1-Mbit chip) to 5.6 μm and feature sizes from 1.25 to 0.75 μm. At the same time, thin-film permalloy detectors, replacing thick-film versions, nearly double the signal strength of the detected bubbles (Fig. 1).

Moreover, a novel multiplexing technique handles the outputs from the eight on-chip detectors, which is double the number used on the 1-Mbit chip. This technique, which Intel is keeping under wraps, permits the higher-density chip to fit into a 22-pin package.

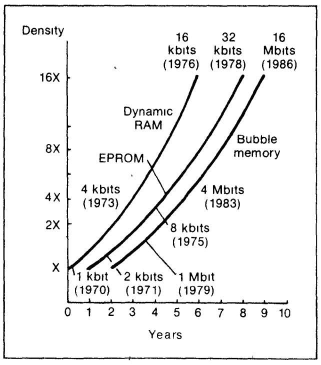
The outcome of all that is the 7114, plus a complement of six support circuits. The 7114 retains the basic architecture of the 1-Mbit 7110, and all the support circuits are pin-compatible with the chips that support the 7110. Aside from a few software changes to handle the larger memory space, the upgrade is totally transparent to the system user, claims Mike Eisele, bubble memory product manager. Thus in many cases the older bubble chips can be removed from a system and new ones plugged in.

Dave Bursky
Electronic Design

However, the support chips cannot control the 1-Mbit device, and some minor hardware changes must be made to accommodate the smaller package used for the 4-Mbit chip. The package's dimensions—1.46 by 1.35



1. A key element of Intel's 4-Mbit bubble memory is this thin-film permalloy detector structure, which delivers twice the output signal of the previously used thick-film detector.



2. Following the same growth curve as UV EPROMs and dynamic RAMs, bubble memory technology still has a good way to go to reach the 16-Mbit level projected for 1986.

in.—represent a savings of nearly 0.9 in.^{2} over the 1-Mbit package's 1.7 by 1.68 in. In addition, the smaller package, which has DIP-like pins, eliminates the need for a socket in many cases and also has a lower profile to permit board spacings as close as 0.6 in. The same package will be used by Motorola Inc. (Phoenix, Ariz.) when it builds the second-generation 1-Mbit chip as called for in the alternative-source agreement signed earlier this year with Intel (ELECTRONIC DESIGN, July 8, p. 23).

However, to bring the price of the bubble memories down to what Eisele feels would be attractive for system users—about $150 for a 4-Mbit chip by 1986—Intel has turned to a Perkin-Elmer X-ray lithography system in what it believes to be the first commercial use of X-ray systems. (Other companies, though, are not very far behind—many semiconductor manufacturers have very active research and development programs to make X-ray systems practical on the production line.)

The production process for the 4-Mbit chip includes 90% of the process steps used for the 1-Mbit device, thus sharing much of the learning-curve experience, in the short run.

Functionally, the 4-Mbit device will appear to operate just like the 1-Mbit memory. However, when the 7114 operates at the 50-kHz field rate of the 1-Mbit device, the access time is double that of the smaller chip, since the loops are longer. But the data rate is double that of the 1-Mbit chip because more detector outputs are multiplexed and then fed out from the chip. Also, a version of the 4-Mbit chip will operate at twice the field rate (100 kHz), for an access time of 41 ms—almost the 40-ms access time of the 1-Mbit chip.

There will be a full kit of parts available from Intel when samples of the memory will be available next year. The largest chip will be the 7224 controller, which duplicates the functions of the 7220 controller but has the internal changes needed to handle the larger memory space. Similarly, the other circuits are the 7234 current-pulse generator, the 7244 formatter-sense amplifier, the 7250 coil predriver, and the 7254 coil drivers.

Bubble memory capacity has been quadrupling about every four to five years. This follows very closely what happened to UV EPROMs (Fig. 2), even though EPROMs went through doubling cycles every two years.

intel®

ARTICLE REPRINT

AR-250

November 1982

Bubble Chip Packs 4 Mbits Into 1-Mbit Space

Hudson Washburn
Design Engineer

Sam Nicolino
Design Engineer
Intel Corporation
Santa Clara, California

Electronic Design
Reprinted with permission from Electronic Design, Volume 30, Number 23 Copyright Hayden Publishing, Inc , 1982

ORDER NUMBER: 210842-001

BehindTheCover

Right after putting their 1-Mbit bubble memory chip into production several years ago, designers at Intel decided to try various sections of what would be needed to build a 4-Mbit device. Although several were fabricated and proved functional, priorities in ironing out the production problems for the 1-Mbit chip forced them to put the 4-Mbit design on the back burner, working on it as a secondary project. Finally, though, the years of patience are paying off, and as our cover story in this issue (p. 1) highlights, the 4-Mbit magnetic bubble memory—the i7114—is functional.

Fortunately, the designers have been able to time the developments so that both the bubble chip and its associated support chips will be ready at the same time. As Mike Eisele, product manager for the Magnetic Bubble Memory Division, notes, that wasn't the case for the 1-Mbit device—it took Intel a lot longer than it expected to make the controller fully functional.

In developing the 4-Mbit memory, Hudson Washburn, design engineer, expected that the control elements on the chip—the bubble generator, transfer gates, replicator, and detector—would be the most difficult sections to get to work, whereas he thought that the propagation paths would be relatively simple to implement. But when actually trying to create the memory chip, he and the other researchers found that the control sections performed fine after only a few iterations while the propagation paths turned out to be the tricky development problem.

Additionally, mastering the technology needed to build the 4-Mbit bubble chip was a long, hard process with many half steps back, Washburn says. However, work on the 1-Mbit device also helped the bigger memory: Every time something happened that caused yield problems on the 1-Mbit chip, work was stopped on the new circuit. When the problem or problems on the 1-Mbit process were solved, the designers applied what they learned to the 4-Mbit technology.

Also, the designers decided to use a thin-film detector structure to boost the signal-to-noise ratio of the output signal. Although building this detector adds a second critical masking level to the production process, the decrease in yield due to the additional step is expected to be more than offset by faster testing. As it turns out, testing tends to be a major part of the chip cost as the capacity reaches 4 Mbits, according to Dave Dossetter, bubble memory product marketing engineer.

Perhaps appropriately for a 4-Mbit memory, Intel worked with a manufacturer of lithography equipment and a mask maker to use X-ray lithography. Although contact printing was employed during development, Intel plans to put X-ray lithography to work for volume production, which would make it the first such commercial use.

A 4-Mbit bubble memory chip, supported by a full complement of six dedicated circuits, stands poised for applications ranging from industrial control to telecommunications to personal computers.

Bubble chip packs 4 Mbits into 1-Mbit space

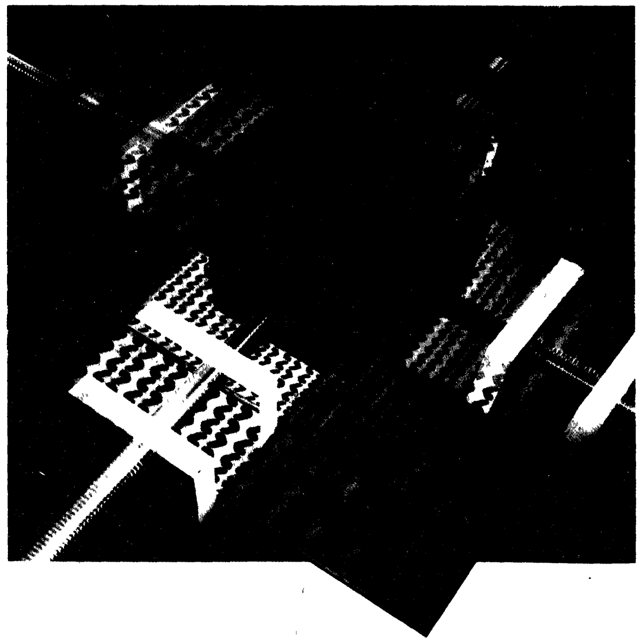
Bubble memories sport a hefty list of advantages for mass storage applications. Yet because of the complexity of interfacing them, most designers have shied away from these devices, leaving them outcasts. But the sheer appeal of 4 Mbits tucked into a 20-pin package, coupled with a set of components that takes care of the complexities of linking a bubble chip to conventional host computers, makes an extremely attractive option for those designers who have previously resigned themselves to simpler but less attractive mass storage.

As for those who have already taken the plunge into bubbles with the chip's 1-Mbit predecessor, the 7110, upgrading to the 4-Mbit 7114 requires only minimal changes.

Some of those ready to benefit from a simplified bubble memory system are portable equipment makers, who will take advantage of the compactness and nonvolatility of bubble chips. Industrial control and robotics manufacturers will appreciate bubble devices' resistance to hostile environments, since they have no moving mechanical parts to succumb to shock, corrosion, or high humidity. These last three qualities also are important to telecommunications suppliers, who need low-cost, reliable buffers for PABX and other message-carrying systems.

Still, to reap the rewards inherent in bubble memories, a full complement of support circuits must accompany the bubble chip itself. Those companions are ready, in the form of the 7224 bubble memory controller, the 7244 formatter and sense amplifier, the 7250 coil predriver, the 7254 VMOS driver transistor, and the 7234 current pulse driver.

Despite these components, a 4-Mbyte bubble memory system takes less space than the previous 1-Mbyte design, since the new bubble chip's package is both narrower, allowing more chips per board, and shorter, giving more room to stack boards next to one another (see "More Memory in Less Space"). Furthermore, the support components are interchangeable and, like the bubble chips, do not have to be matched sets, as was often true of other bubble devices. In fact, any bubble chip is guaranteed to

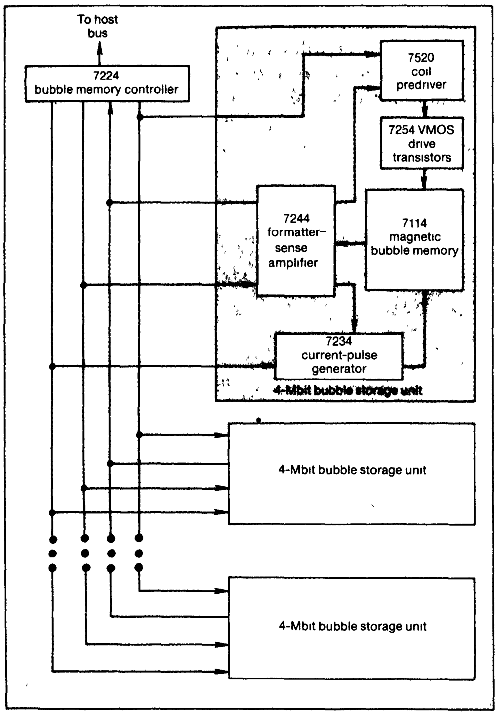


Hudson Washburn, Design Engineer
Sam Nicolino, Design Engineer
Intel Corp.
3065 Bowers Ave., Santa Clara, Calif. 95051

work with any support component, so that components can be replaced in the field without fine tuning.

Also, because the 4-Mbit bubble chip was designed to be compatible with the same hardware and software developed for the 1-Mbit version, the support circuits for both have the same pinouts. Most of the register bits are the same, too. The only differences are those in which the larger memory capacity affects how the bits are defined. Consequently, from a software perspective, any revisions to upgrade to the 4-Mbit chip are minor.

As with the 1-Mbit system, the user's interface with the 4-Mbit system remains simple. The software is written so that, first, parameters are passed to the controller by loading its registers, followed by commands. In addition, data is written or read in any of three transfer modes—DMA, polled, or interrupt—and the controller's 40-byte FIFO acts as a buffer between the host and formatter–sense amplifier chips. The formatter–sense amplifier is responsible for sending and receiving serial data



1. The key to building a 4-Mbyte bubble memory system is the ability of the bubble chip's support ICs to simplify the interface with the host. Five such ICs plus a single 4-Mbit chip (shaded) form the basic memory block. Up to seven additional blocks in parallel, all governed by one memory controller chip, complete the system.

between the bubble and the controller. The host system therefore need only monitor the controller's status register to determine when it is busy and to see if a transfer operation was successful.

The bubble memory controller is the bubble chip's link to the host. It communicates with the host over an 8-bit bidirectional data bus; a single address line (A_0); and a chip-selection, a read and a write control, and an interrupt line. In addition, a ninth data bit line (D_8) can be used to detect parity errors.

The remaining input and output lines of the controller connect the formatter–sense amplifier, the coil predriver, and the current-pulse generator. These components, plus a pair of VMOS drive transistor chips, make up a 4-Mbit bubble storage unit (Fig. 1). Up to eight such units may be connected to a single controller, allowing users to trade off the number of pages against the individual page size to fit their data transfer requirements.

The controller close up

To understand the software and hardware interface with the bubble subsystem requires and understanding of the controller. An HMOS chip, it is housed in a 40-pin DIP and divided into 10 functional blocks (Fig. 2).

The host processor operates the bubble memory system by reading from, or writing to, specific registers within the bubble memory controller. The host selects each register by placing an address on lines A_0 and D_0 through D_4. Specifically, the status register and command register are directly addressed using these six bits; a third register, the register address counter, is also directly addressed and in turn indirectly addresses the remaining registers, including the block-length register, the FIFO data buffer, and the enable register. These remaining registers are called parametric registers because they contain the flags and parameters that determine exactly how the controller will respond to commands written in the command register. The parametric registers are located in a register file and are selected with addresses 1011 through 1111. In general, the parametric registers must be loaded before commands are issued to the controller.

Parametric registers are loaded when they are addressed by the register address counter. The controller automatically increments the counter by one after each data transfer between the host and a parametric register. Thus there is no need to reload the address register in the case of multiple register reads and writes.

The address register increments, starting with the address first loaded, until it reaches binary address 1111. It then wraps around to 0000 and halts until it is reloaded with another address. However, when line A_0 is zero, all data transfers are with the FIFO. In addition, any other commands or a controlled stop sequence will reset the address counter to 0000, which is the FIFO address.

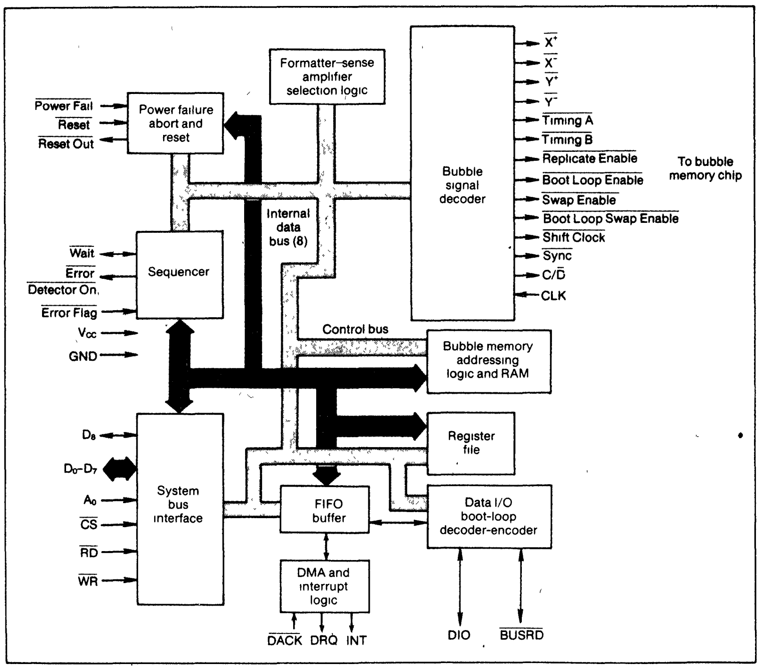
The most commonly used commands (see the table) are Initialize, Read Bubble Data, and Write Bubble Data. Others used in a typical operation are Read Seek, Write Seek, Read Formatter–Sense Amp Status, and Reset FIFO. In addition, two commands —Zero Access Read Seek and Zero Access Read Bubble Data—slash the data access time by a factor of more than 150. Zero Access Read Bubble Data returns the first byte of data in the FIFO within 50 μs after the command is sent, provided the address is known in advance of the access command.

Parameters first

Commands are written by the host into an 8-bit write-once command register. Depending on the command, certain parameters must already be written into their respective registers. For example, the Initialize command must be preceded by the number of formatter—sense amplifiers in the block-length register's first four MSB locations (Fig. 3a). Similarly, before issuing a Read Bubble Data command, the starting address information must already be set in the address register (Fig. 3b), as must be the number of system pages in the block-length register. Thus each command has its specific set of parametric requirements that must be established before it is issued.

If the parametric conditions have been set, the command is issued using a 5-bit command code. For example, Initialize is 00001, Read Bubble Data is 00010, and so on.

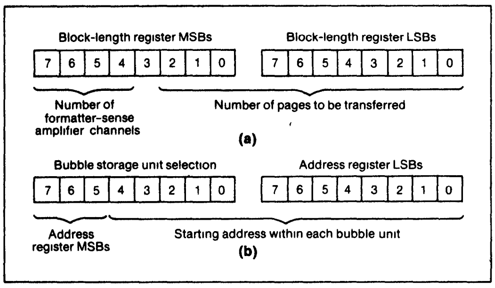
Information about any error condition, the completion or termination of a command, or the controller's readiness is stored in the status register. The host can directly address this register by setting the A_0 line and examining the eight status flags. The status register is updated every microsecond. Bits 1 through 6 (Fig. 4a) are set during command



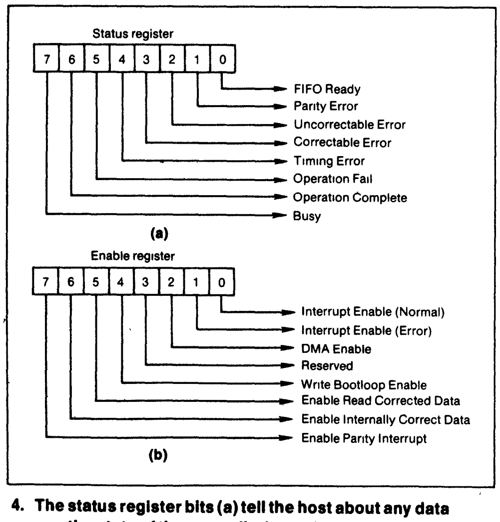
2. The 7224 bubble memory controller interfaces the bubble storage units with the host processor. It performs 10 functions, each represented by a block. The host is connected to an 8-bit data bus with an optional parity bit, a single address line, a chip-selection line, and a read and a write control line. Interrupt and DMA handshaking also are available.

execution and are reset when a new command is issued. The flags in the status register indicate whether the controller is executing a command or has completed one. In addition, they show whether an uncorrectable error or a timing error has occurred. Also, using a parity bit, the controller checks the data the host sends it and generates an odd parity for the data it sends to the host. Any parity errors are flagged.

The system page size and the number of pages to be transferred in response to a single bubble memory data read or write command are set by the block-length register, a 16-bit write-once register. The system page size is proportional to the number of bubble storage units operating in parallel during a data read or write operation. Each bubble chip requires two formatter–sense amplifier channels, with bits 4 through 7 specifying the number of such channels to be accessed. For example, in a 4-Mbyte system, if bits 7 to 4 are 0001, two channels will be accessed, each page will contain 512 bits, and there will be 65,172 pages. Setting the bits to 0100 specifies eight channels, 2048 bits per page, and 16,384 pages.



3. The parametric registers set the basic conditions for transfers between the host and the bubble memory system. The block-length register gives the number of formatter–sense amplifier channels and the number of system pages in a block (a). The address register gives the starting address for a read or write command (b).



4. The status register bits (a) tell the host about any data errors, the state of the controller's readiness, or whether a command was completed properly or not. The register is updated every microsecond and indicates whether a data error was correctable or not, in addition to pointing out parity and timing errors. The enable register bits (b) specify several conditions, including interruption on an error, DMA enabling, and parity error interruption.

The right address

Which bubble memory group is accessed and what the starting address location is within that group are determined by the contents of the address register. Each bubble chip has 8192 address locations for reading or writing data. Consequently, 13 bits are needed to specify an individual bubble storage unit's starting address. Which of the units to be read from or written to is indicated by address register bits 5 through 7. How the controller interprets these bits depends on the number of bubble storage units in a group as specified by the block-length register. For example, if the formatter–sense amplifier channels are numbered 0 through F_{16} and the number of formatter channel bits of the block-length register are set at 0000, the address register bits will specify channels 0 through 7. If, on the other hand, the block-length register bits are in the sequence 0001, the address register bits select the formatter—sense amplifier channel pairs and address register bits 0110 select channels C and D.

The address range for a 4-Mbyte subsystem is 0000–FFFF, or 65,172 pages. Selecting address register bits 0111 puts the data in the last 8192 pages of bubble storage.

Enable register controls

Certain functions in the formatter–sense amplifier and the controller are governed by setting bits in the enable register (Fig. 4b). For example, setting the Enable Parity Interrupt stops the host when the controller detects a parity error on the data bus lines (D_0-D_7). Also, the controller operates in a DMA data transfer mode when the DMA Enable bit is set. In this mode the Data Request and Data Acknowledge interface signals become operational; otherwise, the controller supports interrupt-driven or polled data transfer modes. As a result, users have a choice of three data transfer methods.

The Interrupt Enable (Normal) bit, when set to a 1, allows the controller to interrupt the host system when a command is successfully executed. The Interrupt Enable (Error) bit works in conjunction with

Bubbles by the block

The basic technology of the 7114 4-Mbit bubble chip—known as field access, conductor-first permalloy—is the same as used to build the earlier 7110, a 1-Mbit part, except for several important refinements. These refinements quadruple the bit density and the data transfer rate.

The increased density is produced by halving the period of the basic memory cell (called an asymmetric propagator) to 5.5 μm. The resultant chip size is 501 by 580 mils (compared with the 1-Mbit's 512 by 614 mils). A 0.75-μm minimum feature size, smaller than that of any silicon chip, is being printed now in development volumes using optical contact lithography. However, X-ray lithography techniques will be used for production volumes to achieve repeatible results despite the small minimum-feature size.

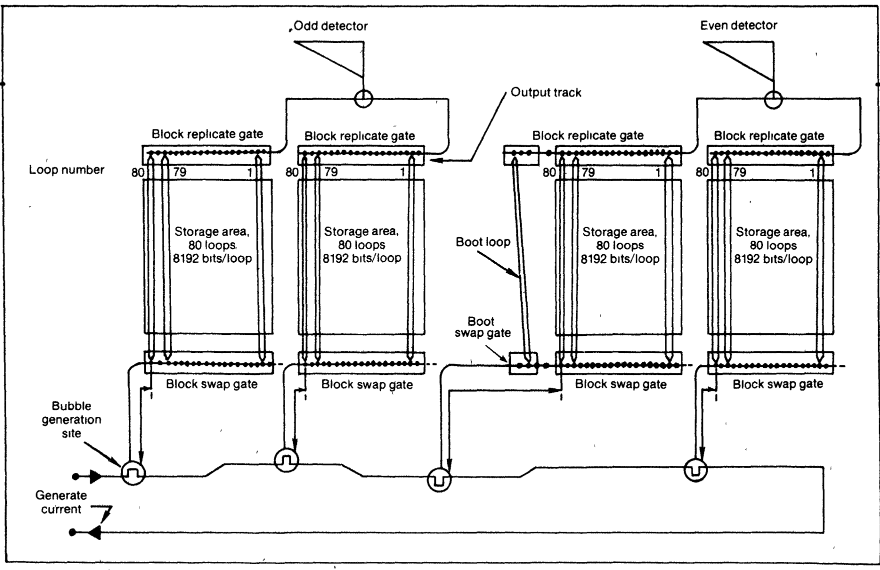
In addition, a thin-film detector was developed that doubles the detected bubble signal compared with the previous thick-film detectors. This makes doubling the data rate feasible. Further, doubling the field rotation rate from 50 to 100 kHz also doubled the data rate, producing the overall 400% increase, which also means an average random access time of 40 ms. (A 50-kHz version will be introduced first that has twice the data rate of the 1-Mbit chip and an 80-ms access time.)

Like the technology, the architecture of the 4-Mbit chip is an enhanced version of the 1-Mbit design. Both use block-swapping and replicating schemes to write and read bubbles in parallel, to ensure nonvolatile storage, and to permit the use of multiplexed replication generators to reduce the number of external pins.

The page length is fixed at 512 bits (64 bytes), but the number of pages has been quadrupled for the 4-Mbit part. Both chips are organized into identical halves. Thus, from an architectural perspective, the higher-density chip looks like a 1-Mbit part with four times the number of pages and either twice (50 kHz) or four times (100 kHz) the data rate.

Actually, the 7114 is divided into eight octants, each comprising 80 minor loops, and each loop containing 8192 bits (see the figure). The 7110, in comparison, is split into four quadrants, each with 80 minor loops, but each loop contains only 4096 bits. Also, whereas the 7110 was designed to sense one bit per side per field rotation, the 7114 senses two bits. In the 50-kHz 4-Mbit part, the longer loops are compensated for by the two-bit-per-rotation sensing.

Like the 1-Mbit device, the 4-Mbit chip has redundant loops to ensure a high yield of devices with the full 4,194,304 bits of storage capacity. Redundancy increases yields and so lowers device cost. During manufacture, each device is individually tested and a record of faulty loop locations is written and stored in the device's bootstrap loop, known as the "boot loop." The boot loop's contents are used by the 7224 bubble memory controller during initialization, reading, and writing to provide a full 4-Mbit memory space to the user while keeping redundant loops invisible. The major-track, minor-loop architecture used by both the 7114 and the 7110 to accomplish the writing, reading, and nonvolatile storage of data also maintains the reliability inherent in bubble technology.



the other enable register bits to support three levels of error correction.

At the first level, setting Enable Internally Correct Data causes the controller to send a command to a formatter–sense amplifier when an error has been detected. The formatter–sense amplifier responds by internally cycling the data through its error-correction network. On completion, it sends its status to the controller, indicating whether or not the error was corrected.

For the second level, the Enable Read Corrected Data bit prompts the controller to issue a command to the appropriate formatter-sense amplifier when an error has been detected. The formatter—sense amplifier then corrects the error if possible and transfers the corrected data to the controller. When the data transfer is complete, the controller reads the formatter–sense amplifier's status to determine whether the error was corrected. Otherwise, faulty data could be transferred to the controller and possibly to the host.

Lastly, setting the Write Bootloop Enable bit permits writing into the bootstrap loop, called here just the "boot loop." Normally, the loop should only be read, but under special circumstances a user may wish to write into it.

Bubble controller command codes

D_4	D_3	D_2	D_1	D_0	Command name
0	0	0	0	0	Write—Boot Loop Register Masked
0	0	0	0	1	Initialize
0	0	0	1	0	Read Bubble Data
0	0	0	1	1	Write Bubble Data
0	0	1	0	0	Read Seek
0	0	1	0	1	Read Boot Loop Register
0	0	1	1	0	Write Boot Loop Register
0	0	1	1	1	Write Boot Loop
0	1	0	0	0	Read Formatter-Sense Amp Status
0	1	0	0	1	Abort
0	1	0	1	0	Write Seek
0	1	0	1	1	Read Boot Loop
0	1	1	0	0	Read Corrected data
0	1	1	0	1	Reset FIFO
0	1	1	1	0	Memory Unit Purge
0	1	1	1	1	Software Reset
1	0	0	1	0	Zero Access Read Bubble Data
1	0	1	0	0	Zero Access Read Seek

More memory in less space

Instead of a leadless package requiring a second, leaded socket, the 7114 4-Mbit bubble chip is housed in a leaded package that can be placed in a socket or soldered directly to a PC board. Like the 1-Mbit package, it has 20 pins. However, the distance between pin rows is smaller, making the footprint smaller and allowing designers to incorporate more components onto the board. Also because the package's height is smaller, boards can be spaced as close as 0.6 in. to one another. Thus consequently, either more boards can be accommodated or the overall system size can be made smaller. As a result, a 4-Mbyte bubble memory system can be built in less space than a 1-Mbyte bubble system.

The FIFO as a data buffer

All data moving between the host and the bubble units passes through the 40-byte FIFO buffer. As a result, the data transfer is asynchronous, with timing constraints relaxed somewhat for both the formatter–sense amplifier and the host system. When the controller is busy executing a command, the FIFO functions as a data buffer; however, when the controller is not busy, the FIFO is available to the host as a general-purpose FIFO register bank.

Actually, a total of 43 bytes of data may be stored in the controller: 40 bytes in the FIFO, 1 byte each in its input and output latch, and 1 byte in the controller's input latch. During execution of a command involving a data transfer between the host and the formatter–sense amplifiers, the data passes through the FIFO and its status is indicated by the FIFO Ready bit in the storage register.

The FIFO is addressed automatically after the last parametric register has been written into; alternatively, the host can explicitly address the FIFO by writing the address 0000 into the register address counter. Also, after a Write Bubble Data, a Write Boot-Loop Register, or a Write Boot-Loop Register Masked command is issued, the controller delays the data transfer until there are at least two bytes of data in the FIFO. Furthermore, it is the host system's responsibility to keep up with the data transfer during execution of a command; otherwise the FIFO could underflow or overflow. If either case occurs, a Timing Error bit is set in the status register.

A look at data transfer

The boot-loop register plays a key role in data transfer both for writing and reading. This 160-bit register contains information detailing the configuration of good and bad loops in the corresponding channel of each bubble chip.

Each bit of the register corresponds to a minor loop in the bubble chip. As data passes through the latter's I/O latches, the contents of the boot-loop register are used during reading to remove the bits corresponding to bad loops and during writing the contents are used to insert 0s in those bit positions that correspond to bad loops.

Meanwhile, the error-correction block implements a 14-bit Fire code error-detection and -correction process. If it has been enabled by the user, the error-correction circuitry appends the 14-bit code to the end of each 256-bit block of data that passes through the FIFO during a data write operation. When data is being read, this circuitry checks the data block and notifies the controller with an error flag when an error has been detected.

As stated earlier, a Write Bubble Data command from the controller to the formatter–sense amplifier permits data from the controller to be written into the good loops of the memory unit. If the error correction is activated, the amplifier automatically adds the 14 error-correction bits to the end of each 256-bit data block.

Similarly, a Read Bubble Data command enables the formatter–sense amplifier to read data from the bubble chip, as was also mentioned previously. This data is sensed by the sense amplifiers and screened by the boot-loop registers so that only data from good loops is written into the FIFOs. If the error correction is selected, data to be read is first buffered. That is, a full block (270 bits) of data is collected in the FIFO before any bits are read out. As a result, the error-correction circuitry detects any errors and interrupts the controller before any data is sent. If there are no errors, the 270-bit block is read from the FIFO and sent to the controller while the next block is loaded into the FIFO.

In contrast, an Internally Correct Data sequence forces the formatter–sense amplifier to cycle the data internally through the error-correction network without sending any of it to the controller. At the end of the operation, the amplifier sets a Correctable or Uncorrectable Error bit in its status register. If the error is correctable, the controller has the option of issuing a Read Corrected Data command. This command cycles the data through the error-correction circuitry as it is being read by the controller. After all 256 bits have been transferred to the controller, the formatter–sense amplifier status register indicates whether the error was found to be correctable or not. The Read Corrected Data command is used even when the data has been previously corrected by the Internally Correct Data command. □

The authors wish to thank Dave Dossetter, Product Marketing Engineer, and Dick Pierce, Marketing Applications Engineer, for their invaluable assistance in preparing this article.

PRELIMINARY

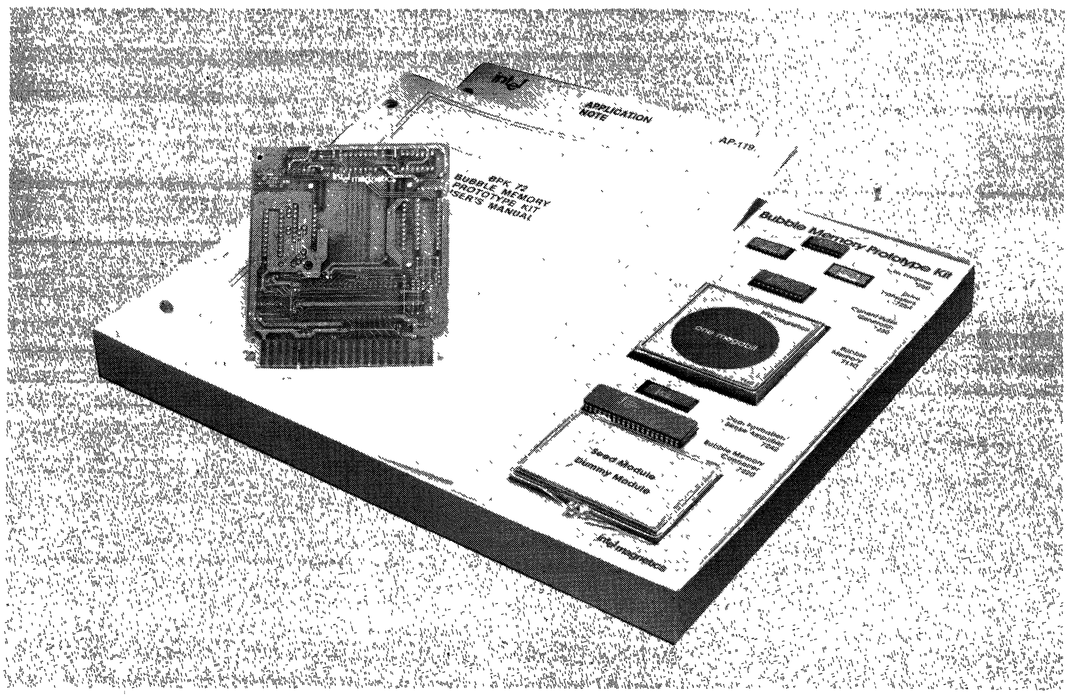
BPK 72
1 MBIT BUBBLE MEMORY PROTOTYPE KIT

BPK 72-1	0°-75° C
BPK 72-4	10°-55° C
BPK 72-5	−20°- 85° C

- **1 Mbit, Non-Volatile, Read-Write, High-Density, Bubble Storage Unit**
- **Operates from +5V and +12V Power Supplies**
- **Average Access Time of 48 ms**
- **Built-in Error Correction/Detection**
- **Complete with Components, Blank Board, Accessories and Documentation for Prototyping**
- **Powerfail Data Protection**
- **Maximum Data Rate of 100K bit/sec**
- **Compatible with 8080/85/86/88 and other Standard Microprocessors**

The BPK 72 prototype kit contains all the necessary items and documentation required to build a 1 Megabit bubble storage prototype system with a minimum of design effort. Thus this unit gives the design engineer the opportunity to learn the characteristics of a Bubble Memory System and to actually test the bubble in a prototype product. Application information on microprocessor interfacing is included in the kit.

Each of the components in the kit, i.e., 7110, 7220, 7230, 7242, 7250, 7254 are described in detail on the respective component data sheet.



NOVEMBER 1982
ORDER NUMBER: 210804-001

ORDERING INFORMATION

Part Number	Temperature 7110 Magnetic Bubble Memory		Support Circuits Min. Operating Temperature	Description
	Operating	Non-Volatile Storage		
BPK 72-1	0° to 75°C Case	−40° to 90°C	0° to 70°C Ambient	1 Mbit Bubble Memory Prototype Kit
BPK 72-4	10° to 55°C Case	−20° to 75°C	10° to 55°C Ambient	1 Mbit Bubble Memory Prototype Kit
BPK 72-5	−20° to 85°C Case	−40° to 100°C	−20° to 85°C Ambient	1 Mbit Bubble Memory Prototype Kit

BPK 72 ITEMS

Item	Description	Part Number
1 MBit Bubble Memory	20-pin package which provides 1 megabit of non-volatile storage.	7110-1/7110-4/7110-5
Socket for 7110	Provides reliable mounting and removability to printed circuit boards.	7905
Seed Module	Recreates a lost seed bubble.	7901
VMOS Transistor	7230 Reference current switch.	7902
Dummy Module	Small PC board used in place of the 7110 during initial prototyping.	7900
Bubble Memory Controller	User interface, performs serial-to-parallel and parallel-to-serial data conversions. Generates timing signals.	7220-1/7220-5
Current Pulse Generator	Converts digital timing signals to analog current pulses suited to the drive requirements of the 7110 MBM. The CPG provides the replicate, swap, generate, boot replicate, and bootswap pulses required by the MBM.	7230/7230-4/7230-5
Dual Formatter/Sense Amp	Provides direct interface to the 7110 Bubble Memory. The FSA contains on-chip sense amplifiers, a full FIFO data block buffer, burst error detection and correction circuits, and circutiry for handling of the bubble memory redundant loops.	7242
Coil Predriver	Provides the high voltage, high current outputs to drive the 7254 Quad VMOS transistors.	7250
2 Quad VMOS Coil Drive Transistors	Switches the required current to drive the X and Y coils of the 7110 Bubble Memory.	7254
Prefabricated Printed Circuit Board		IMB 72
BPK 72 Bubble Memory Prototype Kit User's Manual	Literature	121685-002
Microprocessor Interface for the BPK 72 (AP-119)	Literature	210367

SPECIFICATIONS

Capacity

128K Byte per BPK 72

Performance

Avg. Access Time .... 48 msec
Maximum Data Transfer Rate ...... 100 Kbits/sec
Average Data Transfer Rate .... 68 Kbits/sec

Data Organization

512 bits per page
2048 pages per BPK 70

Addressing Scheme

Logical page number

Environmental

Temperature: See Ordering Information
Operating Humidity: 0–95% Non-Condensing

BPK 72 POWER SUPPLY REQUIREMENTS

Voltage	Margin	Power Off/Power Fail Decay Rate
+12 Volt	±5%	less than 1.10 volts/msec
+5 Volt	±5%	less than 0.45 volts/msec

- Voltage sequencing—no restrictions
- Power on voltage rate of rise—no restrictions

BPK 72 POWER CONSUMPTION

BPK 72 KIT

Power (Watts)					
+5V (Maximum)	+12V (Maximum)	Total Active (Maximum)	Total Active (Typical)	Total Standby (Maximum)	Total Standby (Typical)
1.92	4.80	6.72	3.90	3.03	1.55

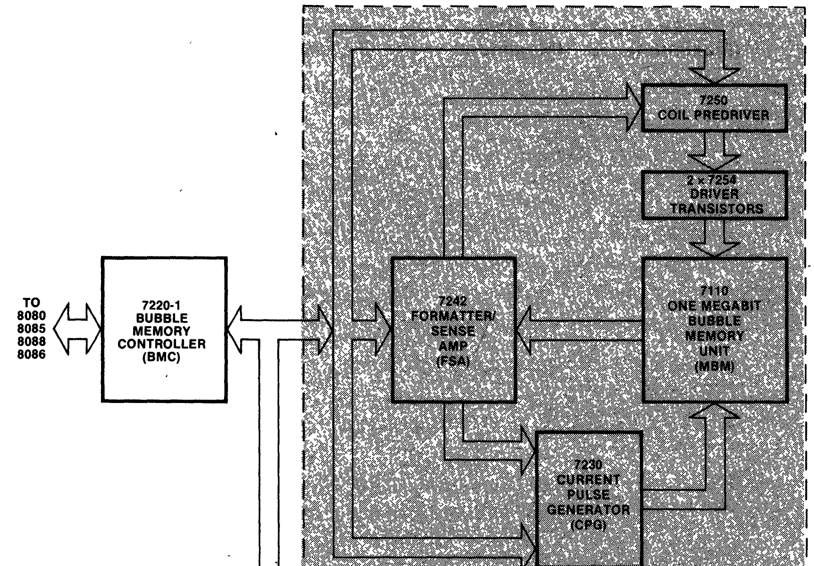
BPK 70
1 MBIT BUBBLE MEMORY SUBSYSTEMS

BPK 70-1	0°-75° C
BPK 70-4	10°-55° C
BPK 70-5	−20°- 85° C

- **1 MBit, Non-Volatile, Read-Write, High-Density Bubble Memory Subsystems**
- **Average Access Time of 48 ms**
- **Operates from +5V and +12V Power Supplies**
- **Maximum Data Rate of 100 KBit/Sec**

A Bubble Storage Subsystem contains components for production of 1 MBit Bubble Storage System. The kit consists of one 1 MBit Magnetic Bubble Memory and five support circuits (shown in the figure below). The BPK 70 Subsystem is controlled by an additional 7220 Bubble Memory Controller. One 7220-1 is capable of controlling up to eight BPK 70-1s or BPK 70-4s and one 7220-5 is capable of controlling up to four BPK 70-5s. Larger systems may be built using multiple 7220's with additional Bubble Storage Subsystems. The user interface of the 7220 is compatible with microprocessor bus systems for 8080, 8085, 8086 and 8088 and other standard microprocessors.

For applications in the 0–75°C and 10–55°C temperature range, the bubble Memory (7110-1/7110-4) and the other support circuits (7230, 7242, 7250, 7254) are available as separate, interchangeable components. Each of the components in the Subsystem are described in detail on the respective component data sheets.



CONFIGURATION OF ONE BPK 70 BUBBLE STORAGE SUBSYSTEM WITH THE 7220 CONTROLLER

OCTOBER 1982
ORDER NUMBER: 210356-002

ORDERING INFORMATION

Part Number	Temperature 7110 Magnetic Bubble Memory		Support Circuits Min. Operating Temperature	Description
	Operating	Non-Volatile Storage		
BPK 70-1	0° to 75°C Case	−40° to 90°C	0° to 70°C Ambient	1 Mbit Bubble Storage Sub-System
BPK 70-4	10° to 55°C Case	−20° to 75°C	10° to 55°C Ambient	1 Mbit Bubble Storage Sub-System
BPK 70-5	−20° to 85°C Case	−40° to 100°C	−20° to 85°C Ambient	1 Mbit Bubble Storage Sub-System

BPK 70 ITEMS

Item	Description	Part Number
1 MBit Bubble Memory	20-pin package which provides 1 megabit of non-volatile storage.	7110-1/7110-4/7110-5
Socket for 7110-1, -4	Provides reliable mounting and removability to printed circuit boards.	7905/7904
Socket for 7110-5	Provides reliable mounting and removability to printed circuit boards.	7905
Current Pulse Generator	Converts digital timing signals to analog current pulses suited to the drive requirements of the 7110 MBM. The CPG provides the replicate, swap, generate, boot replicate, and bootswap pulses required by the MBM.	7230/7230-4/7230-5
Dual Formatter/Sense Amp	Provides direct interface to the 7110 Bubble Memory. The FSA contains on-chip sense amplifiers, a full FIFO data block buffer, burst error detection and correction circuits, and circuitry for handling of the bubble memory redundant loops.	7242
Coil Predriver	Provides the high voltage, high current outputs to drive the 7254 Quad VMOS transistors.	7250
2 Quad VMOS Coil Drive Transistors	Switches the required current to drive the X and Y coils of the 7110 Bubble Memory.	7254

SPECIFICATIONS

Capacity

128K Byte per BPK 70
Maximum 8 BPK 70-1 or 8 BPK 70-4 with one 7220-1 Controller
Maximum 4 BPK 70-5 with one 7220-5 Controller

Performance

Avg. Access Time .... 48 msec

Data Organization

512 bits per page
2048 pages per BPK 70

Addressing Scheme

Logical page number

Environmental

Temperature: See Ordering Information
Operating Humidity: 0–95% Non-Condensing

DATA TRANSFER RATES (Examples of System Configurations)

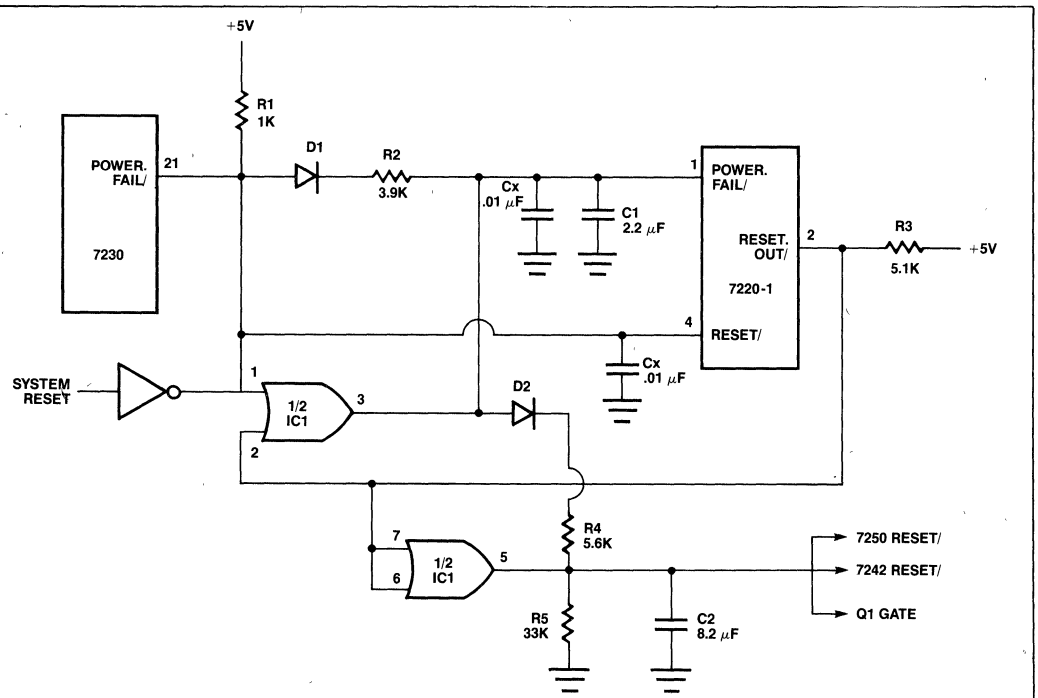
Parameter	One BPK 70 Unit	Four BPK 70 Operated in Parallel[1]	Eight BPK 70[2] Operated in Parallel[1]	Eight BPK 70[2] Multiplexed One at a Time[1]
Capacity	128 kilobytes	512 kilobytes	1 megabyte	1 megabyte
Average Data Rate (kilobits/sec)	68	272	544	68
Maximum Data Rate (kilobits/sec) (Burst)	100	400	800	100

NOTES:
1 Multiple Bubble subsystems can be operated in parallel for maximum performance or multiplexed to conserve power.
2. Only for BPK 70-1 and BPK 70-4 Systems.

BPK 70 POWER SUPPLY REQUIREMENTS

Voltage	Margin	Power Off/Power Fail Decay Rate
+12 Volt	±5%	less than 1.10 volts/msec
+5 Volt	±5%	less than 0.45 volts/msec

- Voltage sequencing—no restrictions
- Power on voltage rate of rise—no restrictions
- The power supply requirements shown are based on the recommended power fail circuitry as shown in Figure 1.



NOTES
1 ALL RESISTORS 1/4 WATT, 5% TOL
2 ALL CAPACITORS 10 VDC, 10% TOL
3 D1,D2 ARE IN914 OR EQUIVALENT
4 IC1 IN75463
5 Cx IS OPTIONAL—RECOMMENDED FOR EXTREMELY NOISY POWER SUPPLY SITUATIONS
6 SYSTEM RESET MAY BE USED AS SHOWN PROVIDED OUTPUT OF INVERTER IS OPEN COLLECTOR

Figure 1. Power Fail Circuit

BPK 70 POWER CONSUMPTION

BPK 70 Components	+5V (Maximum)	+12V (Maximum)	Total Active (Maximum)	Total Active (Typical)	Total Standby (Maximum)	Total Standby (Typical)
	Power (Watts)					
7110	0	1.740	1.740	1.480	0.440	0.290
7230	0.235	0.440	0.675	0 390	0.475	0.225
7242	0.630	0.375	1.005	0.500	1.005	0.500
7250	0	0.945	0.945	0.480	0.060	0.030
7254	0	1.300	1.300	0.550	0	0
Controller (not included in BPK 70)						
7220	1.050	0	1.050	0.500	1.050	0.500
System (Several BPK 70s operate in parallel)						
1 7220-1/-5 and 1 BPK 70-1/-4/-5	1.92	4.80	6.72	3.90	3.03	1.55
1 7220-1/-5 and 2 BPK 70-1/-4/-5	2.79	9.60	12.39	7.30	4.57	2.60
1 7220-1/-5 and 3 BPK 70-1/-4/-5	3.65	14.40	18.05	10.70	6.11	3.65
1 7220-1/-5 and 4 BPK 70-1/-4/-5	4.52	19.20	23.72	14.10	7.65	4.70
1 7220-1 and 5 BPK 70-1/-4	5.38	24.00	29 38	17.50	9.19	5 75
1 7220-1 and 6 BPK 70-1/-4	6.25	28.80	35.05	20 90	10.73	6.80
1 7220-1 and 7 BPK 70-1/-4	7.11	33.60	40 71	24 30	12.27	7.85
1 7220-1 and 8 BPK 70-1/-4	7.98	38.40	46.38	27.70	13.81	8.90

Lower power consumption with lower data transfer rates possible with multiplexed BPK 70s. See Data Transfer Rates.

intel

PRELIMINARY

7110
1-MEGABIT BUBBLE MEMORY

Device	Case Op. Temp. °C	Non-Volatile Storage °C
7110-1	0–75°	−40 to +90°
7110-4	10–55°	−20 to +75°
7110-5	−20 to +85°	−40 to +100°

- **1,048,576 Bits of Usable Data Storage**
- **Non-Volatile, Solid-State Memory**
- **True Binary Organization: 512-Bit Page and 2048 Pages**
- **Major Track–Minor Loop Architecture**
- **Redundant Loops with On-Chip Loop Map and Index**
- **Block Replicate for Read; Block Swap for Write**
- **Single-Chip 20-Pin, Dual In-Line Leadless Package and Socket**
- **Small Physical Volume**
- **Low Power per Bit**
- **Maximum Data Rate 100 Kbit/sec**
- **Average Access Time 40 msec.**

The Intel Magnetics 7110 is a very high-density 1-megabit, non-volatile, solid-state memory utilizing magnetic bubble technology. The usable data storage capacity is 1,048,576 bits. The defect-tolerant design incorporates redundant storage loops. The gross capacity of Intel Magnetics bubble memory is 1,310,720 bits.

The 7110 has a true binary organization to simplify system design, interfacing, and system software. The device is organized as 256 data storage loops each having 4096 storage bits. When used with Intel Magnetics complete family of support electronics, the resultant minimum system is configured as 128K bytes of usable data storage. The support circuits also provide automatic error correction and transparent handling of redundant loops.

The 7110 has a major track–minor loop architecture. It has separate read and write tracks. Logically, the data is organized as a 512-bit page with a total of 2048 pages. The redundant loop information is stored on-chip in the bootstrap loop along with an index address code. When power is disconnected, the 7110 retains the data stored and the bubble memory system is restarted when power is restored via the support electronics under software control.

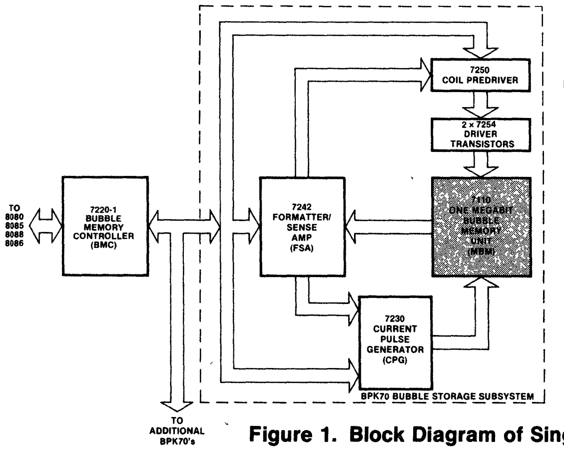
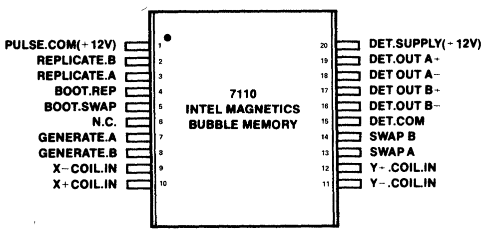


Figure 1. Block Diagram of Single Bubble Memory System—128K Bytes



NOTE THAT PINS 13 AND 14 SHOULD BE EXTERNALLY CONNECTED.

Figure 2. Pin Configuration

OCTOBER 1982
ORDER NUMBER: 210797-001

Table 1. 7110 Pin Description

Symbol	Pin	Name and Function
BOOT.REP	4	Two-level current pulse input for reading the boot loop.
BOOT.SWAP	5	Single-level current pulse for writing data into the boot loop. This pin is normally used only in the manufacture of the MBM.
DET.COM	15	Ground return for the detector bridge.
DET.OUT	16–19	Differential pair (A+, A− and B+, B−) outputs which have signals of several millivolts peak amplitude.
DET.SUPPLY	20	+12 volt supply pin.
GEN.A and GEN.B	7, 8	Two-level current pulses for writing data onto the input track.
PULSE.COM	1	+12 volt supply pin.
REP.A and REP.B	3, 2	Two-level current pulses for replicating data from storage loops to output track.
SWAP.A and SWAP.B	13, 14	Single-level current pulse for swapping data from input track to storage loops.
X−.COIL.IN, X+.COIL.IN	9, 10	Terminals for the X or inner coil.
Y−.COIL.IN, Y+.COIL.IN	11, 12	Terminals for the Y or outer coil.

The 7110 is packaged in a dual in-line leadless package complete with permanent magnets and coils for the in-plane rotating field. In addition, the 7110 has a magnetic shield surrounding the bubble memory chip to protect the data from externally induced magnetic fields.

The 7110 operating data rate is 100 Kbit/sec. The 7110 can be operated asynchronously and has start/stop capability.

FUNCTIONAL DESCRIPTION

The Intel Magnetics 7110 is a 1-megabit bubble memory module organized as two identical 512K binary half sections. See Major Track–Minor Loop architecture diagram. Each half section is in turn organized as two 256K subsections referred to as *quads*.

The module consists of a bubble die mounted in a substrate that accommodates two orthogonal drive coils that surround the die. The drive coils produce a rotating magnetic field in the plane of the die when they are excited by 90° phase-shifted triangular current waveforms. The rotating in-plane field is responsible for bubble propagation. One drive field rotation propagates all bubbles in the device one storage location (or cycle). The die-substrate-coil subassembly is enclosed in a package consisting of permanent magnets and a shield. The shield serves as a flux return path for the permanent magnets in addition to isolating the device from stray magnetic fields. The permenent magnets produce a bias field that is nearly perpendicular to the plane of the die. This field supports the existence of the bubble domains.

The package is constructed to maintain a 2.5 degree tilt between the plane of the bias magnet faces and the plane of the die. This serves to introduce a small component of the bias field into the plane of the die. During operation when the drive coils are energized, this small in-plane component is negligible. During standby or when power is removed, the small in-plane field ensures that the bubbles will be confined to their appropriate storage locations. The direction of the in-plane field introduced by the package tilt (holding field) is coincident with the 0° phase direction of the drive field.

Quad Architecture

A 7110 quad subsection is composed of the following elements shown on the architecture diagram.

1) Storage Loops
   Eighty identical 4096-bit storage loops provide a total maximum capacity of 327,680 bits. The excess storage is provided for two purposes: a) it allows a redundancy scheme to increase device yield; and b) it provides the extra storage required to implement error correction.

2) Replicating Generator (GEN)
   The generator operates by replicating a seed

bubble that is always present at the generator site, (GEN).

3) Input Track and Swap Gate
Bubbles following generation are propagated down an input track. Bubbles are transferred to/from the input track from/to the 80 storage loops via series-connected swap gates spaced every four propagation cycles along the track. The swap gate's ability to transfer bubbles in both directions during an operation eliminates the overhead associated with removing old data from the loops before new data can be written. The swap gate is designed to function such that the logical storage loop position occupied by the bubble transferred out of each loop is filled by the bubble being transferred into each loop. Transferred-out bubbles propagate down the remaining portion of the input track where they are dumped into a bubble bucket guard rail.

4) Output Track and Replicate Gate
Bubbles are read out of the storage loops in a nondestructive fashion via a set of replicate gates. The bubble is split in two. The leading bubble is retained in the storage loop and the trailing bubble is transferred onto the output track. Replicate gates are spaced every four propagation cycles along the output track.

5) Detector
Bubbles, following replication, are propagated along the output track to a detector that operates on the magneto-resistance principle. The cylindrical bubble domains are stretched into long strip domains by a chevron expander and are then propagated to the active portion of the detector. The detector consists of a stack of interconnected chevrons through which a current is passed. As the strip domain propagates through the stack, its magnetic flux causes a fractional change in stack resistance which produces an output signal on the order of a few millivolts. The strip domain following detection is propagated to a bubble bucket guard rail. A "dummy" detector stack sits in the immediate vicinity. It is connected in series with the active detector and serves to cancel common mode pickup which originates predominately from the in-plane drive field.

6) Boot Loop, Boot Swap, and Boot Replicate
One of the two quads in each half chip contains a functionally active Boot Storage Loop. This loop is used to store:

a) A loop mask code that defines which loops within the main storage area should be accessed. Faulty loops are "masked out" by the support electronics.

b) A synchronization code that assigns data addresses (pages) to the data in the storage loops. Since bubbles move from one storage location to the next every field rotation, the actual physical location of a page of data is determined by the number of field rotations that have elapsed with respect to a reference.

The boot loop is read from and written into via the same input and output tracks as the main storage loops. However, it has independently accessed swap and replicate gates. The boot swap, under normal circumstances, is intended only to be used during basic initialization at the factory at which time loop mask and synchronization codes are written. The boot replicate is intended to be accessed every time power is applied to the bubble module and its peripheral control electronics. At such a time, the control electronics would read and store the mask information, plus utilize the synchronization information to establish the location of the data circulating within the loops.

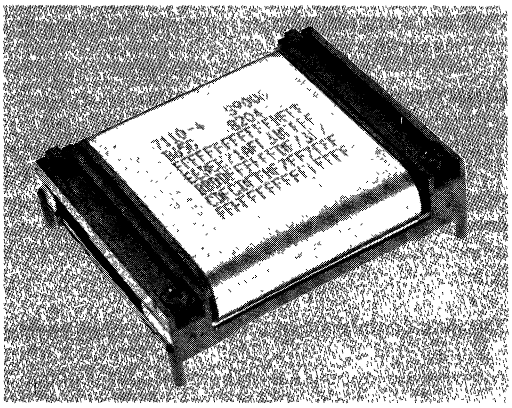


Photo 1. 7110 Package Seated in Socket

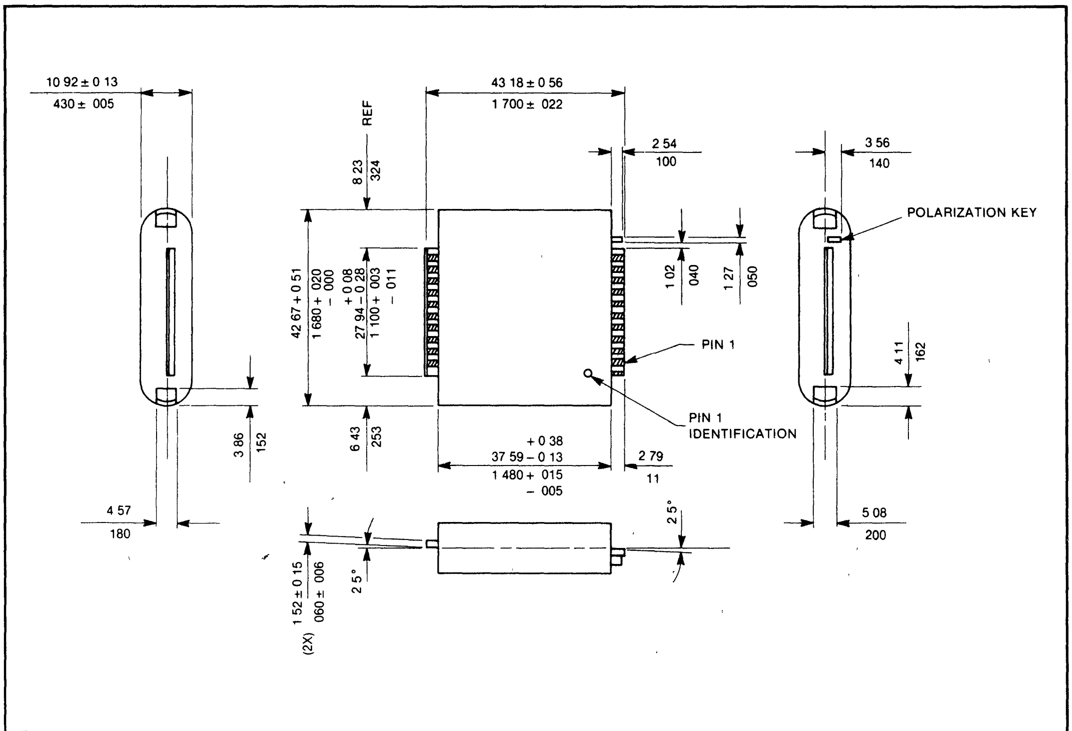
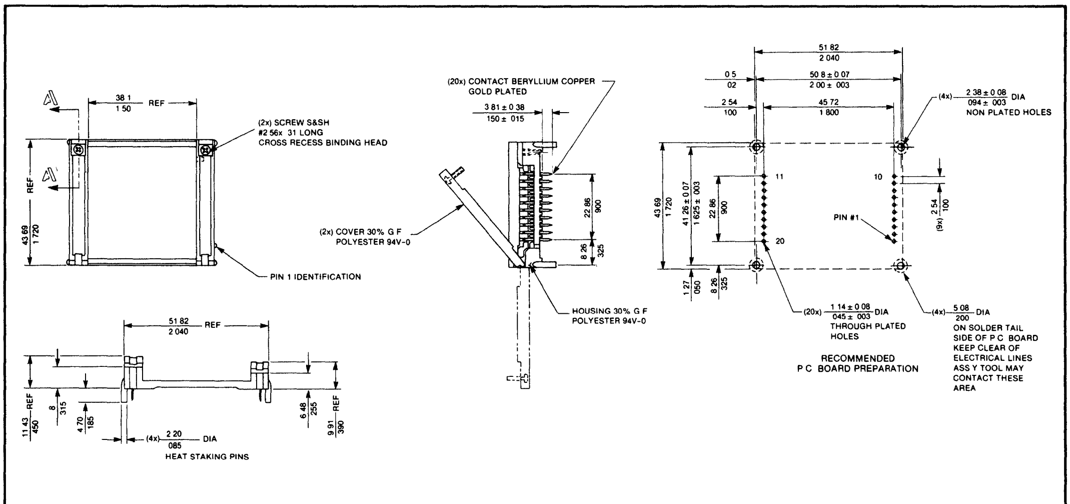


Figure 3. Package Outline



NOTE:
SEE PACKAGING INFORMATION, 1983 MEMORY CATALOG FOR DETAILS ON ALTERNATIVE SOCKETS

Figure 4. Socket Outline

AFN-01483B

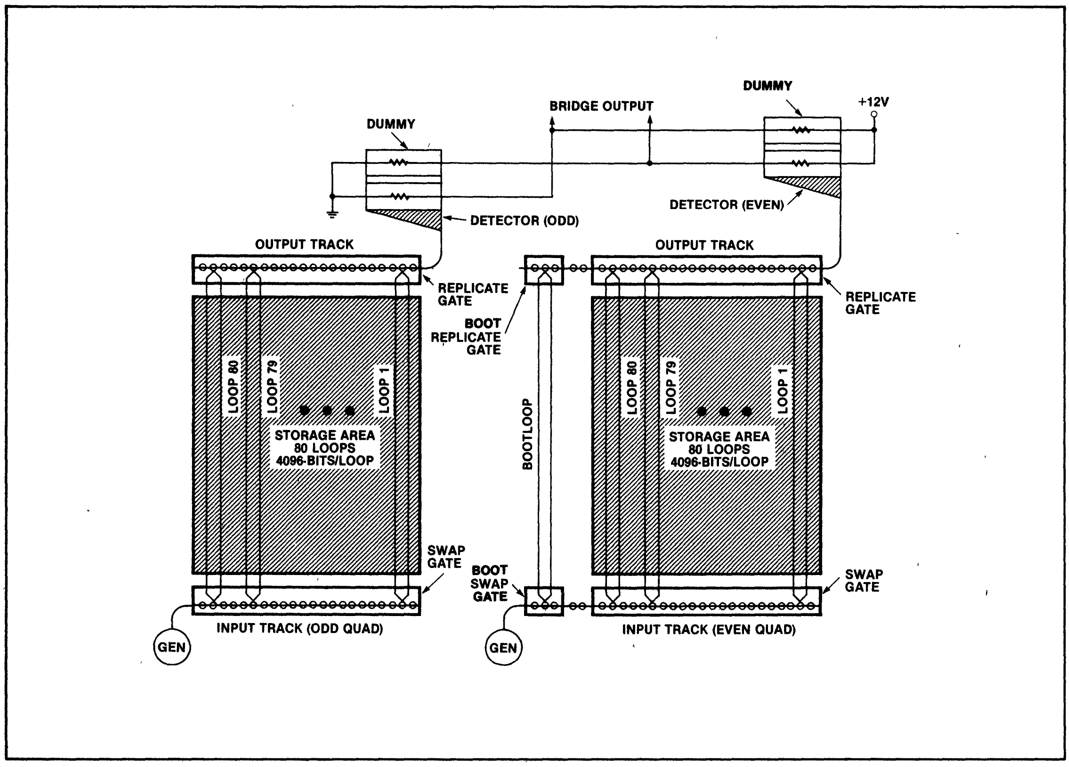


Figure 5. Major Track–Minor Loop Architecture of 7110 (one half shown)

ABSOLUTE MAXIMUM RATINGS*

Operating Temperature .... −20°C to +85°C Case
Relative Humidity 95%
Shelf Storage Temperature (Data
Integrity Not Guaranteed) .... −65°C to +150°C
Voltage Applied to DET.SUPPLY .... 14 Volts
Voltage Applied to PULSE.COM .... 12.6 Volts
Continuous Current between DET.COM and
Detector Outputs .... 10 mA
Coil Current .... 0.5A D.C.
External Magnetic Field for
Non-Volatile Storage .... 20 Oersteds
Non-Operating Handling Shock
(without socket) 200G
Operating Vibration (2 Hz to 2 kHz
with socket) 20G

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

D.C. CHARACTERISTICS (T_C = Range Specified on first page. V_{DD} = 12V ± 5%)

Parameter	7110-1, -4 Limits			7110-5 Limits[5]		Unit
	Min.	Nom.[1]	Max.	Min.	Max.	
RESISTANCE: PULSE.COM to GEN.A or GEN.B	9	30	59	8	61.5	ohms
RESISTANCE: PULSE.COM to REP.A or REP.B	9	20	26	8	27	ohms
RESISTANCE: PULSE.COM to SWAP.A or SWAP.B	44	100	149	40	155.5	ohms
RESISTANCE: PULSE.COM to BOOT.REP	3.5	8	24	3	25	ohms
RESISTANCE: PULSE.COM to BOOT.SWAP	5	15	36	4.5	37.5	ohms
RESISTANCE: DET.OUT A+ to DET.OUT.A−	670	1030	1903	620	1984	ohms
RESISTANCE: DET.OUT B+ to DET.OUT B−	670	1030	1903	620	1984	ohms
RESISTANCE: DET.COM to DET.SUPPLY	355	600	1050	338	1095	ohms
X.COIL RESISTANCE		4.6		329		ohms
Y.COIL RESISTANCE		2.0				ohms
X.COIL INDUCTANCE		97				μH
Y.COIL INDUCTANCE		80				μH
OPERATING POWER		1.20	1.75			watts
STANDBY POWER		0.25	.45			watts

DRIVE REQUIREMENTS CHARACTERISTICS[2] (T_C = Range Specified on first page.)

Symbol	Parameter	Min.	Nom.[1]	Max.	Units
f_R	Field Rotation Frequency	49.95	50.000	50.05	kHz
I_{px}	X.Coil Peak Current		600		ma
I_{py}	Y.Coil Peak Current		750		ma
Θ_{1x}	X.Coil Positive Turn-On Phase	268	270	272	degrees
Θ_{2x}	X.Coil Positive Turn-Off Phase	16	18	20	degrees
Θ_{3x}	X.Coil Negative Turn-On Phase	88	90	92	degrees
Θ_{4x}	X.Coil Negative Turn-Off Phase	196	198	200	
Θ_{1y}	Y.Coil Positive Turn-On Phase	0	0	0	degrees
Θ_{2y}	Y.Coil Positive Turn-Off Phase	106	108	110	degrees
Θ_{3y}	Y.Coil Negative Turn-On Phase	178	180	182	degrees
Θ_{4y}	Y.Coil Negative Turn-Off Phase	286	288	290	degrees

CONTROL PULSE REQUIREMENTS (T_C = range specified on first page)[5]

	Amplitude			Pulse of Leading Edge (Degrees)[3]			Width (Degrees)[3]		
Pulse	Min.	Nom[1]	Max.	Min.	Nom.[1]	Max.	Min.	Nom.[1]	Max.
GEN.A, GEN.B CUT	62	75	81	266 86	270 (Odd) 90 (Even)	274 94	3	6.75	8
GEN.A, GEN.B TRANSFER	34	40	49	266 86	270 (Odd) 90 (Even)	274 94	86	90	94
REP.A, REP.B CUT	170	200	240	268	270	277	3	6.75	8
REP.A, REP.B TRANSFER	126	145	160	268	270	277	86	90	94
SWAP	111	125	134	176	180	184	513	517	521
BOOT.REP CUT	85	100	110	268	270	277	3	6.75	8
BOOT.REP TRANSFER	63	75	80	268	270	277	86	90	94
BOOT.SWAP[4]	63	75	80	176	180	184		360	

NOTES:

1. Nominal values are measured at T_C = 25°C.
2. See Fig 6 for test setup and X-Y coil waveforms.
3. Pulse timing is given in terms of the pulse relations as shown in Figure 7. For example, a 7110 operating at f_R=50 kHz would have a REP.A transfer width of 90° which is 5 μs.
4. Boot.Swap is not normally accessed during operation. It is utilitized at the factory to write the index address and redundant loop information onto the bootstrap loops before shipment
5. 7110-5 is sold only as a matched part with the 7230-5. Matched parts are tested over temperature range for V_{DD} = 12V ±5%.

OUTPUT CHARACTERISTICS

Symbol	Min.[2]	Nom.[1]	Max.[2]	Units	Test Conditions
S_1	2.7	6		mV	See notes 1, 2,3
S_0		1	2.3	mV	

NOTES:

1. Nominal values are measured at $T_C = 25°C$
2. Min./Max. values for S_1/S_0 are measured at worst case conditions and tested to a system error rate of 10^{-9} when used with the 7242 formatter sense amplifier without ECC enabled
3. See Fig 8 for test setup, and Fig 9 for detector output waveforms and timing.

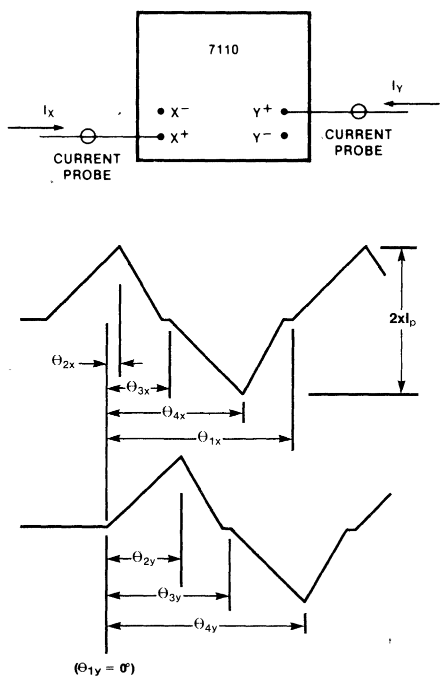


Figure 6. X-Y Coil Waveforms

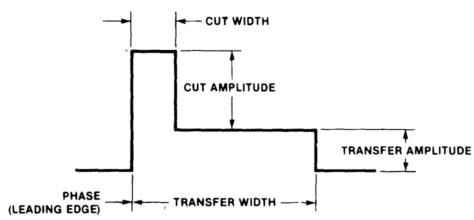


Figure 7. Control Pulse Waveform

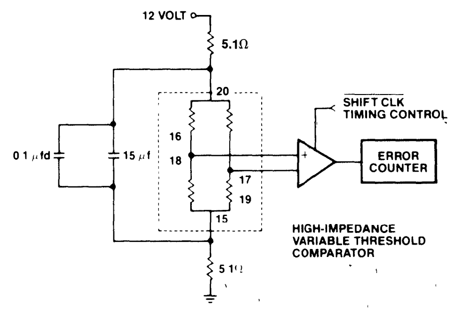


Figure 8. Test Setup for Output Measurement

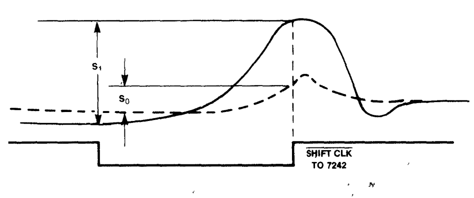


Figure 9. Detector Output Waveforms

intel®

7220-1
BUBBLE MEMORY CONTROLLER

7220-1	0 to 70°C
7220-5	−20 to +85°C

- **8080/8085/8088/8086 Microprocessor Interface**
- **Interfaces Up to Eight BPK-70 Bubble Storage Subsystems**
- **Self-Contained Timing**
- **DMA Handshake Capability**
- **Single or Multiple Page Block Transfers**
- **HMOS Technology**
- **Standard 40-Pin Dual In-Line Package**

The Intel® 7220-1 is a complete Bubble Memory Controller (BMC) designed to provide all the interface between Intel Bubble Memories and standard microprocessors such as the 8080, 8085, 8088, and 8086.

The 7220-1 has self-contained timing generation and DMA handshake capability. Single and/or multiple page block transfer capability is supported.

The 7220–1 is capable of interfacing with up to eight BPK 70 one megabit bubble storage subsystems. The 7220–5 is capable of interfacing with up to four BPK 70 one megabit bubble storage subsystems. The 7220–1 uses Intel's high performance HMOS technology. The 7220–1 is packaged in a standard 40-pin dual in-line package. All inputs and outputs are directly TTL compatible and the device uses a single +5 volt supply.

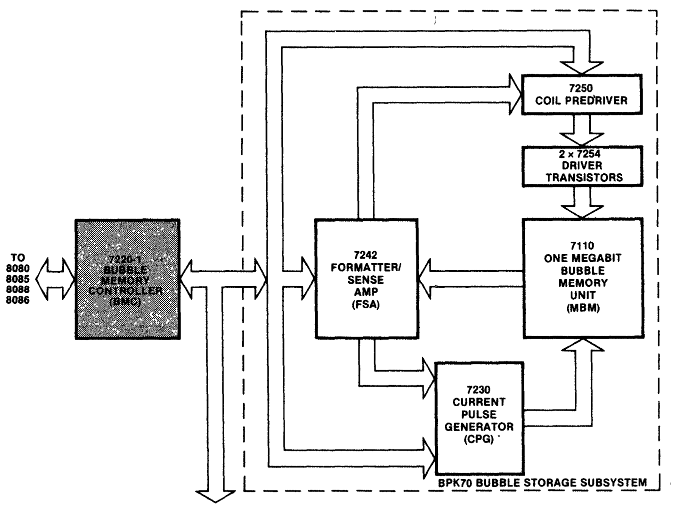


Figure 1. Block Diagram of a 128K Byte Bubble Storage System

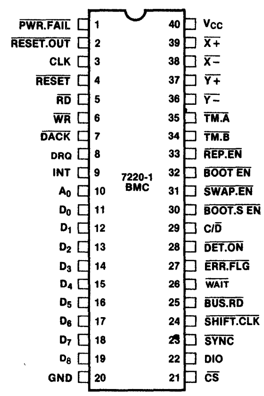


Figure 2. Pin Configuration

NOVEMBER 1982
ORDER NUMBER: 210286-001

HARDWARE DESCRIPTION

The 7220-1 Bubble Memory Controller is packaged in a 40-pin Dual In-Line Package (DIP). The following lists the individual pins and describes their function.

Table 1. Pin Description

Signal Name	Pin No.	I/O	Source/Destination	Description
V_{CC}	40	I		+5 VDC Supply
GND	20	I		Ground
$\overline{\text{PWR.FAIL}}$	1	I	7230 CPG	A low forces a controlled stop sequence and holds BMC in an IDLE state (similar to $\overline{\text{RESET}}$).
$\overline{\text{RESET.OUT}}$	2	O	7250 CPD/7242 FSA 7230 Reference Current Switch	An active low signal to disable external logic initiated by $\overline{\text{PWR.FAIL}}$ or $\overline{\text{RESET}}$ signals, but not active until a stopping point in a field rotation is reached (if the BMC is causing the bubble memory drive field to be rotated).
CLK	3	I	Host Bus	4 MHz, TTL-level clock.
$\overline{\text{RESET}}$	4	I	Host Bus	A low on this pin forces the interruption of any BMC sequencer activity, performs a controlled shut-down, and initiates a reset sequence. After the reset sequence is concluded, a low on this pin causes a low on the $\overline{\text{RESET.OUT}}$ pin, furthermore, the next BMC sequencer command must be either the Initialize or Abort command; all other commands are ignored.
$\overline{\text{RD}}$	5	I	Host Bus	A low on this pin enables the BMC output data to be transferred to the host data bus (D_0-D_8).
$\overline{\text{WR}}$	6	I	Host Bus	A low on this pin enables the contents of the host data bus (D_0-D_8) to be transferred to the BMC.
$\overline{\text{DACK}}$	7	I	Host Bus	A low signal is a DMA acknowledge. This notifies the BMC that the next memory cycle is available to transfer data. This line should be active only when DMA transfer is desired and the DMA ENABLE bit has been set. $\overline{\text{CS}}$ should not be active during DMA transfers except to read status. If DMA is not used, $\overline{\text{DACK}}$ requires an external pullup to V_{cc} (5.1K ohm).
DRQ	8	O	Host Bus	A high on this pin indicates that a data transfer between the BMC and the host memory is being requested.
INT	9	O	Host Bus	A high on this pin indicates that the BMC has a new status and requires servicing when enabled by the host CPU.
A_0	10	I	Host Bus	A high on this pin selects the command/status registers. A low on this pin selects the data register.
D_0-D_7	11-18	I/O	Host Bus	Host CPU data bus. An eight-bit bidirectional port which can be read or written by using the $\overline{\text{RD}}$ and $\overline{\text{WR}}$ strobes. D_0 shall be the LSB.
D_8	19	I/O	Host Bus	Parity bit.

AFN-01756C

Table 1. Pin Description (Continued)

Signal Name	Pin No.	I/O	Source/Destination	Description
$\overline{\text{CS}}$	21	I	Host Bus	Chip Select Input. A high on this pin shall disable the device to all but DMA transfers (i.e., it ignores bus activity and goes into a high impedance state).
DIO	22	I/O	7242 FSA	A bidirectional active high data line that shall be used for serial communications with 7242 FSA devices.
$\overline{\text{SYNC}}$	23	O	7242 FSA	An active low output utilized to create time division multiplexing slots in a 7242 FSA chain. It shall also indicate the beginning of a data or command transfer between BMC and 7242 FSA.
$\overline{\text{SHIFT.CLK}}$	24	O	7242 FSA	A controller generated clock that initiates data transfer between selected FSAs and their corresponding bubble memory devices. The timing of $\overline{\text{SHIFT.CLK}}$ shall vary depending upon whether data is being read or written to the bubble memory.
$\overline{\text{BUS.RD}}$	25	O	*	An active low signal that indicates that the DIO line is in the output mode. It shall be used to allow off-board expansion of 7242 FSA devices.
$\overline{\text{WAIT}}$	26	I/O	*	A bidirectional pin that shall be tied to the $\overline{\text{WAIT}}$ pin on other BMCs when operated in parallel. It shall indicate that an interrupt has been generated and that the other BMCs should halt in synchronization with the interrupting BMC. $\overline{\text{WAIT}}$ is an open collector active low signal. Requires an external pullup resistor to V_{CC} (5.1K ohm).
$\overline{\text{ERR.FLG}}$	27	I	7242 FSA	An active low input generated externally by 7242 FSA indicating that an error condition exists. It is an open collector input which requires an external pullup resistor (5.1K ohm).
$\overline{\text{DET.ON}}$	28	O	*	An active low signal that indicates the system is in the read mode and may be detecting. It is useful for power saving in the MBM.
C/$\overline{\text{D}}$	29	O	7242 FSA	A high on this line indicates that the BMC is beginning an FSA command sequence. A low on this line indicates that the BMC is beginning a data transmit or receive sequence.
$\overline{\text{BOOT.SW.EN}}$	30	O	7230 CPG	An active low signal which may be used for enabling the BOOT.SWAP of the 7230 CPG.
$\overline{\text{SWAP.EN}}$	31	O	7230 CPG	An active low signal used to create the swap function in external circuits.
$\overline{\text{BOOT.EN}}$	32	O	7230 CPG	An active low signal enabling the bootstrap loop replicate function in external circuitry.
$\overline{\text{REP.EN}}$	33	O	7230 CPG	An active low signal used to enable the replicate function in external circuitry.
$\overline{\text{TM.B}}$	34	O	7230 CPG	An active low timing signal generated by the decoder logic for determining TRANSFER pulse width.
$\overline{\text{TM.A}}$	35	O	7230 CPG	An active low timing signal generated by the decoder logic for determining CUT pulse width.
$\overline{\text{Y}-}$, $\overline{\text{Y}+}$, $\overline{\text{X}-}$, $\overline{\text{X}+}$	36-39	O	7250 CPD	Four active low timing signals generated by the decoding logic and used to create coil drive currents in the bubble memory device.

*Not used in minimum (128K byte) system

FUNCTIONAL DESCRIPTION

The 7220-1 Bubble Memory Controller provides the user interface to the bubble memory system. The BMC generates all memory system timing and control, maintains memory address information, interprets and executes user request for data transfers, and provides a Microprocessor-Bus compatible interface for the magnetic bubble memory system.

Figure 3 is a block diagram of the 7220-1 Bubble Memory Controller (BMC). The following paragraphs describe the functions of the individual functional sections of the BMC.

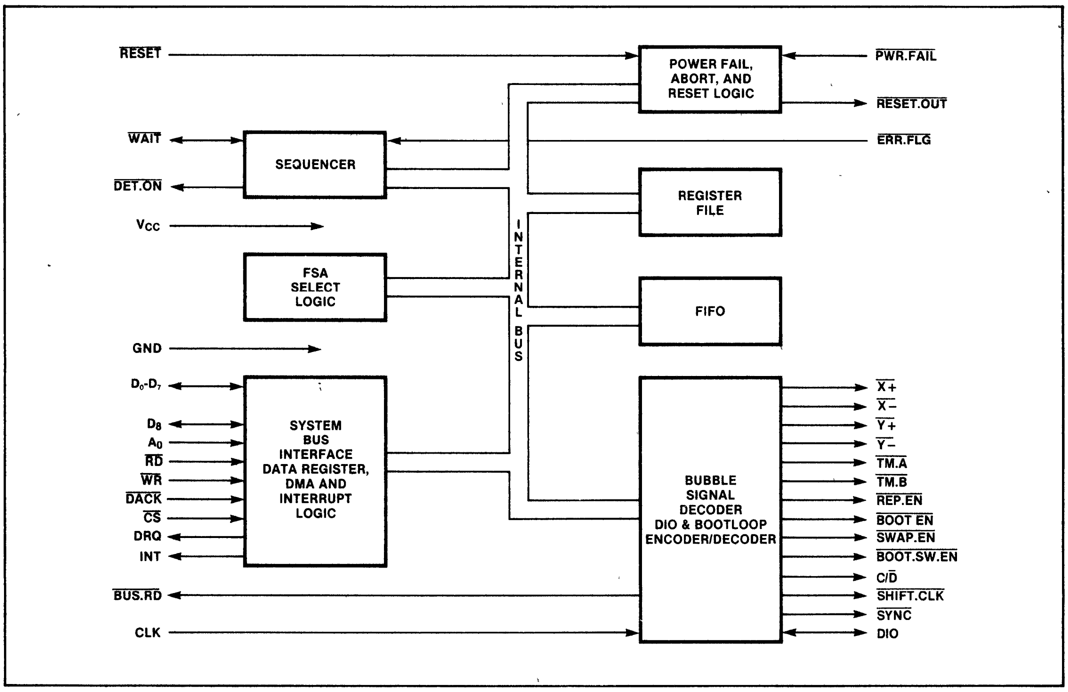


Figure 3. 7220-1 Bubble Memory Controller (BMC), Block Diagram

System Bus Interface—The System Bus Interface (SBI) logic contains the timing and control logic required to interface the BMC to a non-multiplexed bus. The logic also contains the circuitry to check and generate odd parity on transfers across the bus. The interface has input data, output data, and status data latches. The BMC can interface asynchronously to the host CPU. With a 4-MHz clock, it is capable of sustaining a 1.14 Mbyte per second transfer rate, while data is available in the BMC FIFO.

FIFO—The FIFO consists of a 40 × 8 bit FIFO RAM for data storage. The FIFO block also contains input and output data latches, providing double data buffering, to improve the R/W cycle times seen at the system bus interface. The FIFO may be used as a general purpose FIFO when a command is not being executed by the BMC Sequencer. In this mode, the FIFO READY status bit becomes a FIFO not-empty indicator indicating that the RAM and input/output latches have at least one byte of data.

DMA and Interrupt Logic—The DRQ pin has two functions:

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

(2) If the DMA enable bit is reset, the DRQ pin acts as a "ready for data transfer interrupt" pin. It becomes active when 22 bytes may be read from or written into the BMC; it is reset when this condition no longer exists.

Register File—The register file contains 7 eight-bit registers that are accessible by the host CPU. Refer to the Register Section for details.

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

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

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

DIO Bootloop Decoder/Encoder—Performs parallel-to-serial and serial-to-parallel conversions between the FIFO data and the serial bit stream on the DIO line. This block also generates the BUS.RD signal, which indicates the direction of data transfer on the DIO line (this is useful in situations which require external buffering on the DIO line). This block also contains the circuitry which decodes the bootloop data during a Read Bootloop or Initialize operation, and encodes the bootloop data during a Write Bootloop operation.

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

Power Fail and Reset—Provides a means of resetting the bubble systems in an orderly manner, when activated by the PWR.FAIL signal, the RESET signal, or the ABORT command. The additive noise on the PWR.FAIL pin should be less than 150 mV for proper powerfail operation.

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

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

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

X+, X−, Y+, and Y− go to the 7250 CPDs, and are used to enable the coil drive currents in the MBMs.

TM.A and TM.B go to the 7230 CPGs, and are used to determine, respectively, the pulse widths for the CUT and TRANSFER functions used in replicating and generating the bubbles.

Table 2. 7220-1 BMC Timing (Degrees)**

Signal	Start	Width	End
X+	270°	108°	378°
Y+	0°	108°	108°
X−	90°	108°	198°
Y−	180°	108°	288°
TM.A (ODD)	270°	4°	274.5°
TM.A (EVEN)	90°	4°	94.5°
TM.B (ODD)	270°	90°	360°
TM.B (EVEN)	90°	90°	180°
BOOT.EN	252°	108°	360°
REP.EN	252°	108°	360°
SWAP.EN	180°	5.7°	697°
BOOT.SW.EN	180°	DC*	180°
SHIFTCLK (RD)	186.75°	99°	285.75°
SHIFTCLK (WR)	72°	288°	360°

*Stays low for 4118 field rotation periods when writing the MBM Bootloop.

**All phases relative to Y+ start phase All entries ±1 26° except TM A width which is ±0 5°

SWAP.EN, REP.EN, BOOT.SW.EN, and BOOT.EN all go to the 7230 CPG. They are used to enable, respectively, the data swap, data replicate, boot swap, and boot replicate functions within the MBMs.

SHIFT.CLK goes to the FSAs. It is used to control the timing of events at the interface between each FSA and its corresponding MBM. (Refer to 7242 FSA Specification for a description of the BMC/FSA interface.)

SYNC and C/D control the serial communications between the BMC and the FSAs (on the DIO line).

USER-ACCESSIBLE REGISTERS

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

Register Addressing

Selection of the user-accessible registers depends on register address information sent from the user to the BMC. This address information is sent via a single address line (designated A_0) and data bus lines D_0 through D_4.

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

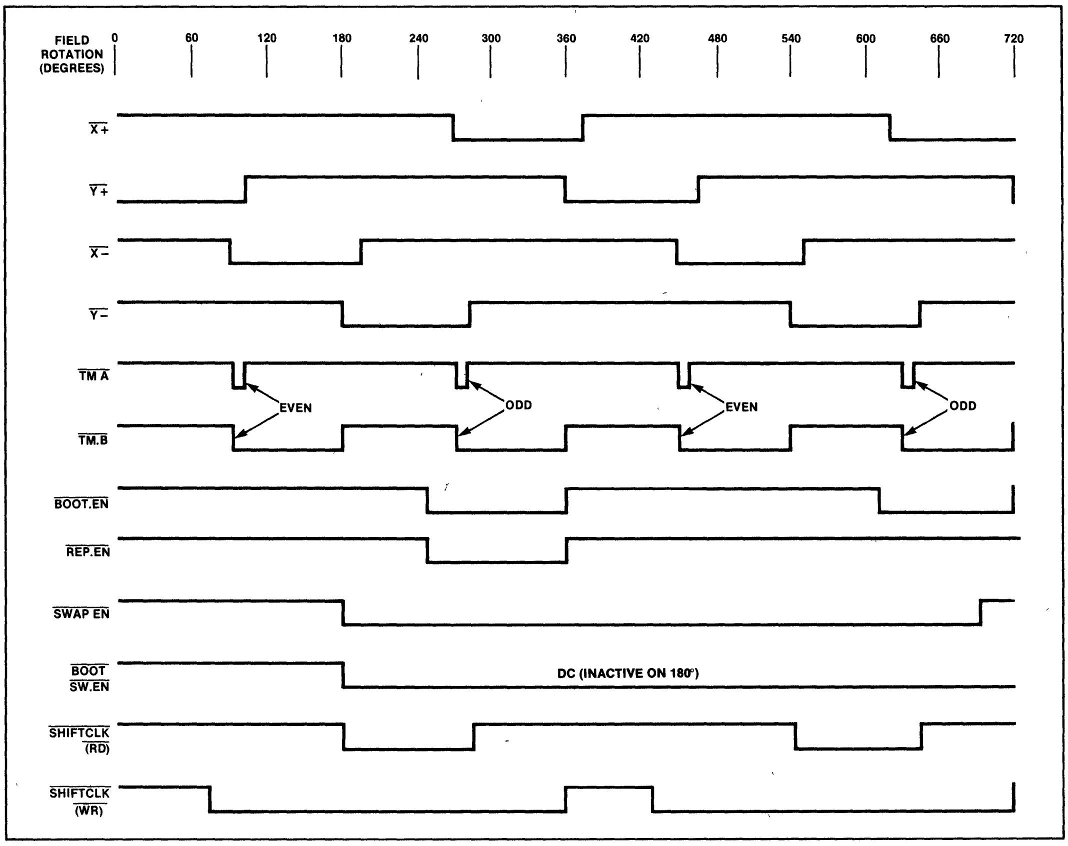


Figure 4. 7220-1 BMC Timing Diagram

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

Table 3. Address Assignments for the User-Accessible Registers

A0	D7	D6	D5	D4	D3	D2	D1	D0	Symbol	Name of Register	Read/Write
1	0	0	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

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

A0	RAC 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

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 o write request with A0 = 0.
LSB = Least Significant Byte
MSB = Most Significant Byte

The register file contains the registers with address 1010 through 1111. These registers are also called parametric registers because they contain flags and parameters that determine exactly how the BMC will respond to commands written to the CMDR.

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

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

REGISTER DESCRIPTIONS

Command Register (CMDR) 4 Bits, Write Only

The user issues a command to the BMC by writing a 4-bit command code to the CMDR.

Table 4 lists the 4-bit command codes used to issue the sixteen commands recognized by the BMC:

Table 7 is a listing of the commands and their functions.

Table 4. Command Code Definitions

D3	D2	D1	Do	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 most commonly used commands in normal operation are:

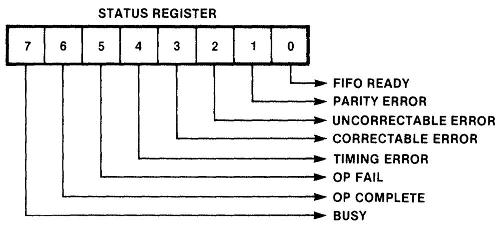
Initialize
Read Bubble Data
Write Bubble Data
Reset FIFO
Read Seek
Write Seek
Abort
Read Corrected Data
Software Reset
Read FSA Status
MBM Purge

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

Read Bootloop Register
Write Bootloop Register
Write Bootloop Register Masked
Read Bootloop
Write Bootloop

Status Register (STR) 8 Bits, Read Only

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



BUSY (when = 1) indicates that the BMC is in the process of executing a command. When equal to 0, BUSY indicates that the BMC is ready to receive a new command. In the case of Read Bubble Data, Read Bootloop, read Bootloop Register, or Read Corrected Data commands, BUSY may also indicate that the data has not been completely removed from the FIFO, and that DRQ is still active. BUSY will then drop as soon as DRQ does (after the user has finished reading the data remaining in the FIFO).

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

OP FAIL (when = 1) indicates that the BUSY bit has gone inactive with either the TIMING ERROR or UNCORRECTABLE ERROR bits active.

TIMING ERROR (when = 1) indicates that a FSA has reported a timing error to the BMC, or that the host system has failed to keep up with the BMC, thereby causing the BMC FIFO to overflow or to underflow. TIMING ERROR is also set if no bootloop sync word is found during initialization, or if a Write Bootloop command is issued when the WRITE BOOTLOOP ENABLE bit is equal to zero in the enable register.

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

AFN-01756C

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

PARITY ERROR (when=1) indicates that the BMC's parity check circuitry has detected a parity error on a data byte sent to the BMC by the user on the data lines D_0-D_8.

FIFO READY has two functions. The FIFO READY functions are as follows:

NOTE: IF RAC ≠ FIFO, FIFO READY = 1

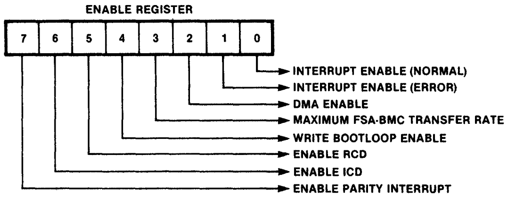
STATUS BITS		READ	WRITE
FIFO READY	BUSY		
1	1	data in FIFO	space in FIFO
0	1	no data	no space
1	0	— data in FIFO —	
0	0	— FIFO empty —	

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

STR Bits 1 through 6 are reset when a new command is issued. They may also be reset by making a write request (WR=0) to the BMC with A_0=1, D_4=0, and D_5=1 (that is, writing the RAC with D_5=1). This operation also resets the "INT" pin to "0". NOTE: A byte of FIFO data can be lost when using this procedure if the RAC is written to other than the FIFO address when data is still present in FIFO.

Enable Register (ER) 8 Bits, Write Only

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



In the above figure and in the text below, the following abbreviations are used:

ICD = INTERNALLY CORRECT DATA
RCD = READ CORRECTED DATA
UCE = UNCORRECTABLE ERROR
CE = CORRECTABLE ERROR
TE = TIMING ERROR

ENABLE PARITY INTERRUPT enables the BMC to interrupt the host system (via the INT line) when the BMC detects a parity error on the data bus lines D_0-D_7.

ENABLE ICD enables the BMC to give the Internally Correct Data command to the FSAs when an error has been detected by the FSA's error detection and correction circuitry. Each FSA responds to such a command by internally cycling the data through its error correction network. When finished, the FSA returns status to the BMC as to whether or not the error is correctable. The value of ENABLE ICD affects the action of INTERRUPT ENABLE (ERROR).

ENABLE RCD enables the BMC to give the Read Corrected Data command to the FSAs when an error has been detected. This causes each FSA to correct the error (if possible) and also to transfer the corrected data to the BMC. The Read Corrected Data command is also used to read into the BMC data previously corrected by the FSA in response to an Internally Correct Data command. In either case, when the data transfer has been completed, the BMC reads each FSA's status to determine whether or not the error was correctable. In the case of an uncorrectable error, bad data may have been sent to the user. The value of ENABLE RCD affects the action of INTERRUPT ENABLE (ERROR).

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

MFBTR controls the maximum burst transfer rate from FSA(s) to BMC FIFO. This rate is variable on the "last page" of a multiple page transfer. (In one page transfers the last page is the only page.) See Table 5 for effects of this bit on the various 7220-1 commands.

Table 5. MFBTR Bit Definitions

Number of MBMs Operated in Parallel	Maximum Required Host Interface Data Rate	MFBTR Bit	
		Read Command	Write Command
1	50K byte/sec	0	N/A
2	100K byte/sec	0	N/A
4	200K byte/sec	0	N/A
8	400K byte/sec	0	N/A
1	12 5K byte/sec	1	0
2	25K byte/sec	1	0
4	50K byte/sec	1	0
8	100K byte/sec	1	0

NOTE. The MFBTR bit should always be set to "0" for all commands except "Read Bubble Data."

DMA ENABLE (when=1) enables the BMC to operate in DMA data transfer mode, using the DRQ and $\overline{DACK}$ signals in interaction with a DMA controller. When equal to zero, DMA ENABLE sets up the controller to support interrupt driven or polled data transfer.

INTERRUPT ENABLE (ERROR) selects error conditions under which the BMC stops command execution and interrupts the host processor (via the INT line). INTERRUPT ENABLE (ERROR) operates in conjunction with ENABLE ICD and ENABLE RCD.

Enable ICD	Enable RCD	Interrupt Enable (ERROR)	Interrupt Action
0	0	0	No interrupts due to errors
0	0	1	Interrupt on TE only
0	1	0	Interrupt on UCE or TE
0	1	1	Interrupt on UCE, CE, or TE
1	0	0	Interrupt on UCE or TE
1	0	1	Interrupt on UCE, CE, or TE
1	1	0	Not used
1	1	1	Not used

TE = Timing Error, CE = Correctable Error,
UCE = Uncorrectable Error.

INTERRUPT ENABLE (NORMAL) (when = 1) enables the BMC to interrupt the host system (via the INT line), when a command execution has been successfully completed (OP COMPLETE = 1 in the STR).

Utility Register (UR) 8 Bits, Read or Write

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 BMC operation, but is provided as a convenience to the user.

Block Length Register (BLR) 16 Bits, Write Only

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

BLOCK LENGTH REGISTER MSB: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
BLOCK LENGTH REGISTER LSB: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

MSB bits 7–4: NUMBER OF FSA CHANNELS (NFC); MSB bit 3: X; MSB bits 2–0 and LSB bits 7–0: NUMBER OF PAGES TO BE TRANSFERRED

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

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

Address Register (AR) 16 Bits, Read or Write

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

ADDRESS REGISTER MSB: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
ADDRESS REGISTER LSB: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

MSB bit 7: X; MSB bits 6–3: MBM SELECT; MSB bits 2–0 and LSB bits 7–0: STARTING ADDRESS WITHIN EACH MBM

Within each MBM there are 2048 possible starting address locations for a data read or write operation, hence the requirement for 11 bits in the starting address.

The selection of the MBMs to be read or written is specified by AR MSB Bits 3-6. The BMCs interpretation of these bits depends on the number of MBMs in a group, which is specified by BLR MSB Bits 4-7.

Table 6 shows which MBM groups are selected in response to given values for BLR MSB Bits 4-7 and AR MSB Bits 3-6. A 1-megabyte system (8 MBMs) is represented, with the FSA channels numbered 0 through F:

Table 6. Selection of FSA Channels

AR MSB Bits (6,5,4,3)	BLR MSB Bits (7,6,5,4)				
	0000	0001	0010	0100	1000
0000	0	0,1	0,1,2,3	0 to 7	0 to F
0001	1	2,3	4,5,6,7	8 to F	
0010	2	4,5	8,9,A,B		
0011	3	6,7	C,D,E,F		
0100	4	8,9			
0101	5	A,B			
0110	6	C,D			
0111	7	E,F			
1000	8				
1001	9				
1010	A				
1011	B				
1100	C				
1101	D				
1110	E				
1111	F				

The accessing of single FSA channels is done only as part of diagnostic processes. AR MSB Bit 7 is not used.

FIFO Data Buffer (FIFO) 40 × 8 Bits, Read or Write

The BMC FIFO is a 40-byte buffer through which data passes on its way from the FSAs to the user, or from the user to the FSAs. The FIFO allows the data transfer to proceed in an asynchronous and flexible manner, and relaxes timing constraints, both to the FSAs and also to the user's equipment. The user's system must, however, meet the data rate requirements. When the BMC is busy (executing a command) the FIFO functions as a data buffer. When the BMC is not busy, the FIFO is available to the user as a general purpose FIFO.

FUNCTIONAL OPERATION

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

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

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

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

From the FSA FIFO, data is sent to the bubble memory controller (BMC) in the form of a serial bit stream, via a one-line bidirectional data bus (DIO). The data is multiplexed onto the DIO line, with data bits coming alternately from the A and B channels of the FSA. The BMC outputs a $\overline{\text{SYNC}}$ pulse to the $\overline{\text{SELECT.IN}}$ input of the FSA. The FSA responds by placing a data bit from the A channel FIFO on the DIO line. One clock cycle later, a data bit from the B channel FIFO is placed on the DIO line. The BMC continues to output $\overline{\text{SYNC}}$ pulses, once every 20 or 80 clock cycles, each time receiving two data bits in return.

In the BMC, the data undergoes serial-to-parallel conversion, and is assembled into bytes, which are then placed in the BMC FIFO, which can hold 40 bytes of data. From this FIFO, the data bytes are written onto the user interface.

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

Multiple-MBM Systems

The 7220-1 BMC can interface up to 8 one-megabit BPK70 Bubble Storage subsystems. The data flow in a multiple-BPK70 system is in most respects similar to that which occurs in a one-BKP70 subsystem. The difference is in the time-division multiplexing that occurs on the DIO bus line between the BMC and the FSAs.

For data transfer operations, the BMC may exchange data with as few as two FSA channels (one BPK70) or as many as 16 FSA channels (eight BPK70s).

SOFTWARE INTERFACE—The general procedure for communicating with the BMC is:

Pass parameters to the BMC by loading the registers.

Send the desired command.

Read the status/command register until BMC is not busy (or use "INT" pin).

Examine the status register to determine whether the operation was successful.

Table 7. Detailed Command Descriptions

Initialize	The BMC executes the Initialize command by first interrogating the bubble system to determine how many FSAs are present, then reading and decoding the bootloop from each MBM and storing the results in the corresponding FSA's bootloop register. All the parametric registers must be properly set up before issuing the Initialize command.
Read Bubble Data	The Read Bubble Data command causes data to be read from the MBMs into the BMC FIFO. The selection of the MBMs to be accessed and the starting address for the read operation is specified in the address register (AR). The block length register (BLR) specifies the number of system pages to be read. All the parametric registers must be properly set up before issuing the Read Bubble Data command.
Write Bubble Data	The Write Bubble Data command causes data to be read from the BMC FIFO and written into the MBMs. The selection of the MBMs to be accessed and the starting address for the write operation is specified in the address register (AR). The block length register (BLR) specifies the number of system pages to be written. All the parametric registers must be properly set up before issuing the Write Bubble Data command.
Read Seek	The Read Seek command rotates the selected MBMs to a designated page address location. No data transfer occurs. The positioning is such that the next data location available to be read is the specified (in AR) page address plus one. The Read Seek command may be used to reduce latency (access time) in cases where information is available for the user to predict the location of an impending read reference to the MBMs.

Table 7. Detailed Command Descriptions (Continued)

Write Seek	The Write Seek command rotates the selected MBMs to a designated page address location. No data transfer occurs. The positioning is such that the next data location available to be written is the specified (in AR) page address plus one. The Write Seek command may be used to reduce latency (access time) in cases where information is available for the user to predict the location of an impending write reference to the MBMs.
Abort	The Abort command causes a controlled termination of the command currently being executed by the BMC. The Abort command will be accepted by the BMC (and is typically issued) when the BMC is busy.
MBM Purge	The MBM Purge command clears all BMC registers, counters, and the MBM address RAM. Furthermore, it determines how many FSA channels are present in the system and stores this value in the 7220-1. The "INITIALIZE" command uses this command as a subroutine.
Read Corrected Data	The Read Corrected Data command causes the BMC to read into the BMC FIFO a 256-bit block of data from the FIFO of each selected FSA channel, after an error has been detected. The data cycles through the error correction network of the FSA. After the data has been read, the FSA reports to the BMC whether or not the error was correctable. The Read Corrected Data command is used only when the system is in error correction mode (ENABLE ICD or ENABLE RCD set in the ER).
Software Reset	The Software Reset command clears the BMC FIFO and all registers, except those containing initialization parameters. It also causes the BMC to send the Software Reset command to selected FSAs in the system. No reinitialization is needed after this command.
Read FSA Status	The Read FSA Status command causes the BMC to read the 8-bit status register of all FSAs, and to store this information in the BMC FIFO. The Read FSA Status command is independent all parametric registers.
Read Bootloop Register	The Read Bootloop Register command causes the BMC to read the bootloop register of the selected FSA channels and to store this information in the BMC FIFO. Twenty bytes are transferred for each FSA channel selected.
Write Bootloop Register Masked	Proper operation of the FSAs during data transfer to or from the MBMs requires that the bootloop register contain (if error correction is used) exactly 270 logic 1s for each FSA bootloop register. The user may select any subset of 270 "good" loops from the total number of available loops (if error correction is not used, 270 replaced by 272). As an alternative, the Write Bootloop Register Masked command may be used. This command counts the number of logic 1s and masks out the remaining 1s after the proper count has been reached. The Initialize command uses this command as a subroutine.
Read Bootloop	The Read Bootloop command causes the BMC to read the bootloop from the selected MBM, and to store the decoded bootloop information in the BMC FIFO. The Initialize command uses this command as a subroutine.
Write Bootloop	The Write Bootloop command causes the existing contents of the selected MBM's bootloop to be replaced by new bootloop data based on 40 bytes of information stored in the FIFO (the user must actually write 41 bytes, where the 41st byte is all 0s). Encoding of the bootloop data is done by the BMC hardware.

ABSOLUTE MAXIMUM RATINGS

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

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

D.C. CHARACTERISTICS (T_A = see front page; V_{CC} = 5.0V +5%, −10%)

Symbol	Parameter	Min.	Max.	Unit	Test Condition
V_{IL}	Input Low Voltage		0.8	V	
$V_{IH(1)}$	Input High Voltage (all but $\overline{PWR.FAIL}$)	2.0	V_{CC}+0.5V	V	
$V_{IH(2)}$	Input High Voltage ($\overline{PWR.FAIL}$)	2.5	V_{CC}+0.5V	V	
$V_{OL(1)}$	Output Low Voltage (All outputs except $\overline{DET.ON}$, $\overline{BUS.RD}$, $\overline{SHIFT.CLK}$, and $\overline{SYNC}$		.45	V	I_{OL} = 3.2 mA
$V_{OL(2)}$	Output Low Voltage $\overline{DET.ON}$, $\overline{BUS.RD}$, $\overline{SHIFT.CLK}$, $\overline{SYNC}$		.45	V	I_{OL} = 1.6 mA
V_{OH}	Output High Voltage	2.4		V	I_{OH} = 400μA
$\lvert I_{IL} \rvert$	Input Leakage Current		10	μA	$0 \leq V_{IN} \leq V_{CC}$
$\lvert_{OFL}\rvert$	Output Float Leakage		10	μA	$0.45 \leq V_{OUT} \leq V_{CC}$
I_{CC}	Power Supply Current from V_{CC}		200	mA	

A.C. CHARACTERISTICS

(T_A = see table 1; V_{CC} = 5.0V + 5%, −10%; C_L = 150 pF; unless otherwise noted.)

Symbol	Parameter	Min.	Max.	Unit	Test Condition
t_P	Clock Period	249.75	250.25	ns	
t_ϕ	Clock Phase Width (High Time)	.45 t_P	.55 t_P		
t_R-t_F	Input Signal Rise and Fall Time		30	ns	

FSA INTERFACE TIMINGS (under pin loading)

Symbol	Parameter	Min.	Max.	Unit	Test Condition
t_{CDV}	CLK to DIO Valid Delay		150	ns	Under Pin Loads*
t_{CDF}	CLK to DIO Entering Float	10	250	ns	Under Pin Loads*
t_{CDE}	CLK to DIO Enabled from Float		150	ns	Under Pin Loads*
t_{CDH}	CLK to DIO Hold Time	0		ns	Under Pin Loads*
t_{CSOL}	CLK to $\overline{SYNC}$ Leading Edge Delay		120	ns	Under Pin Loads*
t_{CSOT}	CLK to $\overline{SYNC}$ Trailing Edge Delay	10	100	ns	Under Pin Loads*
t_{DC}	DIO Setup Time to Clock	80		ns	Under Pin Loads*
t_{DHC}	DIO Hold Time from Clock	0		ns	Under Pin Loads*
t_{COL}	CLK to Output Leading Edge		150	ns	Under Pin Loads*
t_{COT}	CLK to Output Trailing Edge	0	190	ns	Under Pin Loads*
t_{EW}	$\overline{ERR. FLG}$ Pulse Width	200		ns	Under Pin Loads*
t_{SCFT}	$\overline{SHIFTCLK}$ to $\overline{Y-}$ Trailing Edge	80	200	ns	Under Pin Loads*

AFN-01756C

A.C. CHARACTERISTICS (Continued) (T_A = see table 1; V_{CC} = 5.0 + 5%, − 10%; C_L = 150 pF; unless otherwise noted.)

READ CYCLE (HOST INTERFACE)

Symbol	Parameter	Min.	Max.	Unit	Test Condition
t_{AC}	Select Setup to $\overline{RD}\downarrow$	0		ns	
t_{CA}	Select Hold from $\overline{RD}\uparrow$	0		ns	
t_{RR}	$\overline{RD}$ Pulse Width	200		ns	
t_{AD}	Data Delay from Address		150	ns	
t_{RD}	Data Delay from $\overline{RD}\downarrow$		150	ns	
t_{DF}	Output Float Delay	10	100	ns	
t_{DC}	DACK Setup to $\overline{RD}\downarrow$	0		ns	
t_{CD}	DACK Hold from $\overline{RD}\uparrow$	0		ns	
t_{KD}	Data Delay from $\overline{DACK}\downarrow$		150	ns	
t_{CYCR}	"Read" Cycle Time	(DMA Mode) $4t_p$-$t_ø$		ns	In non DMA mode. t_{CYCR} Min. = $6t_p$-$t_ø$

WRITE CYCLE (HOST INTERFACE)

Symbol	Parameter	Min.	Max.	Unit	Test Condition
t_{AC}	Select Setup to $\overline{WR}\downarrow$	0		ns	
t_{CA}	Select Hold from $\overline{WR}\uparrow$	0		ns	
t_{WW}	$\overline{WR}$ Pulse Width	200		ns	
t_{DW}	Data Setup to $\overline{WR}\uparrow$	200		ns	
t_{WD}	Data Hold from $\overline{WR}\uparrow$	0		ns	
t_{DC}	$\overline{DACK}$ Setup to $\overline{WR}\downarrow$	0		ns	
t_{CD}	$\overline{DACK}$ Hold from $\overline{WR}\uparrow$	0		ns	
t_{CYCW}	"Write" Cycle Time	$4t_P + t_{ww}$			
t_{CQ}	Request Hold from $\overline{RD}$ or $\overline{WR}$ (Non-Burst Mode)		150	ns	
t_{DEADW}	Inactive Time between $\overline{WR}\uparrow$ and $\overline{WR}\downarrow$	$4t_P$		ns	
t_{DEADR}	Inactive Time between $\overline{RD}\uparrow$ and $\overline{RD}\downarrow$	150			

7250-7230 INTERFACE TIMINGS

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

*Bubble Pin Loads Shown Below

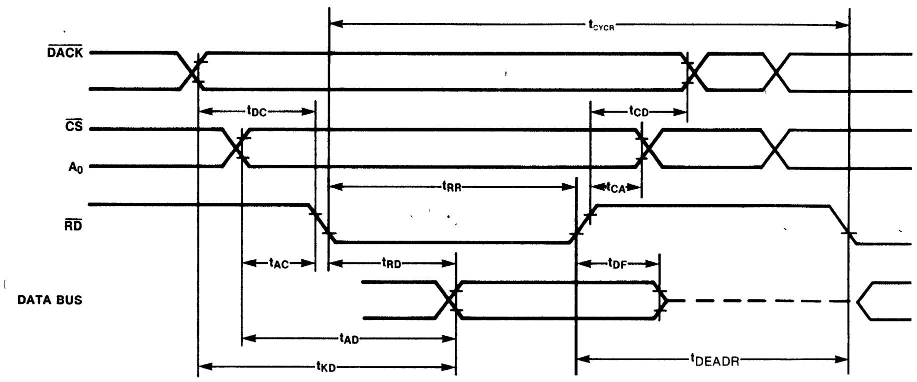
PIN LOADINGS

Pin Names	Value	Unit
$\overline{X+}$, $\overline{X-}$, $\overline{Y+}$, $\overline{Y-}$	150	pF
$\overline{TM\ A}$, $\overline{TM\ B}$, $\overline{REP\ EN}$, $\overline{BOOT\ EN}$, $\overline{SWAP\ EN}$, $\overline{BOOT\ SW\ EN}$, C/D, $\overline{ERR\ FLG}$, $\overline{WAIT}$, $\overline{SYNC}$	50	pF
$\overline{DET\ ON}$ & $\overline{SHIFT\ CLK}$	100	pF
$\overline{BUS\ READ}$	10	pF

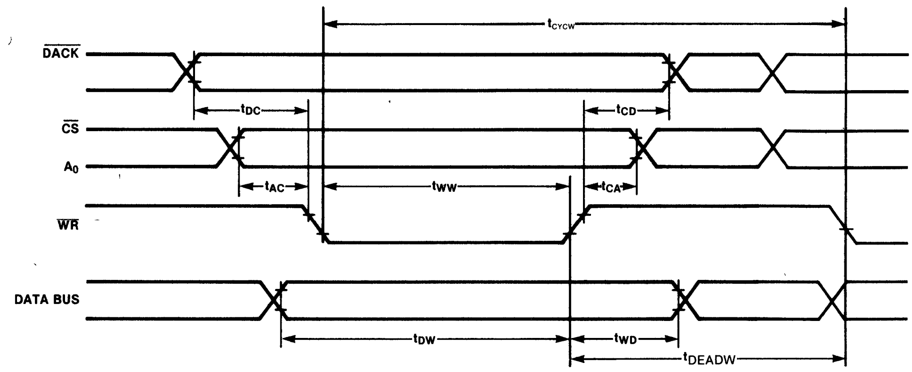
AFN-01756C

WAVEFORMS

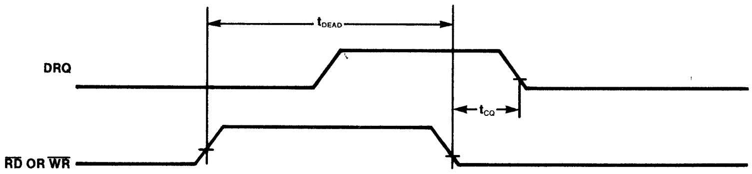
READ (HOST INTERFACE)



WRITE (HOST INTERFACE)



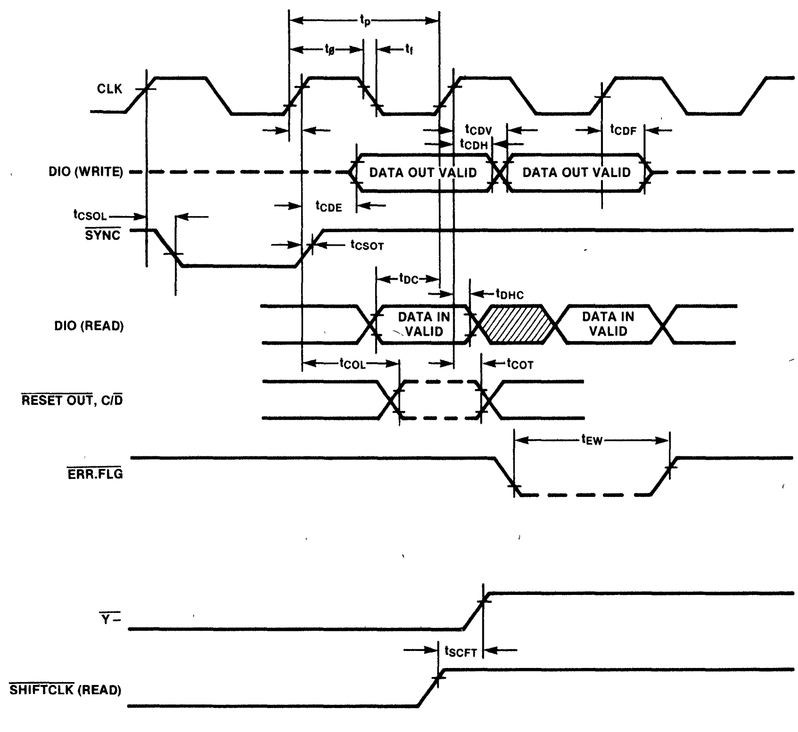
DMA (HOST INTERFACE)



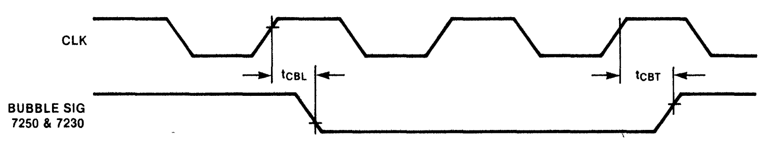
AFN-01756C

WAVEFORMS (Continued)

7242 INTERFACE TIMINGS



7250 & 7230 INTERFACE TIMINGS



7230
CURRENT PULSE GENERATOR FOR BUBBLE MEMORIES

7230	0 to 70°C
7230-4	10 to 55°C
7230-5	−20 to +85°C

- **TTL Compatible Inputs**
- **Provides All Pulses for Intel Bubble Memories**
  —Replicate, Swap, Generate, Boot Replicate and Bootswap
- **Current Sink Outputs Designed to Directly Drive Bubble Memory**
- **Direct Interface to Bubble Memory Controller**
- **Automatic Power Fail and Reset**
- **Operates from +5 and +12 Volts Only**
- **Schottky Bipolar Technology**
- **Standard 22-Pin Dual-In-Line Package**

The Intel 7230 is a Current Pulse Generator (CPG) designed to drive Intel Magnetics Bubble Memories. The 7230 is a Schottky Bipolar, TTL input compatible device that converts digital timing signals to analog current pulses. The CPG provides all pulses for Intel Magnetics Bubble Memories (7110 Family). These include Replicate, Swap, Generate, Boot Replicate and Bootswap pulses. The high-current sinking outputs directly drive the bubble memory. It also directly interfaces to the Intel Magnetics Bubble Memory Controller (7220-1) and Formatter/Sense amplifier (7242).

The 7230 operates from 5-volt and 12-volt power supplies and is in a standard 22-pin dual-in-line package.

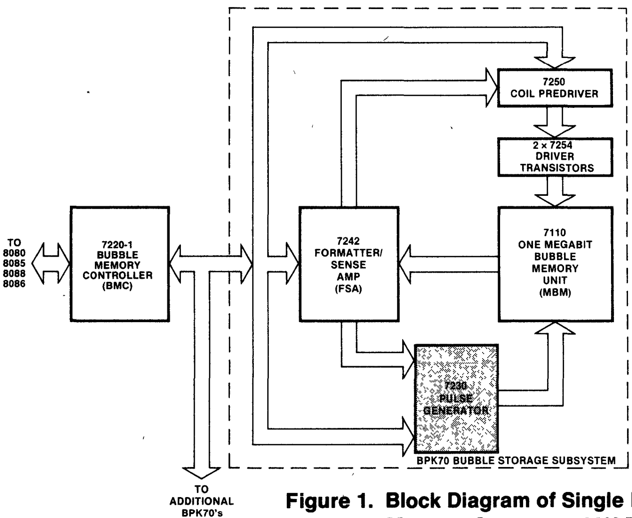


Figure 1. Block Diagram of Single Bubble Memory System—128K Bytes

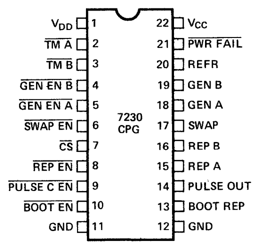


Figure 2. Pin Configuration

NOVEMBER 1982
ORDER NUMBER: 210798-001

EXTERNAL RESISTOR REQUIREMENTS

Connect a 1% 3.48K ohm resistor based between pin 20 and ground or referenced current switch as outlined in BPK72 User's Manual.

Table 1. Pin Description

Symbol	Pin No.	Description
$\overline{\text{BOOT.EN}}$	10	An active low input enabling the BOOT.REP output current pulse.
BOOT.REP	13	An output providing the current pulse for bootstrap loop replication in the bubble memory.
BOOT.SWAP	14	An output providing a current pulse which may be used for writing data into the bootstrap loop.
$\overline{\text{BOOT.SW.EN}}$	9	An active low input enabling the BOOT.SWAP output current pulse.
$\overline{\text{CS}}$	7	An active low input for selecting the chip. The chip powers down during deselect.
GEN.A	18	An output providing the current pulse for writing data into the "A" quads of the bubble memory.
GEN.B	19	An output providing the current pulse for writing data into the "B" quads of the bubble memory.
$\overline{\text{GEN.EN.A}}$	5	An active low input enabling the GEN.A output current pulse.
$\overline{\text{GEN.EN.B}}$	4	An active low input enabling the GEN.B output current pulse.
$\overline{\text{PWR.FAIL}}$	21	An active low, open collector output indicating that either V_{CC} or V_{DD} is below its threshold value.
REFR.	20	The pin for the reference current generator to which an external resistance must be connected.
REP.A	15	An output providing the current pulse for replication of data in the "A" quads of the bubble memory.
REP.B	16	An output providing the current pulse for replication of data in the "B" quads of the bubble memory.
$\overline{\text{REP.EN}}$	8	An active low input enabling the REP.A and REP.B outputs.
SWAP	17	An output providing the current pulse for exchanging the data between the input track and the storage loops in the bubble memory.
$\overline{\text{SWAP.EN}}$	6	An active low input enabling the SWAP output.
$\overline{\text{TM.A}}$	2	An active low timing signal determining the cut pulse widths of the BOOT.REP, GEN.A, GEN.B, REP.A and REP.B outputs.
$\overline{\text{TM.B}}$	3	An active low timing signal determining the transfer pulse widths of the BOOT.REP, GEN.A, GEN.B, REP.A and REP.B outputs.

AFN-01367B

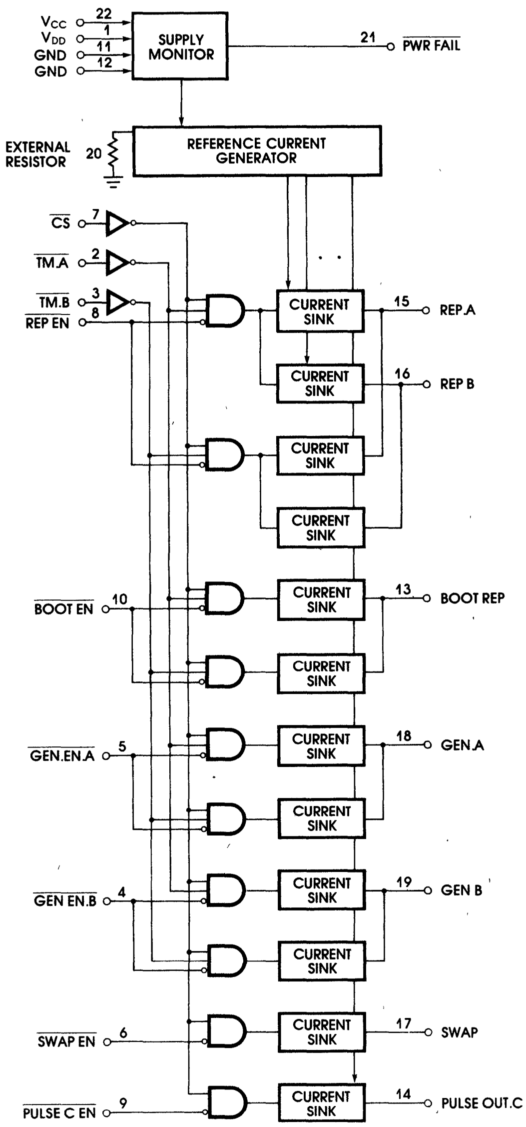


Figure 3. Logic Diagram

AFN-01367B

ABSOLUTE MAXIMUM RATINGS*

Temperature Under Bias .... −40° to +100°C
Storage Temperatuare .... −65°C to +150°C
V_{CC} and Input Voltages .... −0.5V to +7V
V_{DD} and Output Voltages .... −0.5V to +12.6V
Power Dissipation 1W

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

D.C. CHARACTERISTICS

(T_A = range specified in Table 1; V_{CC} = 5.0V ± 5%, ±5% V_{DD} = 12V ±5%; unless otherwise specified)

Symbol	Parameter	Limits			Unit	Test Conditions
		Min.	Typ.	Max.		
I_{IL}	Input Low Current			−0.4	mA	V_{IL} = 0.4V, V_{CC} = 5.25V
I_{IH}	Input High Current			20	μA	V_{IH} = V_{CC} = 5.25V
V_{IL}	Input Low Voltage			0.8	V	
V_{IH}	Input High Voltage	2.0			V	
V_C	Input Clamp Voltage			−1.5	V	I = −18 mA, V_{CC} = 4.75V
I_{CEX1}	Output Leakage Current (All Outputs except PWR.FAIL)			1.0	mA	V_{CC} = 5.25V, V_{DD} = 12.6V
I_{CEX2}	$\overline{PWR.FAIL}$ Output Leakage Current			40	μA	V_{OH} = V_{CC} = 5.25V
V_{OL}	$\overline{PWR.FAIL}$ Output Low Voltage			0.4	V	I_{OL} = 4 mA, V_{CC} = 4.75V
I_{CC1}	Current from V_{CC}—Selected		30	45	mA	CS = V_{IL}. V_{CC} = 5.25V
I_{DD1}	Current from V_{DD}—Selected		20	35	mA	CS = V_{IL}, V_{CC} = 5.25V
I_{DD2}	Current from V_{DD}—Power Down		12	19	mA	CS = V_{IH}, V_{DD} = 12.6V

A.C. CHARACTERISTICS*

V_{CC} = 5V ± 5%; V_{DD} = 12V ±5%

Symbol	Parameter	Min.	Max.	Unit
t_{ENON}	Delay On		260	ns
t_{DISOFF}	Delay Off		70	ns
t_{CSON}	CS Enable		500	ns
t_{CSOFF}	CS Disable		70	ns

*These parameters are sample tested, not 100% tested.

POWER FAIL CHARACTERISTICS**

T_A = 0°C to 70°C

	Min.	Typ.	Max.	Test Conditions
V_{CCTH}	4.43V	4.60V	4.70V	
V_{DDTH}	10.75V	11.10V	11.28V	

**Power fail characteristics apply to 7110 Bubble Memory Data Integrity only and not to full memory operation.

AFN-01367B

CAPACITANCE* (T_A = 25°C)

Symbol	Parameter	Typ.	Max.	Unit	Test Conditions*
C_{IN}	Input Capacitance		10	pF	

*This parameter is periodically sampled and not 100% tested. Condition of measurement is f = 1 MHz.

OUTPUT CURRENTS (T_A = range specified in Table 1, V_{CC} = 5.0V ±5%, V_{DD} = 12V ±5%)

Parameter	Current (mA)			Test Conditions			
				Voltage Out		Voltage Out (7230-5 only)	
	Min.	Nom.	Max.	Min.	Max.	Min.	Max.
GEN.A, GEN.B CUT	62	75	81	5.7	11.5	5.5	11.6
GEN.A, GEN.B TRANSFER	34	40	49	5.7	12.2	5.5	12.2
REP.A, REP.B CUT	170	200	240	3.7	9.0	3.4	9.3
REP.A, REP.B TRANSFER	126	145	160	3.7	11.2	3.4	11.4
SWAP	111	125	134	3.1	9.7	2.7	9.9
BOOT.REP CUT	85	100	110	7.8	12.0	7.7	12.1
BOOT.REP TRANSFER	63	75	80	7.8	12.4	7.7	12.4
BOOT.SWAP	63	75	80	9.1	12.2	9.0	12.3

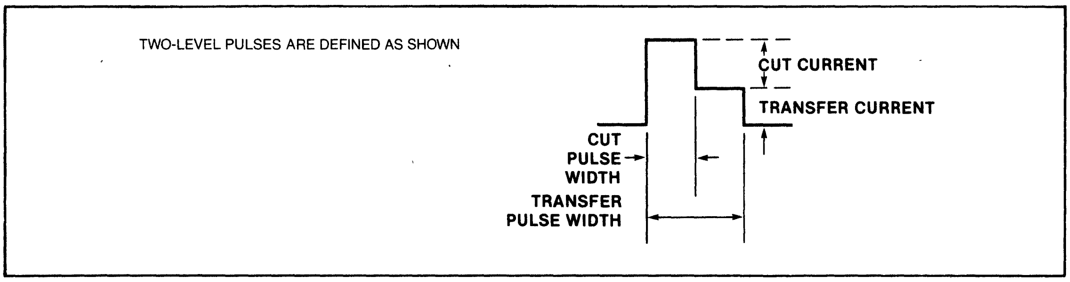
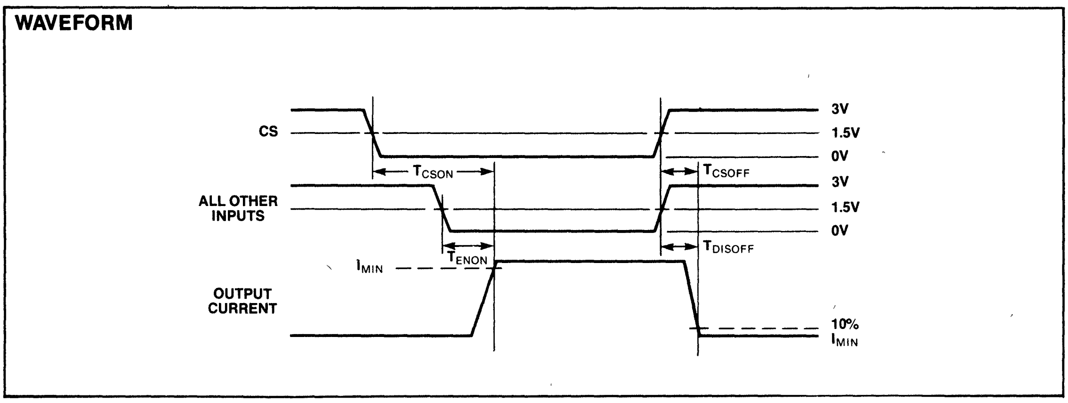


Figure 4. Output Pulse Diagram



intel®

7242
DUAL FORMATTER/SENSE AMPLIFIER FOR BUBBLE MEMORIES

7242	0 to 70°C
7242-5	−20 to +85°C

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

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

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

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

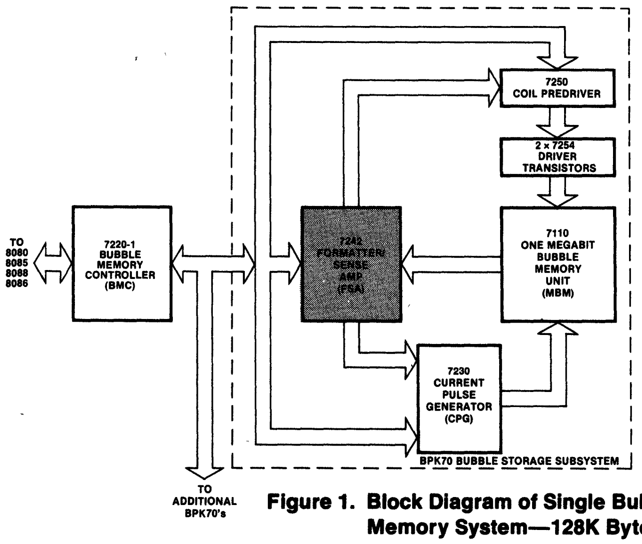


Figure 1. Block Diagram of Single Bubble Memory System—128K Bytes

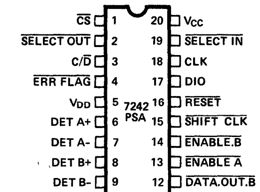


Figure 2. Pin Configuration

NOVEMBER 1982
ORDER NUMBER: 210799-001

Table 1. Pin Description

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

FUNCTIONAL DESCRIPTION

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

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

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

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

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

FIFO—The FIFO is a variable-length First-In-First-Out buffer utilized to store data passing to and from the MBM module. The FIFO is logically 272 bits in length in the "no error correction" mode. It is 270 bits in the "error correction" mode, since 256 bits of the data and a 14-bit error correction code must be used in this mode of operation.

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

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

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

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

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

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

COMMANDS

FSA Commands

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

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

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

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

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

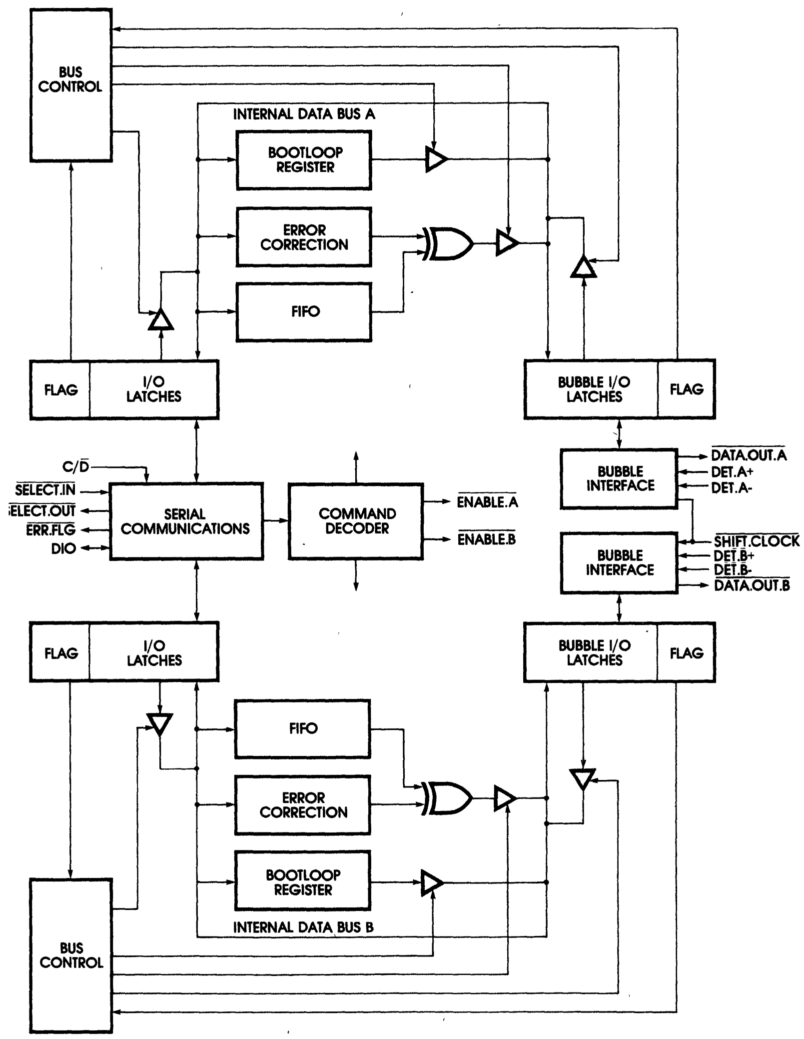


Figure 3. Logic Diagram

Table 2. Command Code Description

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

Table 3. Command Function Summary

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

SERIAL INTERFACE

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

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

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

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

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

DIO in the clock period following receipt of SELECT.IN. (A logic one indicates that the FSA shall accept the command.)

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

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

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

   a. For the Read ERR.FLG Status command the second SYNC can occur 40 clocks after the first SYNC. This SYNC (or SELECT.IN) causes each addressed FSA to send the appropriate Status Information. No further SYNCs (without C/D high) should be sent.
   b. For the Read MBM Data (or initialize) command the second SYNC must wait the appropriate number of SHIFT.CLOCKs to assure that valid data is available in the FIFO.
      After this wait, each addressed FSA channel sends one bit of data on the DIO line for each SYNC (or SELECT.IN) pulse.
   c. For the Read Bootloop Register command, the second SYNC can occur 60 clock cycles after the first SYNC. The data transfer then proceeds as in b. above.
   d. For the Write MBM Data or Write Bootloop commands, the DIO line is used to transfer data to the FSA on successive SYNC pulses. The first data bit can be transferred by a second SYNC pulse, 40 clock cycles after the first SYNC. (However, data to the MBM will not be available at the Dataout pins until 40 clock cycles after the SYNC which transferred it.) Each transfer to the addressed FSA will be initiated by a SYNC (or SELECT.IN).

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

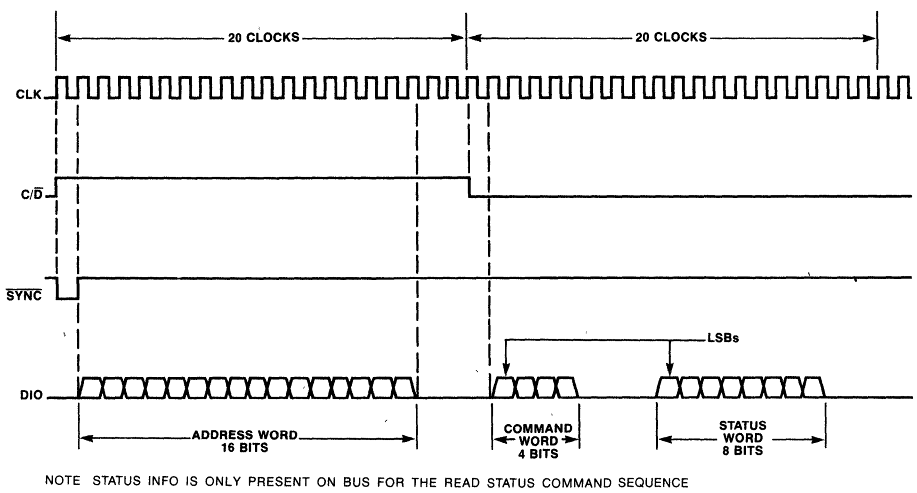


Figure 4. Command Sequences

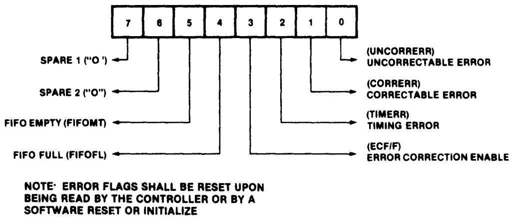
Data Sequences—Bubble data shall be passed between the Controller and FSAs in the following fashion (see Figure 5).

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

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

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

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



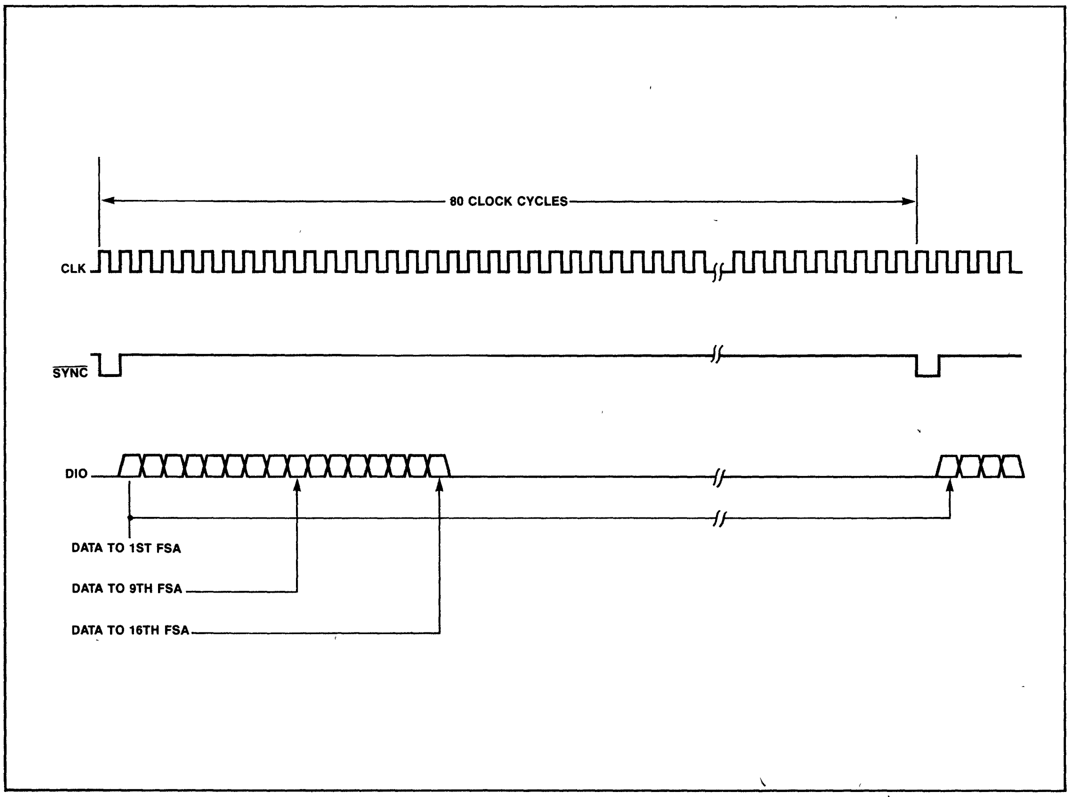


Figure 5. Data Sequences

BUBBLE INTERFACE

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

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

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

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

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

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

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

FSA ERRROR CORRECTION

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

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

DATA FORMAT

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

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

Data Block Format—
$o_1 e_1 o_1 e_1 o_2 e_2 o_2 e_2 \ldots o_{80} e_{80} o_{80} e_{80}$

1st bit   320th bit

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

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

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

ABSOLUTE MAXIMUM RATINGS*

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

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

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

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

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

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

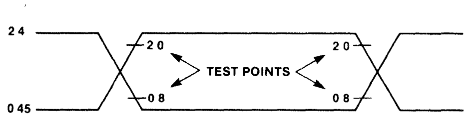
Symbol	Parameter	Min.	Max.	Unit	Test Conditions
t_p	Clock Period	240	500	ns	
t_f	Clock Phase Width	.45 t_p	.55 t_p		
t_n t_f	Clock Rise and Fall Time		30	ns	
t_{SIC}	$\overline{\text{SELECT.IN}}$ Setup Time to CLK	50		ns	
t_{CDC}	C/$\overline{\text{D}}$ Setup Time to CLK	50		ns	
t_{CYC}	$\overline{\text{SELECT.IN}}$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

CAPACITANCE (T_A = 25°C, V_{CC} = 0V, f = 1 MHz)

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

*DIO Write Mode

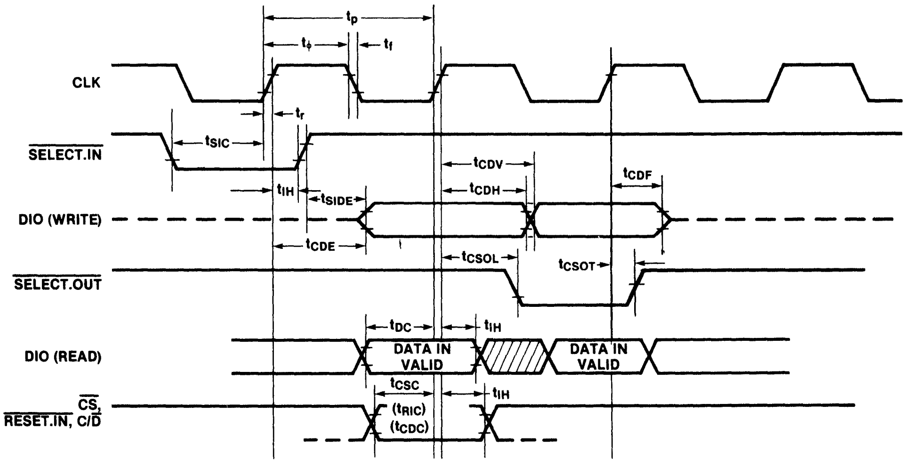
A.C. TESTING INPUT, OUTPUT WAVEFORM



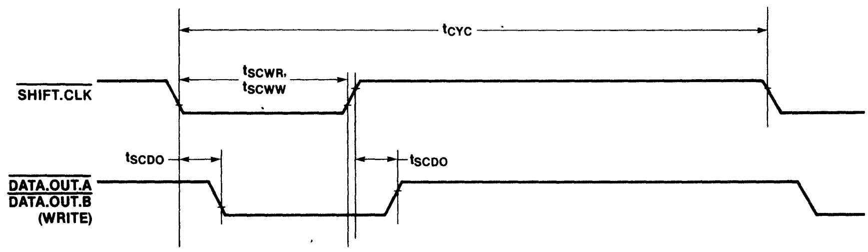
A.C. TESTING: INPUTS ARE DRIVEN AT 2.4V FOR A LOGIC "1" AND 0.45V FOR A LOGIC "0". TIMING MEASUREMENTS ARE MADE AT 2.0V FOR A LOGIC "1" AND 0.8V FOR A LOGIC "0".

WAVEFORMS

DIO INTERFACE TIMING



BUBBLE DATA INTERFACE TIMING



AFN-01358B

intel

7250
COIL PRE-DRIVE
FOR BUBBLE MEMORIES

7250	0 to 70°C
7250-5	−20 to +85°C

- **Very Low Power**
- **Power Fail Reset for Maximum Protection of Bubble Memory**
- **TTL Compatible Inputs**
- **Only One Power Supply Required, +12V**
- **CMOS Technology**
- **Standard 16-Pin Dual In-Line Package**

The Intel 7250 is a low power Coil Pre-Driver (CPD) for use with Intel Magnetics Bubble Memories. The 7250 is controlled by the Intel 7220-1 Bubble Memory Controller (BMC) and directly drives Quad VMOS transistor packs, which are connected to the coils of the bubble memory.

The 7250 is a high-voltage, high-current driver constructed using CMOS technology. The device has TTL compatible inputs and the outputs are designed to drive either low on-resistance VMOS transistors or bipolar transistors.

The 7250 is in a standard 16-pin dual in-line package.

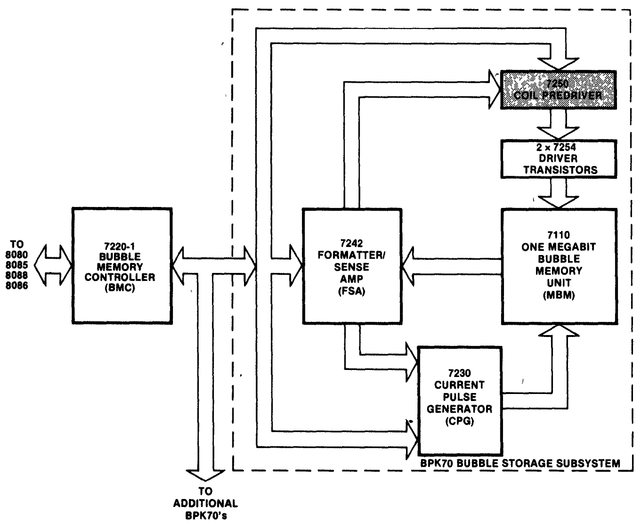


Figure 1. Block Diagram of Single Bubble Memory System—128K Bytes

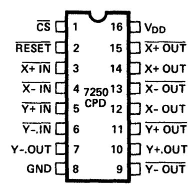


Figure 2. Pin Configuration

OCTOBER 1982
ORDER NUMBER: 210800-001

Table 1. Pin Description

Symbol	Pin No.	Description
$\overline{\text{CS}}$	1	Chip select. It is active low. When high chip is deselected and I_{DD} is significantly reduced.
$\overline{\text{RESET}}$	2	Active low input from RESET.OUT of 7220-1 Controller forces 7250 outputs inactive so that bubble memory is protected in the event of power supply failure.
$\overline{\text{X+IN}}$, $\overline{\text{.X−.IN}}$	3, 4	Active low inputs from Controller which turn on the high-current X outputs.
X−.OUT $\overline{\text{X−.OUT}}$ $\overline{\text{X+.OUT}}$ X+.OUT	12, 13, 14, 15	High-current outputs and their complements for driving the gates of the 7254 VMOS quad transistors which in turn drive the X coils of the bubble memory.
$\overline{\text{Y+.IN}}$, $\overline{\text{Y−.IN}}$	5, 6	Active low inputs from Controller which turn on the high-current Y outputs.
Y−OUT $\overline{\text{Y−.OUT}}$ $\overline{\text{Y+.OUT}}$ Y+.OUT	7, 9, 10, 11	High-current outputs and their complements for driving the gates of the 7254 VMOS quad transistors which in turn drive the Y coils of the bubble memory.

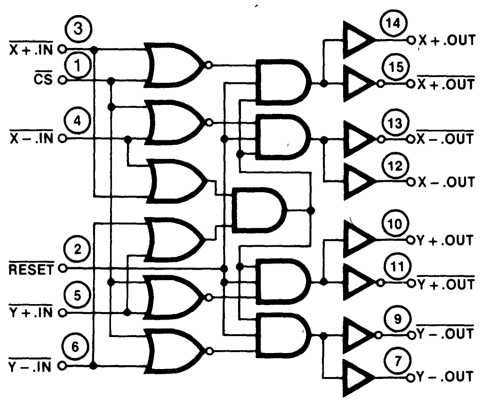


Figure 3. Logic Diagram

ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature Under Bias ... −40°C to +100°C
Storage Temperature .... −65°C to +150°C
Voltage on Any Pin with
Respect to Ground .... −0.5 to V_{DD} +0.5V
Supply Voltage, V_{PP} .... −0.5 to +14V
Output Current .... 250 mA
(One Output @ 100% Duty Cycle)

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

D.C. CHARACTERISTICS (T_A = see range specified on first page V_{DD} = 12V +5%, −10%; unless otherwise specified)

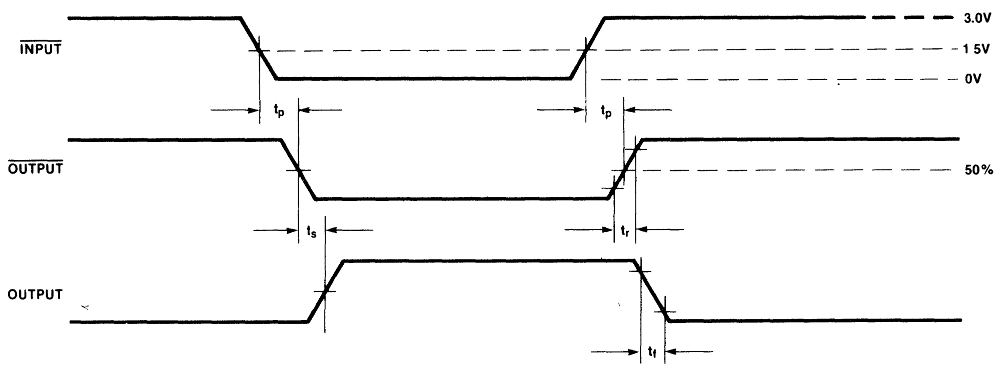
Symbol	Parameter	Limits Min.	Limits Typ.	Limits Max.	Unit	Test Conditions
$\|I_{IN}\|$	Input Current			5	μA	V_I = 0.8V
V_{IL}	Low-Level Input Voltage			0.8	V	
V_{IH}	High-Level Input Voltage	2.2			V	
V_{OL1}	Output Low Voltage			2.0	V	I_{OL} = 100 mA
V_{OL2}	Output Low Voltage			0.2	V	I_{OL} = 10 mA
V_{OH1}	Output High Voltage	V_{DD}−2			V	I_{OH} = −100 mA
V_{OH2}	Output High Voltage	V_{DD}−0.2			V	I_{OH} = −10 mA
I_{OL}	Output Sink Current	100			mA	V_{OL} = 2.0V
$\|I_{OH}\|$	Output Source Current	100			mA	V_{OH} = V_{DD}−2.0V
I_{DD0}	Supply Current			4.5	mA	Chip Deselected: $\overline{CS}$ = V_{IH}, V_{DD} = 12.6V
I_{DD1}	Supply Current			75	mA	f = 100 kHz, V_{DD} = 12.6V, Outputs Unloaded

AFN-01359B

A.C. CHARACTERISTICS (T_A = see range specified on first page
V_{DD} = 12V ±5%, unless otherwise specified)

Symbol	Parameter	Min.	Typ.	Max.	Unit	Test Conditions
t_{p1}	Propagation Delay from $\overline{X+.IN}$, $\overline{X-.IN}$, $\overline{Y+.IN}$, $\overline{Y-.IN}$			100	ns	500 pF Load
t_{p2}	Propagation Delay from $\overline{CS}$ or $\overline{RESET}$			150	ns	500 pF Load
t_r	Rise Time (10% to 90%)			45	ns	500 pF Load
t_F	Fail Time (90% to 10%)			45	ns	500 pF Load
t_S	Skew Between an Output and Its Complements			20	ns	

A.C. TEST CONDITIONS



CAPACITANCE* (T_A = 25°C, V_{DD} = 0V, V_{BIAS} = 2V, f = 1 MHz)

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

*This parameter is periodically sampled and is not 100% tested.

intel®

7254
QUAD VMOS DRIVE TRANSISTORS FOR BUBBLE MEMORIES

- Designed to Drive X and Y Coils of 7110 Bubble Memories
- No Bias Currents Required
- Fast Turn-On and Turn-Off: 30 ns Maximum
- Built-In Diode Commutates Coil Current When Transistor is Turned Off
- Operates from V_{DD} Only
- VMOS FET Technology
- N-Channel and P-Channel Transistors in the Same Package
- Standard 14-Pin Dual-In-Line Package

The 7254 is a quad transistor pack designed to drive the X and Y coils of Intel Magnetics Bubble Memories. Two 7254 packages are required for each bubble memory device. Each 7254 package would drive either the X or Y coil as shown under "circuit diagram." This recommended connection circuit takes into account the fact the Q1/Q2 and Q3/Q4 are tested as a pair for "On" resistance value to assure optimal bubble performance.

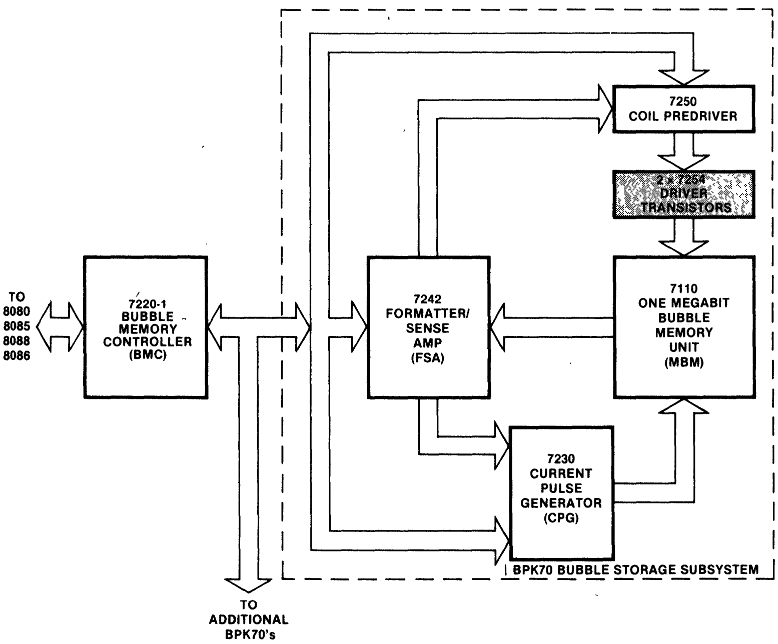


Figure 1. Block Diagram of Single Bubble Memory System—128K Bytes

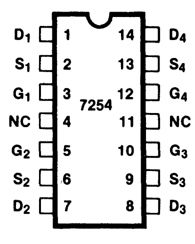


Figure 2. Pin Configuration

OCTOBER 1982
ORDER NUMBER: 210801-001

ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature Under Bias ... −40°C to +100°C
Storage Temperature .... −65°C to +150°C
Gate Voltage (with respect to
Source and Drain 15V
Continuous Drain Current 2A
Peak Drain Current 3A
Power Dissipation (T_A = 80°C) 1.05W
Power Dissipation (T_A = 25°C) 1.75W

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

D.C. CHARACTERISTICS All Limits Apply for N- and P-Channel transistors, T_A = −30° to 85°C unless otherwide noted.

Symbol	Parameter	Limits				Test Conditions
		Min.	Typ.	Max.	Unit	
BV_{DSS}	Drain-Source Breakdown Voltage	20			V	V_{GS} = 0, I_D = 10 μA
$V_{GS}(th)$	Gate-Source Threshold Voltage	0.8			V	V_{GS} = V_{DS}, I_D = 1 mA, T_A = 25°C
		0.65			V	V_{GS} = V_{DS}, I_D = 1 mA, T_A = 85°C
I_{GSS}	Gate Leakage Current			100	nA	V_{GS} = 12V, V_{DS} = 0, T_A = 25°C
I_{DSS}	Drain Leakage Current			500	nA	V_{GS} = 0, V_{DS} = 20V, T_A = 25°C
R_{DS}	On-Resistance for sum of Q1+Q2, Q3+Q4 (Note 1)	2 0	2.5	3.0	Ω	V_{GS} = 11.4V, I_D = 1A, T_A = 25°C
V_{F1}	Parasitic Diode Forward Voltage (Note 1)			.75	V	V_{GS} = 0V, I_D = 50 mA, T_A = 25°C
$V_{F}2$	Parasitic Diode Forward Voltage (Note 1)			1.20	V	V_{GS} = 0V, I_D = 1000 mA, T_A = 25°C

NOTE:
1 Pulse test—80 μs pulse, 1% duty cycle, r_{DS} increase 0.8%/°C.

A.C. CHARACTERISTICS T_A = 25°C

Symbol	Parameter	Min.	Typ.	Max.	Unit	Test Conditions
$T_{ON}(N)$	N-Channel Turn-On Time			20	ns	
$t_{ON}(P)$	P-Channel Turn-On Time			30	ns	
$t_{OFF}(N)$	N-Channel Turn-Off Time			20	ns	
$t_{OFF}(P)$	P-Channel Turn-Off Time			30	ns	

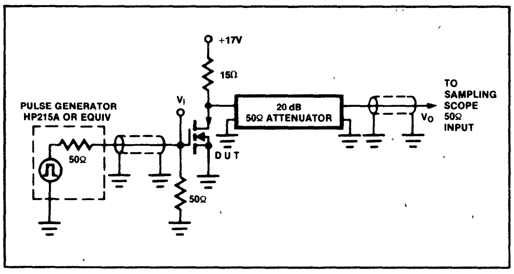


Figure 3. Switching Time Test Circuit

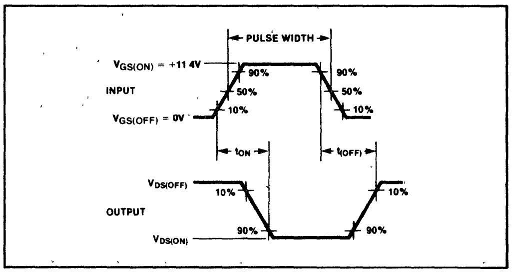


Figure 4. Switching Time Test Waveforms

AFN-01360B

CAPACITANCE $T_A = 25°C$

Symbol	Parameter	Min.	Typ.	Max.	Unit	Test Conditions
$C_{iss}(N)$	N-Channel Input Capacitance			175	pF	$V_{GS} = 0$, $V_{DS} = 12V$, f = 1 MHz
$C_{iss}(P)$	P-Channel Input Capacitance			190	pF	$V_{GS} = 0$, $V_{DS} = 12V$, f = 1 MHz

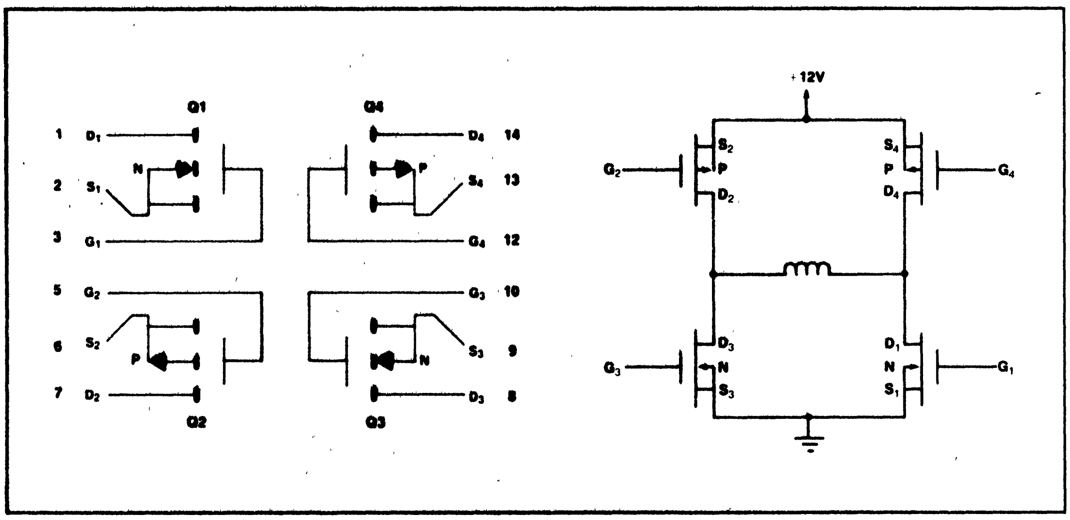


Figure 5. Circuit Diagram

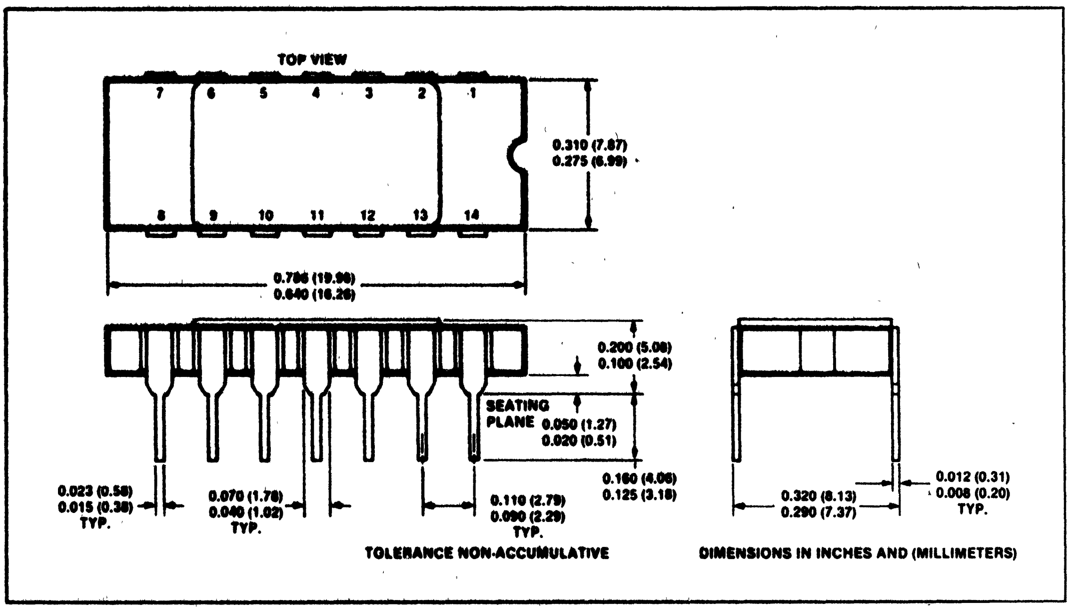


Figure 6. Packaging Information

AFN-01360B

7114
4-MEGABIT BUBBLE MEMORY

OPERATING FREQUENCY		CASE OPERATING TEMP. (°C)	NON-VOLATILE STORAGE (°C)
100 KHz	50 KHz		
7114 A-1	7114-1	0→75	−40→+90
7114 A-4	7114-4	10→55	−20→+75

- **4,194,304 Bits of Usable Data Storage**
- **Non-Volatile, Solid-State Memory**
- **True Binary Organization: 512-Bit Page and 8,192 Pages**
- **Major Track–Minor Loop Architecture**
- **Redundant Loops with On-Chip Loop Map and Index**
- **Block Replicate for Read; Block Swap for Write**
- **Single-Chip 20-Pin Dual In-Line Package**
- **Small Physical Volume**
- **Maximum Data Rate 400 Kbit/Sec (7114A)**
- **Average Access Time 40 msec (7114A)**

The Intel Magnetics 7114 (unless otherwise indicated 7114 refers also to 7114A) is a very high-density 4-megabit non-volatile, solid-state memory utilizing magnetic bubble technology. The usable data storage capacity is 4,194,304 bits. The defect-tolerant design incorporates redundant storage loops. The gross capacity of Intel Magnetics bubble memory is 5,242,880 bits.

The 7114 has a true binary organization to simplify system design, interfacing, and system software. The device is organized as 512 data storage loops each having 8,192 storage bits. When used with Intel Magnetics complete family of support electronics, the resultant minimum system is configured as 512K bytes of usable data storage. The support circuits also provide automatic error correction and transparent handling of redundant loops.

The 7114 has a major track–minor loop architecture. It has separate read and write tracks. Logically, the data is organized as a 512-bit page with a total of 8,192 pages. The redundant loop information is stored on-chip in the boot loop along with an index address code. The 7114 provides totally non-volatile data storage when operated within the stated limits.

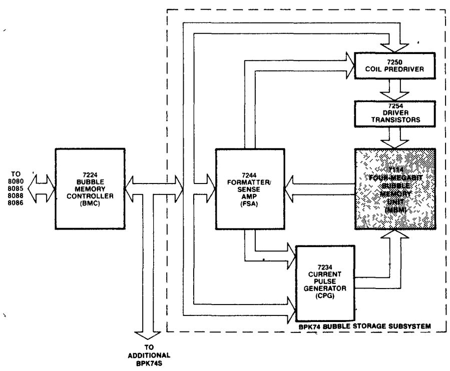


Figure 1. Block Diagram of Single Bubble Memory System—512K Bytes

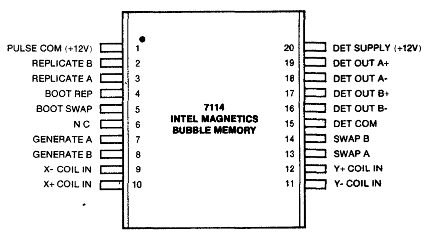


Figure 2. Pin Configuration

NOVEMBER 1982
ORDER NUMBER: 210783-001

Table 1. 7114 Pin Description

Symbol	Pin#	Name and Function
BOOT.REP	4	Two-level current pulse input for reading the boot loop.
BOOT.SWAP	5	Single-level current pulse for writing data into the boot loop. This pin is normally used only in the manufacture of the MBM.
DET.COM	15	Ground return for the detector bridge.
DET.OUT	16–19	Differential pair (A+, A− and B+, B−) outputs which have signals of several millivolts peak amplitude.
DET.SUPPLY	20	+12 volt supply pin.
GEN.A and GEN.B	7, 8	Two-level current pulses for writing data onto the input track.
PULSE.COM	1	+12 volt supply pin.
REP.A and REP.B	3, 2	Two-level current pulses for replicating data from storage loops to output track.
SWAP.A and SWAP.B	13, 14	Single-level current pulse for swapping data from input track to storage loops.
X−.COIL.IN, X+.COIL.IN	9, 10	Terminals for the X or inner coil.
Y−.COIL.IN, Y+.COIL.IN	11, 12	Terminals for the Y or outer coil.

The 7114 is packaged in a dual in-line leaded package complete with permanent magnets and coils for the in-plane rotating field. In addition, the 7114 has a magnetic shield surrounding the bubble memory chip to protect the data from external magnetic fields.

The operating data rate is 400 Kbit/sec for 7114A, and 200 Kbit/sec for 7114. The 7114 can be operated asynchronously and has start/stop capability.

GENERAL FUNCTION DESCRIPTION

The Intel Magnetics 7114 is a 4-megabit bubble memory module organized as two identical 2,048K binary half sections. See Major Track–Minor Loop architecture diagram. Each half section is in turn organized as four 512K subsections referred to as octants.

The module consists of a bubble die mounted in a substrate that accommodates two orthogonal drive coils that surround the die. The drive coils produce a rotating magnetic field in the plane of the die when they are excited by 90° phase-shifted triangular current waveforms. The rotating in-plane field is responsible for bubble propagation. One drive field rotation propagates all bubbles in the device one storage location (or cycle). The die-substrate-coil subassembly is enclosed in a package consisting of permanent magnets and a shield. The shield serves as a flux return path for the permanent magnets in addition to isolating the device from stray magnetic fields. The permanent magnets produce a bias field that is nearly perpendicular to the plane of the die. This field supports the existence of the bubble domains.

The package is constructed to maintain 1.5 degree tilt between the plane of the bias magnet faces and the plane of the die. This serves to introduce a small component of the bias field into the plane of the die. During operation when the drive coils are energized this small in-plane component is negligible. During standby or when power is removed the small in-plane field ensures that the bubbles will be confined to their appropriate storage locations. The direction of the in-plane field introduced by the package tilt (holding field) is coincident with the 0° phase direction of the drive field.

Architecture

A 7114 octant subsection is composed of the following elements shown on the architecture diagram.

STORAGE LOOPS

Each octant subsection contains eighty identical 8,192-bit storage loops to provide a total maximum capacity of 655,360 bits. The excess storage is provided for two purposes: a) it allows a redundancy scheme to increase device yield; and b) it provides the extra storage required to implement error correction.

REPLICATING GENERATOR (GEN)

The generator operates by replicating a seed bubble that is always present at the generator site (GEN).

INPUT TRACK AND SWAP GATE

Bubbles following generation are propagated down an input track. Bubbles are transferred to/from the input track from/to the 80 storage loops via series-connected swap gates spaced every two propagation cycles along the track. The swap gate's ability to transfer bubbles in both directions during an operation eliminates the overhead associated with removing old data from the loops before new data can be written. The swap gate is designed to function such that the logical storage loop position occupied by the bubble transferred out of each loop is filled by the bubble being transferred into each loop. Transferred-out bubbles propagate down the remaining portion of the input track where they are dumped into a bubble bucket guard rail.

OUTPUT TRACK AND REPLICATE GATE

Bubbles are read out of the storage loops in a non-destructive fashion via a set of replicate gates. The bubble is split in two. The leading bubble is retained in the storage loop and the trailing bubble is transferred onto the output track. Replicate gates are spaced every two propagation cycles along the output track.

DETECTOR

Bubbles, following replication, are propagated along the output track to a detector that operates on the magneto-resistance principle. The cylindrical bubble domains are stretched into long strip domains by a chevron expander and are then propagated to the active portion of the detector. The detector consists of a thin film, lying underneath a stack of chevrons, through which a current is passed. As the strip domain propagates below the thin-film detector, its magnetic flux causes a fractional change in film resistance which produces an output signal of several millivolts. The strip domain following detection is propagated to a bubble bucket guard rail. A "dummy" detector stack sits in the immediate vicinity. It is connected in series with the active detector and serves to cancel common mode pickup which originates predominately from the in-plane drive field.

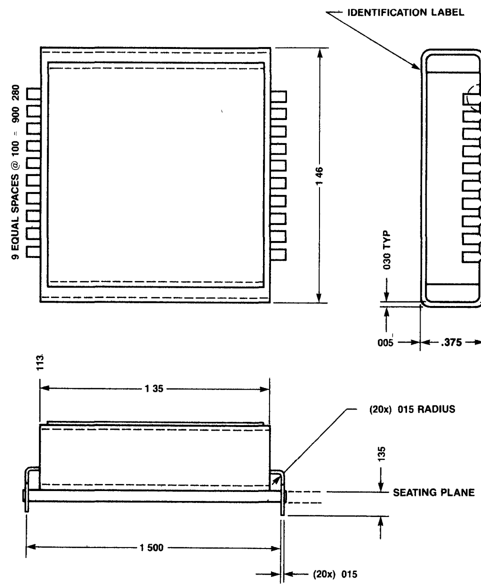
BOOT LOOP, BOOT SWAP, AND BOOT REPLICATE

One of the four octants in each half chip contains a functionally active Boot Storage Loop. This loop is used to store:

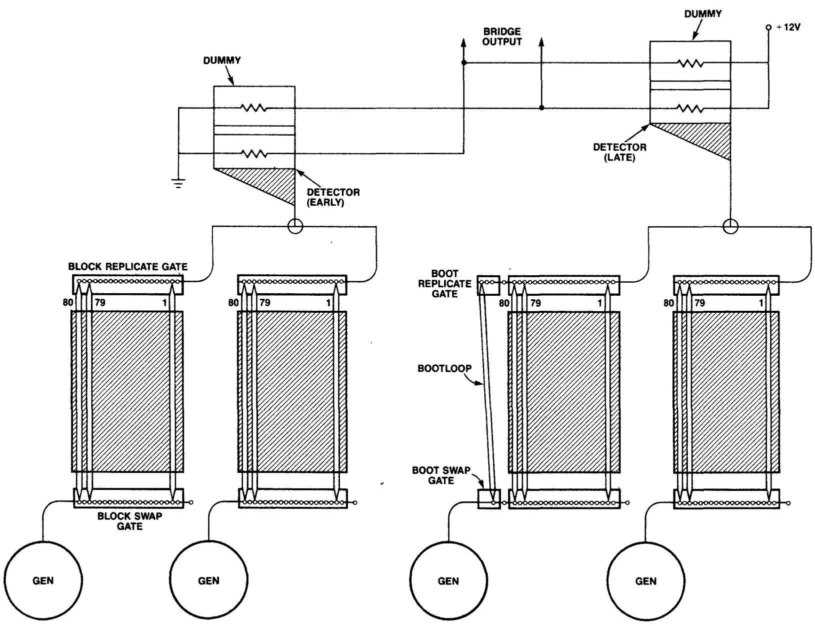
a) A loop mask code that defines which loops within the main storage area should be accessed. Faulty loops are "masked out" by the support electronics.

b) A synchronization code that assigns data addresses (pages) to the data in the storage loops. Since bubbles move from one storage location to the next every field rotation, the actual physical location of a page of data is determined by the number of field rotations that have elapsed with respect to a reference.

The boot loop is read from and written into via the same input and output tracks as the main storage loops. However, it has independently accessed swap and replicate gates. The boot swap, under normal circumstances, is intended only to be used during basic initialization at the factory at which time loop mask and synchronization codes are written. The boot replicate is intended to be accessed every time power is applied to the bubble module and its peripheral control electronics. At such a time, the control electronics would read and store the mask information, plus utilize the synchronization information to establish the location of the data circulating within the loops.



PACKAGE OUTLINE



MAJOR TRACK–MINOR LOOP ARCHITECTURE OF 7114 (ONE HALF SHOWN).

Figure 3. Package Outline and Device Architecture

ABSOLUTE MAXIMUM RATINGS*

Operating Case Temperature .... 0°C to 75°C Case
Relative Humidity 95%
Shelf Storage Temperature (Data
Integrity Not Guaranteed) .... −65°C to +150°C
Voltage Applied to DET.SUPPLY .... 14 Volts
Voltage Applied to PULSE.COM .... 12.6 Volts
Continuous Current between DET.COM and
Detector Outputs .... 10 mA
Coil Current .... 2.5A D.C. or A.C. RMS
External Magnetic Field for
Non-Volatile Storage .... 20 Oersteds
Non-Operating Handling Shock 200G
Operating Vibration (2 Hz to 2 KHz) 20G

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

D.C. AND OPERATING CHARACTERISTICS (T_C = Range Specified on First Page)

Parameter	Limits			Unit
	Min.	Nom.[1]	Max.	
RESISTANCE: PULSE.COM to GEN.A or GEN.B	12	30	58	ohms
RESISTANCE: PULSE.COM to REP.A or REP.B	13	27	35	ohms
RESISTANCE: PULSE.COM to SWAP.A or SWAP.B	20	47	71	ohms
RESISTANCE: PULSE.COM to BOOT.REP	3.5	8	23	ohms
RESISTANCE: PULSE.COM to BOOT.SWAP	5	20	49	ohms
RESISTANCE: DET.OUT A+ to DET.OUT A−	770	1190	2200	ohms
RESISTANCE: DET.OUT B+ to DET.OUT B−	770	1190	2200	ohms
RESISTANCE: DET.COM to DET.SUPPLY	560	950	2100	ohms

NOTE:
1 Nominal values are measured at 25°C.

AFN-01439A

DRIVE REQUIREMENTS (T_C = Range specified on First Page) (See note 2) Vdd = 12V ±5%

		7114			7114A			
Symbol	**Parameter**	**Min.**	**Nom.[1]**	**Max.**	**Min.**	**Nom.[1]**	**Max.**	**Units**
f_R	Field Rotation Frequency	49.95	50.00	50.05	99.90	100.00	100.10	KHz
I_{px}	X.Coil Peak Current		.58			1.6		A
I_{py}	Y.Coil Peak Current		.74			2.1		A
Θ_{1x}	X.Coil Positive Turn-On Phase	268	270	272	268	270	272	Degrees
Θ_{2x}	X.Coil Positive Turn-Off Phase	16	18	20	16	18	20	Degrees
Θ_{3x}	X.Coil Negative Turn-On Phase	88	90	92	88	90	92	Degrees
Θ_{4x}	X.Coil Negative Turn-Off Phase	196	198	200	196	198	200	Degrees
Θ_{1y}	Y.Coil Positive Turn-On Phase	0	0	0	0	0	0	Degrees
Θ_{2y}	Y.Coil Positive Turn-Off Phase	106	108	110	106	108	110	Degrees
Θ_{3y}	Y.Coil Negative Turn-On Phase	178	180	182	178	180	182	Degrees
Θ_{4y}	Y.Coil Negative Turn-Off Phase	286	288	290	286	288	290	Degrees
P_T	Total Coil Power		1.5			2.9		Watts
R_x	X.Coil D.C. Resistance		7.4			1.0		Ohms
R_y	Y.Coil D.C. Resistance		3.3			0.4		Ohms
L_x	X.Coil Inductance		89			15		μH
L_y	Y.Coil Inductance		78			14		μH

NOTES:
1. Nominal values are measured at T_C = 25°C.
2. See Figure 4 for test set-up and X-Y Coil waveform.

CONTROL PULSE REQUIREMENTS (see Notes 2 and 3) (T_C = Range Specified on First Page)

	Current (mA)			Phase of Leading Edge (Degrees)			Width (Degrees)		
Pulse	**Min.**	**Nom.**	**Max.**	**Min.**	**Nom.[1]**	**Max.**	**Min.**	**Nom.[1]**	**Max.**
GEN.A, GEN.B CUT	39	44	50	275 95	279 (late) 99 (early)	283 103	6	9	13.5
GEN.A, GEN.B TRANSFER	25	29	33	275 95	279 (late) 99 (early)	283 103	86	90	94
REP.A, REP.B CUT	130	148	165	284	288	292	6	9	13.5
REP.A, REP.B TRANSFER	100	115	130	284	288	292	86	90	94
SWAP	140	152	165	176	180	184	513	517	521
BOOT.REP CUT	33	38	42	284	288	292	6	9	13.5
BOOT.REP TRANSFER	25	29	33	284	288	292	86	90	94
BOOT SWAP	35	39	44	176	180	184	See Note 4		

NOTES:
1. Nominal values are at T_C = 25°C.
2. Pulse timing is given in terms of the phase relations as shown below. For example, a 7114 operating at t_R = 50.000 KHz would have a REP.A transfer width of 90° which is 5 μsec.
3. Two level pulses are described as shown below in Figure 5.
4. BOOT.SWAP is not normally accessed during operation. It is utilized at the factory to write the index address and redundant loop information into the bootstrap loops before shipment.

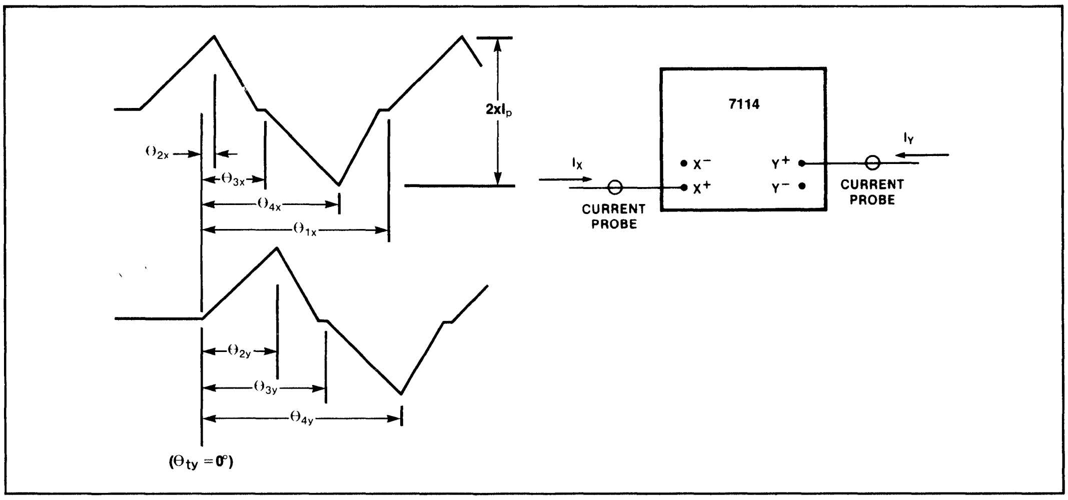


Figure 4. X-Y Coil Waveforms and Test Set-Up

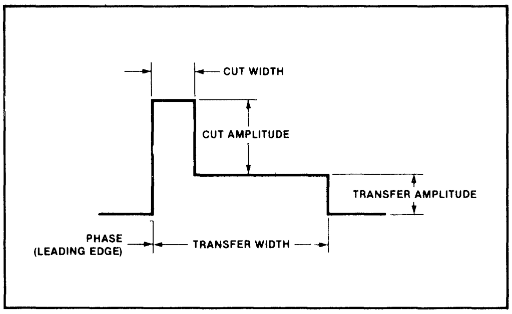


Figure 5. Two-Level Current Pulse

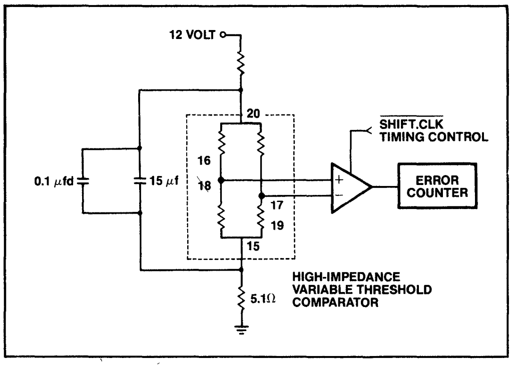


Figure 6. Test Set-Up for Output Voltage Measurement

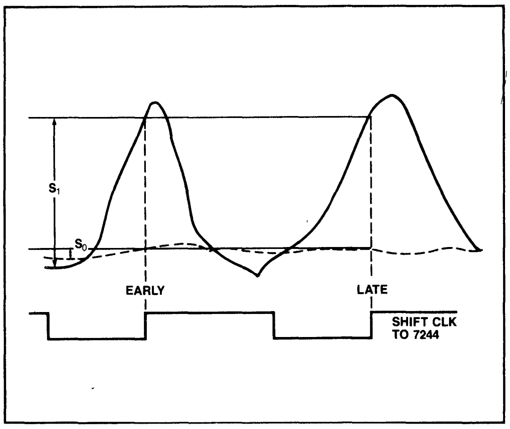


Figure 7. Detector Output Waveforms

OUTPUT CHARACTERISTICS (T_C = Range Specified on Front Page)

Symbol	Nom.	Units	Test Conditions
S_1	18	mV	See notes
S_0	1	mV	1, 2

NOTES:

1 Nominal values are measured at T_C = 25°C
2 See Figure 6 for test set-up, and Figure 7 for detector output waveforms and timing